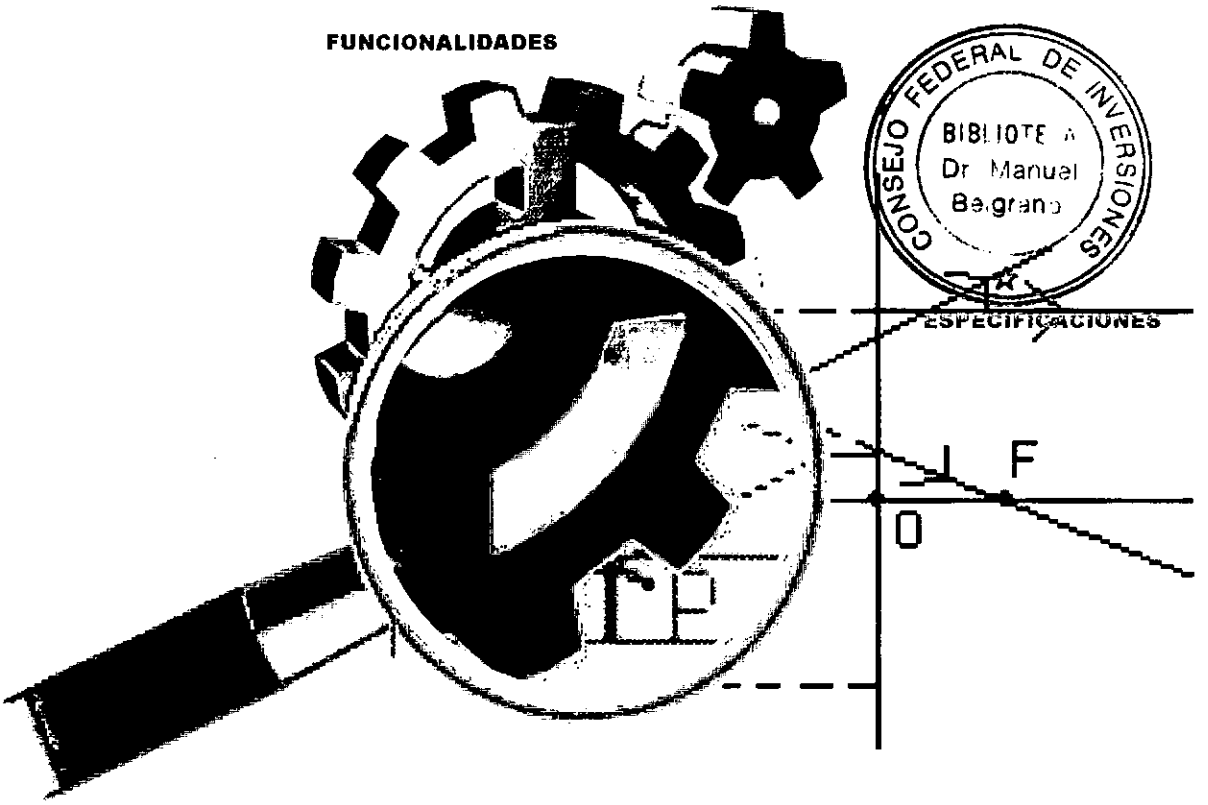
" LA CALIDAD EN EL PROCESO DE TESTING DE SISTEMAS "



CFI - 2002

PROGRAMA:

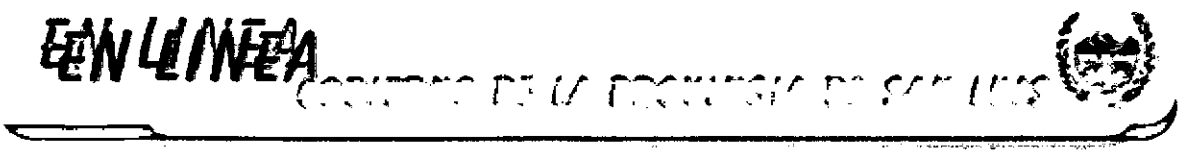
" SAN LUIS – GESTIÓN PARA LA IMPLEMENTACIÓN

DE UNA PROVINCIA DIGITAL "

PROYECTO:

" LA CALIDAD EN EL PROCESO

DE TESTING DE SISTEMAS "



ÍNDICE

RESUMEN GERENCIAL DE PROYECTO

El objetivo de este proyecto consiste en proveer al Gobierno de la Provincia de San Luis requisitos mínimos, lineamientos y documentación necesaria para generar una estrategia de testing de aplicativos metódicamente, permitiendo llevar a cabo los procedimientos de testing en forma adecuada y coherente, maximizando la cantidad de errores encontrados y obteniéndose así un producto final más confiable y de mayor calidad.

En la actividad 1 se observa el producto de una intensa investigación de la cual surge el desarrollo conceptual de los procedimientos teóricos necesarios para llevar a cabo el testing de aplicativos. Entre ellos podemos mencionar: definiciones, principios y atributos de la prueba de sistemas, métodos de prueba de software (pruebas de caja negra, pruebas de caja blanca, pruebas no funcionales), niveles de prueba, diseño de casos de prueba, prueba de sistemas web, etc.

En la actividad 2 se detalla un conjunto elaborado de requisitos mínimos, estrategias y procedimientos adecuados a las necesidades de la S.E.T.I., contemplando, entre otras cosas: la planificación de los procesos de prueba, plan de pruebas, estrategia, alcance, niveles de prueba, diseño, elaboración y ejecución de los casos de prueba, mecánica de la prueba, análisis de riesgo, criterios de finalización de las pruebas, ejecución del plan de pruebas, seguimiento de las mismas, etc.

En la actividad 3 se describe un conjunto de plantillas que sirven de base para la generación de la documentación de los resultados obtenidos de la ejecución

de los procedimientos e implementación de las estrategias de testing generadas en este proyecto. Las mismas sirven para: el registro, revisión y aprobación de casos de prueba, captura de pantallas y seguimiento de errores.

En la actividad 4 se muestra la aplicación en un caso práctico de testing sistema basado en las técnicas y estrategias generadas en las actividades anteriores. Para llevar a cabo el mismo la S.E.T.I. propuso que se realizara un testing funcional sobre tres módulos del Sistema de Capacitación en su versión alfa.

INTRODUCCIÓN

La implementación del proyecto "Autopista de la Información" en la Provincia de San Luis tiene como objetivo mejorar la calidad de vida del pueblo sanluiseño mediante el uso de las tecnologías de la información. En este marco, la Secretaría de Estado de Tecnologías de la Información ha contratado el desarrollo de aplicativos para diversas áreas.

El chequeo de presencia de errores, o testing, es una etapa indispensable en todo proceso de desarrollo de software. Hasta el momento, el Gobierno de la Provincia de San Luis realizaba el proceso de testing de los desarrollos y aplicativos contratados en forma prácticamente artesanal. Es inconcebible que esta situación perdure teniendo en cuenta que la puesta en producción de los aplicativos de la Autopista de la Información afecta a múltiples áreas del Gobierno en una forma sin precedentes.

La necesidad de un marco eficaz para ser utilizado en el testing de cada uno de los aplicativos hace evidente que debe sistematizarse el proceso.

El resultado del proceso de testing debe reflejar, como mínimo, si el software cumple con los requerimientos del Gobierno de la Provincia de San Luis, comportándose según lo esperado e interactuando correctamente con los otros sistemas y con tiempos de respuesta aceptables.

Considerando la magnitud y el número de sistemas contratados, la complejidad natural de las interacciones demandadas y que los procesos de prueba de software culminan con la aceptación de los mismos, surge la necesidad de

replantear las estrategias de testing. Las estrategias resultantes deben basarse en un sustento metodológico preciso y ordenado, ya que sin el mismo, la calidad del proceso de testing se ve deteriorada, generando consecuencias no deseadas en los procesos de auditoría del proyecto.

La ejecución de este proyecto brindará un trabajo de intensa investigación que permitirá generar estrategias y lineamientos necesarios para lograr procesos de testing confiables y coherentes donde se maximice la cantidad de errores encontrados.

OBJETIVO ESPECÍFICO

Proveer los requisitos mínimos y documentación necesaria para generar una estrategia de testing de aplicativos contemplando, entre otras cosas: la planificación de los procesos de prueba, la elaboración y ejecución de los casos de prueba y, finalmente, la evaluación de los resultados.

INFORME DEL DESARROLLO DE LA ACTIVIDAD N° 1

1 ENUNCIADO DE LA ACTIVIDAD

Investigación y desarrollo conceptual de los procedimientos y pasos a llevar a cabo en el testing de aplicativos.

En esta actividad se elaborarán, entre otros:

- Introducción a la verificación y validación de software.
- Definiciones y conceptos.
- Descripción de las fases del proceso de pruebas.
- Explicación de la importancia de la planificación de las pruebas.

2 DESARROLLO

La prueba de software es una actividad que de una u otra manera es llevada a cabo en algún momento al menos por su desarrollador original. Para ello, el equipo de desarrollo puede planear y realizar un proceso de prueba enfocado a un ambiente en el que espera operará su producto. Aunque esto es relativamente aceptable, puede no ser suficiente. Muchas veces el ambiente de operación pensado por el equipo de desarrollo puede diferir del ambiente que se presenta para la persona que utiliza finalmente dicho de software, por lo que en la prueba desarrollada se pueden haber ignorado algunas situaciones.

Lo ideal para llevar a cabo el testing de software es contar con un equipo de prueba de software (ver punto 2.8.1) que realice la misma a lo largo de todas las etapas del desarrollo (análisis, especificación, diseño, implementación y mantenimiento), planteando la estrategia adecuada que se debe aplicar en cada una de ellas.

La prueba de software es una actividad necesaria que ayuda a establecer la calidad de éste. La importancia de la misma se ve reforzada con el paso de los años convirtiéndose en un factor fundamental, llegándose a plantear que la prueba de software se aplique en cada una de las etapas del desarrollo como algo necesario.

En general, se puede decir que la prueba de software permite determinar si el producto generado satisface las especificaciones establecidas. Así mismo, una prueba de software permite detectar la presencia de errores que pudieran generar salidas o comportamientos inapropiados durante la ejecución del mismo.

Debemos tener en claro y diferenciar testing o prueba de software de depuración de software (debugging); el primero tiene como objetivo encontrar o detectar errores y el segundo busca el problema que los generó. La confusión surge porque cuando se encuentra un error se busca el por qué del mismo para resolver el problema. El límite entre testing y debugging es que en el testing por cada vez que se encuentra un error se debe registrar formalmente en un documento (en la Actividad 3 se generarán la plantillas para tal fin) y una vez finalizado el mismo, el equipo de desarrollo utiliza los resultados del testing,

registrados en el documento llevado a cabo para tal fin, para realizar la depuración de los errores.

En esta actividad se presentan técnicamente los conceptos teóricos sobre la prueba de software en general, orientado a objeto, basado en componentes y web. La misma no pretende desarrollar toda la teoría de testing de sistemas, sino más bien, dar los conceptos que forman la base para plantear lineamientos confiables de una estrategia de testing que sean coherentes y que maximicen la cantidad de errores encontrados.

2.1 PRUEBA DE SISTEMAS

Una de las actividades asociadas al desarrollo de software es el proceso prueba o "testing" del mismo. Debido a la importancia que ha adquirido el testing de software se ha establecido que el mismo es una actividad fundamental dentro de cada una de las etapas del proceso de desarrollo.

La prueba es indispensable, puesto que a partir de ella se puede determinar la calidad de los productos implementados; a pesar de esto, no es difícil percibir como su importancia se ha subestimado y en ocasiones hasta ignorado.

La **prueba o testing de software** se define como: "*Una actividad en la cual un sistema o componente es ejecutado bajo condiciones especificas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente*". Aunque una definición más usada para la prueba de software es: "*El proceso de ejecutar un programa con el fin de encontrar errores*".

Cuando se habla de condiciones específicas, en la definición anterior, se puede suponer la presencia de una especie de ambiente de operación de la prueba, para el cual deben existir determinados valores para las entradas y las salidas, así como también ciertas condiciones que delimitan a dicho ambiente de operación. Todo lo mencionado se plasma sobre un plan de pruebas.

El **Plan de prueba** establece las estrategias, recursos y cronograma de ejecución de las pruebas. La **estrategia** se refiere al tipo de pruebas, sus objetivos, nivel de análisis y porcentajes de los resultados aceptables sobre las pruebas que se llevarán a cabo en cada etapa del desarrollo e implementación de los sistemas y componentes.

Las pruebas de software crecen a medida que se progresa en la creación del software hasta que se adquiera una estabilidad en las evoluciones del ciclo de vida, lo que significa, un gran esfuerzo en las últimas fases. Debido a esto, es indispensable mantener el modelo de pruebas a lo largo del ciclo de vida del software, aunque sea necesario cambiar dicho modelo debido a modificaciones o refinamiento natural que sufre el software en la creación.

El **modelo de prueba** describe como se van a probar los componentes ejecutables con distintos métodos de prueba, como de integración y sistema, y como han de ser probados aspectos del sistema. El modelo de pruebas está compuesto de *Casos de prueba*, *Procedimientos de prueba* y *Componentes de prueba*, los cuales serán explicados a continuación.

Se define un **caso de prueba** como: "*Un conjunto de entradas, condiciones de ejecución y resultados esperados diseñados para un objetivo particular*".

En un proceso de prueba de software, se pueden identificar las siguientes acciones:

I. *preparar una serie de casos de prueba,*

II. *llevar a cabo dichos casos de prueba,*

III. *decidir cuándo suspender la prueba,*

IV. *evaluar los resultados generados por la prueba,*

V. *emitir un criterio de evaluación.*

De las mismas surgen cuestionamientos tales como: ¿Cómo seleccionar casos de prueba representativos?, ¿Cuántas pruebas realizar? o bien ¿Cómo decidir si es o no de calidad el producto evaluado?, los cuales permiten entender que cada una de estas acciones requiere especial atención.

El caso de prueba especifica una forma de probar el sistema, incluyendo que probar, los datos de entrada y salida, y las condiciones de la prueba. Surge a raíz de un requerimiento o un conjunto de requerimientos cuya implementación justifica la realización de una prueba.

El **Procedimiento de prueba** especifica como realizar uno o más casos de prueba o parte de ellos. Tiene instrucciones precisas de como interactuar con un sistema o componente, o la forma de configurar un componente de prueba.

Es posible que un procedimiento de prueba sea combinado con uno o varios casos de prueba, y viceversa.

Un **componente de prueba** automatiza uno o más procedimientos de prueba.

Puede ser desarrollado utilizando cualquier lenguaje de programación o herramienta de automatización. Se utiliza para probar componentes mediante la provisión de los datos de entrada y evaluación de los datos de salida. Puede incluir el monitoreo de actividad entre los componentes que se prueban y la generación de reportes con los resultados de la prueba.

Un **defecto** representa una anomalía en el funcionamiento de los sistemas. Es utilizado para describir aquellos aspectos que los desarrolladores utilizarán para rastrear y corregir la falla.

Los términos tales como falla, equivocación y error, pueden considerarse como sinónimos, sin embargo, dentro del contexto de prueba de software no es prudente realizar esta suposición. Con el propósito de evitar confusiones y presentar conceptos básicos en materia de pruebas, se presentan estas definiciones:

a) Equivocación (mistake):

Acción del ser humano que produce un resultado incorrecto.

b) ***Defecto o falta (fault):***

Proceso o definición de dato incorrecto en un programa de computadora. El resultado de una equivocación.

c) ***Falla (failure):***

Resultado incorrecto. El resultado de una falla.

d) ***Error (error):***

Magnitud por la que el resultado es incorrecto.

A continuación se definen dos conceptos que se deben tener bien en claro en la prueba de software ya que la misma forma parte de un concepto más amplio, referenciado generalmente como verificación y validación.

La **Verificación** se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. Ésta responde a la pregunta: ¿Se está construyendo el producto correctamente?

La **Validación** se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. Ésta responde a la pregunta: ¿Se está construyendo el producto correcto?, es decir, ¿Se ajusta a los requisitos del cliente?

La prueba confirma los niveles de calidad alcanzados en todas las etapas del ciclo de vida mediante la aplicación adecuada de métodos y herramientas.

Las estrategias actuales de verificación y validación se aplican a todas las etapas del ciclo de vida. Por ejemplo, los cuatro criterios básicos para la verificación y validación de las especificaciones de requisitos y de diseño son:

- **Completitud:** una especificación es completa si todas sus partes están presentes y cada parte está completamente desarrollada.
- **Consistencia:** interna (las especificaciones no deben entrar en conflicto unas con otras) y externa (las especificaciones no deben entrar en conflicto con especificaciones o entidades externas). Los detalles de las especificaciones deben tener claros antecedentes en especificaciones anteriores o en los objetivos del sistema.

- **Factibilidad:** una especificación es factible si los beneficios del sistema exceden sus costos.
- **Verificabilidad:** una especificación es verificable si se puede identificar una técnica económicamente factible que permita determinar si un software satisfará o no una especificación.

2.1.1 PRINCIPIOS PARA PROBAR SOFTWARE

Los siguientes son principios que han surgido de las mejores prácticas de pruebas de software:

- La definición del resultado esperado a la salida del programa es una parte integrante y necesaria del caso de prueba. Si el resultado esperado de la prueba no ha sido predefinido cabe la posibilidad que un resultado plausible pero erróneo se interprete como correcto.
- El grupo encargado de realizar la prueba de software debe ser, en lo posible, distinto del que realiza el desarrollo del mismo.
- Examinar a conciencia el resultado de cada prueba. Ocurre que errores que suelen aparecer casualmente son errores expuestos por los casos de prueba pero que se escaparon a la detección por falta de inspección.
- Evitar los casos de prueba desechables a menos que el programa sea verdaderamente desechable. Es una práctica sentarse frente al monitor, improvisar un caso de prueba y pasarlo al programa. El problema está en la pérdida de tiempo, ya que cuando se tenga que probar nuevamente (por

haberse corregido algún error) se tendrá que reinventar la prueba. El planeado y documentación de los casos de prueba juegan un papel importantísimo en el testing de software.

- Determinar cuándo finaliza la prueba, ya que no definir esto desde un comienzo trae muchos problemas.
- La probabilidad de encontrar errores adicionales en un módulo del programa es proporcional al número de errores ya encontrados en dicho módulo. Contrariamente a lo que la intuición indicaría, esto ha sido verificado en muchos programas.
- No alterar nunca el programa para que la prueba resulte más fácil.
- La prueba como cualquier otra actividad debe comenzar con la definición de sus objetivos.

2.2 ESTRATEGIAS DE PRUEBAS

Una estrategia de prueba de software integra las técnicas de diseño de casos de prueba en una serie de pasos planificados que dan como resultado una correcta construcción del software. La estrategia proporciona un plano o guía que describe los pasos a llevar a cabo en el testing de software, cuándo se deben planificar y realizar esos pasos, y cuánto esfuerzo, tiempo y recurso se van a requerir.

El Plan de trabajo para realizar las pruebas responde a las preguntas: qué, cuándo y cómo llevar a cabo la misma.

Una **estrategia de prueba** es un procedimiento que busca reducir los riesgos de cometer errores. Esta debe:

- Prever los riegos más peligrosos
- Señalar la forma de reducir los riesgos (tácticas de prueba)
- Distinguir los posibles riesgos dentro del proceso total

Entre las tácticas de prueba podemos mencionar: los planes de pruebas específicos, técnicas particulares, uso de herramientas, etc.

En resumen, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de los casos de prueba, la ejecución de las pruebas la agrupación y evaluación de los datos resultantes.

En la Actividad 2 del presente proyecto se desarrollará la estrategia de prueba de software que se adecue a las necesidades de la S.E.T.I.

Nota:

Los **riesgos** son condiciones que pueden conducir a pérdidas. Un riesgo es peligroso en la medida en que sea posible (probable).

Las **pruebas** tienen por objeto reducir la posibilidad de riesgos mientras que una táctica busca reducir la posibilidad de que un riesgo específico tenga lugar.

La **estrategia** es una planeación de tácticas que busca reducir los riesgos en el proceso de desarrollo.

2.2.1 FUNDAMENTOS DE LA PRUEBA DEL SOFTWARE

Los desarrolladores de software son por naturaleza constructivos mientras que lo que se busca con la prueba es buscar la forma de mostrar que tiene errores. En el testing se requiere que se descarten las ideas preconcebidas sobre la *corrección* del software que se acaba de desarrollar y se supere cualquier conflicto de intereses que aparezcan cuando se descubren errores. La prueba de software es uno de los pasos de la Ingeniería de Software que se puede ver como destructivo en lugar de constructivo, pero esto sólo es un punto de vista. Desde otra perspectiva más optimista, la prueba de software intenta de descubrir la máxima cantidad errores.

2.2.1.1 Objetivos de la Prueba de Software

La prueba del software es una fase clave del ciclo de vida de un producto. En ella se determina la disposición del producto para ser entregado basándose en criterios preestablecidos de correctitud y calidad. La actividad de verificación del software debe ser, por lo tanto, cuidadosamente planificada con el objeto de poder asegurar que la fase de prueba garantiza esos niveles de correctitud y calidad.

Existen dos enfoques a la hora de probar un software que no son auto-excluyentes y de hecho son complementarios:

- El hecho de realizar una prueba no garantiza la ausencia de defectos, sino solamente se demuestra la existencia de éstos.

- La prueba de software se realiza con el propósito de encontrar algo que difiera a las especificaciones planteadas para el producto o bien, para detectar la presencia de situaciones que pudieran generar resultados inapropiados.

Aunque ambos enfoques pueden orientar el sentido de una prueba, entre los objetivos principales de la prueba de software podemos mencionar:

I. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.

II. Un buen caso de prueba es aquél que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

III. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Otra característica deseable del proceso de prueba es que sea capaz de encontrar diferentes clases de errores en la mínima cantidad de tiempo posible y con el mínimo esfuerzo.

2.2.1.2 Facilidad de Prueba

La facilidad de prueba del software tiene que ver con que si el software es probable o no y de serlo, cuan fácil de probar es.

Los siguientes puntos de comprobación permiten verificar si un software es fácil de probar:

Operatividad: Cuando mejor funcione, más eficientemente se puede probar.

Observabilidad: Lo que se ve es lo que se prueba.

Controlabilidad: Cuanto mejor se pueda controlar el software, más se puede automatizar y optimizar.

Capacidad de Descomposición: Controlando el ámbito de las pruebas, se puede aislar más rápidamente los problemas y llevar a cabo mejores pruebas de regresión.

Si el sistema de software está construido con módulos independientes entonces los módulos del software se pueden probar independientemente.

Simplicidad: Cuanto menos haya que probar, más rápidamente podremos probarlo.

Estabilidad: Cuanto menos cambios, menos interrupciones a las pruebas.

- Los cambios del software son poco frecuentes.
- Los cambios del software están controlados.
- Los cambios del software no invalidan las pruebas existentes.
- El software se recupera bien de los fallos.

Facilidad de Comprensión: Cuanta más información se tenga, más inteligentes serán las pruebas.

- El diseño se ha entendido perfectamente.
- Las dependencias entre los componentes internos, externos y compartidos se han entendido perfectamente.
- Se han comunicado los cambios del diseño.
- La documentación técnica es instantáneamente accesible.
- La documentación técnica está bien organizada.
- La documentación técnica es específica y detallada.
- La documentación técnica es exacta.

2.2.1.3 Atributos de una Buena Prueba

Una buena prueba está caracterizada por cumplir los siguientes atributos:

I. Tiene una alta probabilidad de encontrar un error.

II. No debe ser redundante.

III. Debería ser la mejor de las pruebas realizables.

IV. No debería ser ni demasiado sencilla ni demasiado compleja.

2.2.2 FACTORES DE PRUEBA EN EL DESARROLLO DE SOFTWARE

Los factores de prueba son aquellos aspectos que deben ser aprobados como parte de una estrategia de prueba de software o corresponden a los riesgos que la estrategia considera:

- Riesgo: son aquellos factores que pueden producir resultados incorrectos, y
- Factor de prueba: correctitud.

Debemos tener en cuenta que no todo factor de prueba tiene que ser considerado sino que esto depende de la clase de software que se desarrolle.

Entre los factores de prueba más comunes podemos mencionar:

Correctitud: adecuación del software a las necesidades del usuario, es decir, si las especificaciones y los objetivos satisfacen al mismo. Para probar este factor debemos responder a la pregunta: ¿El sistema hace lo que se espera que haga de acuerdo a lo requerido?

Fiabilidad o confiabilidad: cantidad de fallas por período de tiempo. Definir qué son fallas y posibles tipos. Para probar este factor debemos responder a la pregunta: ¿El sistema es confiable todo el tiempo?

Eficacia: consumo de recursos que hará el sistema. Tiempos de respuesta a consultas específicas. Para probar este factor debemos responder a la pregunta: ¿Se ejecutará el sistema en mi hardware lo mejor posible?

Integridad: mantener datos coherentes pase lo que pase. Definir hasta qué punto el sistema detecta errores en los datos (inconsistencia). Para probar este factor debemos responder a las preguntas: ¿Es seguro el sistema?, ¿Es consistente?

Completitud: grado en que se han implementado las funciones requeridas. Para probar este factor debemos responder a la pregunta: ¿Se han implementado todos los casos de uso, módulos, funciones, etc.?

Testeabilidad: facilidad de probarlo. Para probar este factor debemos responder a la pregunta: ¿Se puede probar el sistema?

Flexibilidad: facilidad para cambiar componentes. Posibilidad de agregar o quitar módulos ante nuevas necesidades del usuario. Para probar este factor debemos responder a la pregunta: ¿Se puede cambiar el software?

Mantenibilidad: facilidad para mantenerlo. Calidad de la documentación, comentarios en fuentes, etc. Para probar este factor debemos responder a la pregunta: ¿El software se puede corregir?

Reutilización: reuso de los componentes para otros sistemas u otros eventos del sistema. Administrar correctamente los repositorios de componentes. Para probar este factor debemos responder a la pregunta: ¿Puedo reusar alguna parte del sistema?

Interoperabilidad: posibilidad de acoplarlo con otros sistemas. Para probar este factor debemos responder a la pregunta: ¿Puede interactuar el sistema con otro?

Acoplabilidad: conectividad entre módulos. Para probar este factor debemos responder a las preguntas: ¿Cuál es el grado de conectividad del sistema?, ¿Los módulos son independientes?

Auditabilidad: facilidad para comprobar las acciones que los usuarios llevan acabo en el sistema. Para probar este factor debemos responder a las

preguntas: ¿Permite auditar?, ¿Es auditable?, ¿Registra las acciones llevadas a cabo por los usuarios?

Facilidad de uso: esfuerzo requerido para aprender, manejar e interpretar lo que realiza el software. Para probar este factor debemos responder a la pregunta: ¿Se puede usar con facilidad el sistema?

Portabilidad: esfuerzo de trasferencia de una plataforma de hardware-software a otra. Para probar este factor debemos responder a la pregunta: ¿Se puede usar en otra máquina?

2.3 MÉTODOS DE PRUEBA DEL SOFTWARE

"*La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación*".

Los métodos de pruebas de software pueden ser clasificados en tres grandes grupos: Prueba de Caja Blanca, Prueba de Caja Negra y Prueba de Requerimientos no Funcionales.

2.3.1 PRUEBA DE CAJA BLANCA

La prueba de caja blanca, denominada también prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño del programa para los casos de prueba, es decir, se aprovecha del conocimiento de la estructura interna del mismo.

La prueba de caja blanca permite obtener casos de prueba que:

I. Garanticen que se recorran por lo menos una vez todos los caminos independientes de cada módulo.

II. Recorran todas las decisiones lógicas en sus vértices verdadero y falso.

III. Ejecuten todos los ciclos en sus límites y con sus límites operacionales

IV. Recorran las estructuras internas de datos para asegurar su validez.

La pruebas de caja blanca enfocan su atención a los detalles procedimentales del software, por ello la implementación de estas pruebas depende fuertemente de la disponibilidad de código fuente. Este tipo de pruebas, permiten generar casos para probar y validar los caminos de cada módulo, las condiciones lógicas, los bucles y sus límites, así como también para las estructuras de datos. Las pruebas de caja blanca también son conocidas como pruebas estructurales.

Una justificación para la realización de este tipo de prueba (en vez de invertir el tiempo asegurando que se han alcanzado los requisitos funcionales del programa -prueba de la caja negra-, y que a su vez surge como ventaja de emplear tiempo y energía probando las minuciosidades lógicas de un Software) la encontramos en :

I. Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa. Generalmente se cometen más errores al diseñar o implementar funciones que no son habituales realizar.

II. A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma normal.

III. Los errores de digitación (escritura) son aleatorios.

Algunas de las pruebas más significativas dentro de este enfoque se detallan en los puntos subsiguientes:

2.3.1.1 Notación de Grafo de Flujo

A continuación se definen los conceptos sobre grafos en que se basan las pruebas estructurales.

Grafo de flujo: representan el flujo de control de un programa y ayudan en la obtención de conjuntos de prueba. Está formado por un conjunto de nodos y arcos.

Nodos: cada nodo del grafo representa una o más sentencias del programa.

Arcos: son líneas dirigidas que unen dos nodos. Los arcos entre nodos representan flujo de control. Un arco debe terminar en un nodo.

Región: es un área rodeada por nodos y arcos.

Nodo predicado: es un nodo que contiene una condición.

Caminos independientes: es un camino que introduce por lo menos un nuevo conjunto de sentencias de proceso (debe moverse por lo menos por un arco nuevo en el camino).

Medida de la complejidad ciclomática *V(G)*: da un valor límite para el número de pruebas que se deben diseñar y ejecutar para garantizar que se cubren

todas las sentencias del programa. Este límite es el *número ciclomático*, que mide el número de caminos linealmente independientes para un grafo de flujo dado, y que viene dado por la fórmula:

V(G) = 1 + **d** ; donde **d** es el número de nodos predicado del grafo G.

Conjunto básico: colección de caminos que garantizan la ejecución por lo menos una vez de todas las sentencias del programa.

La Figura 1 muestra un ejemplo de grafo de flujo con nodos predicado, con V(G) = 1 + 3 = 4.

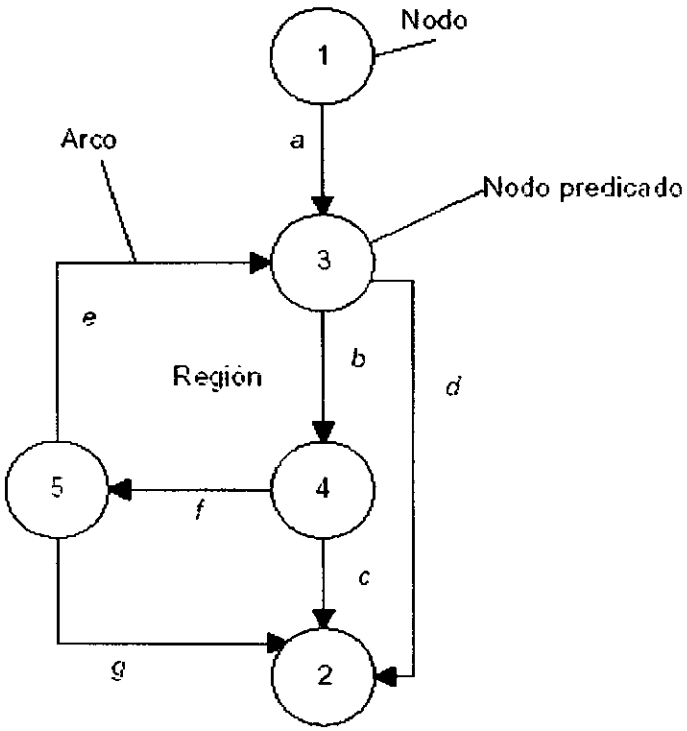


Figura 1 – Grafo de Flujo

2.3.1.2 Matrices de Grafos

Son útiles para mecanizar la obtención del grafo de flujo y la determinación de un conjunto básico de caminos:

- En cada celda se indica la arista que conecta dos nodos dados.
- La representación en forma de matriz de conexiones reemplaza el identificador de la arista por un 1.
- Peso de enlace: da información adicional sobre la matriz de conexión. De forma sencilla un peso 1 indica que existe una conexión y un peso 0 indica que no existe conexión.

La Figura 2 muestra la matriz de grafo y la matriz de conexiones correspondiente al grafo de flujo de la figura 1.

NODO

NODO	1	2	3	4	5
1			*a*		
2					
3		*d*		*b*	
4		*c*			*f*
5		*g*	*e*		

NODO

	1	2	3	4	5
1			*1*		
2					
3		*1*		*1*	
4		*1*			*1*
5		*1*	*1*		

Figura 2 – Matriz de Grafo y Matriz de Conexiones

2.3.1.3 Estrategias de Pruebas Estructurales

A continuación se describen brevemente las diferentes estrategias de pruebas estructurales o de Caja Blanca.

2.3.1.3.1 Prueba por Cobertura de Sentencias

Consiste en seleccionar casos de prueba que garanticen que cada sentencia o secuencia de sentencias sin puntos de decisión ha/ n sido probada/ s al menos una vez.

2.3.1.3.2 Prueba por Cobertura de Ramas

Consiste en seleccionar un conjunto de caminos de tal forma que cada rama del programa cubre en al menos un camino. Un 100% de cobertura de ramas garantiza un 100% de cobertura de sentencias, pues toda sentencia está en alguna rama.

2.3.1.3.3 Prueba de Caminos

Es la estrategia de prueba de Caja Blanca más exhaustiva y consiste en seleccionar casos de prueba de tal forma que cada camino posible del programa es ejecutado al menos una vez.

En este tipo de prueba se realiza un análisis sobre una representación gráfica de un programa denominada grafo de control. En este grafo, los nodos representan bloques de instrucciones de un programa y los flujos de ejecución

para dichas instrucciones se representan por medio de aristas. A partir de este grafo, se puede identificar un conjunto básico de caminos de ejecución, sobre el cual se pueden realizar pruebas con el propósito de probar el flujo de ejecución de los caminos en una unidad.

Este método permite al diseñador derivar una medida de la complejidad lógica de un programa y usarla como una guía para definir un conjunto básico de caminos de ejecución. Se garantiza que los casos de prueba que verifican el conjunto básico ejecutan todas las sentencias del programa al menos una vez. La derivación de casos de prueba se realiza según los siguientes pasos:

I. Derivar un grafo de flujo partiendo del diseño o del código fuente.

II. Determinar la complejidad ciclomática de este grafo de flujo.

III. Determinar un conjunto básico de caminos linealmente independientes.

IV. Preparar casos de prueba que fuercen la ejecución de cada camino del conjunto básico.

La existencia de un simple bucle puede implicar la existencia de un número infinito de caminos y también pueden existir caminos inalcanzables para cualquier valor de la entrada. Debido a estos problemas se han sugerido otras estrategias que intentan eliminarlos:

2.3.1.3.3.1 Prueba del camino simple

Consiste en la ejecución de todos los caminos simples (un camino simple es aquél que no contiene la misma rama más de una vez).

2.3.1.3.3.2 Prueba de caminos linealmente independientes

Consiste en la ejecución de todos los caminos linealmente independientes a través del grafo de flujo del programa. Un concepto que debemos recordar para esto es el de número ciclomático, el cuál es igual al número de casos de prueba requeridos para satisfacer la estrategia.

La obtención del *índice de efectividad de las pruebas* permite conocer el grado de extensión en que los casos de prueba satisfacen una estrategia de prueba particular para un programa dado y un conjunto de casos de prueba. Siendo T una estrategia de prueba para cubrir una clase de objetos (como caminos, caminos simples, caminos linealmente independientes, ramas o sentencias), el *índice de efectividad de las pruebas* se define como la proporción entre el número de objetos probados al menos una vez y el número total de objetos.

Los apartados siguientes muestran, con un mayor grado de profundidad, las estrategias de prueba del camino y la prueba de ciclos.

2.3.1.3.4 Prueba de Condiciones

Basándose de igual forma en un grafo de control, pueden generarse casos de prueba para elementos individuales de expresiones lógicas. De esta forma, se pretende probar cada condición con todas sus posibles alternativas.

2.3.1.3.5 Prueba de Ciclos

A partir del grafo de control, pueden generarse casos de prueba para las iteraciones definidas en los programas con el propósito de verificar si se realizan de forma correcta.

Los posibles errores en las construcciones de bucles o ciclos son:

- De iniciación.
- De indexación o incremento.
- En los límites.

Esta técnica se centra exclusivamente en la validez de las construcciones de bucles. Se definen cuatro clases de bucles diferentes y un conjunto de pruebas para cada tipo de bucle:

2.3.1.3.5.1 Bucles simples

Si n es el máximo número de pasos permitidos por el bucle, entonces:

I. Saltar el bucle completamente.

II. Pasar una sola vez por el bucle.

III. Hacer m pasos por el bucle con $m < n$.

IV. Hacer n–1, n y n+1 pasos por el bucle.

2.3.1.3.5.2 Bucles anidados

I. Comenzar en el bucle más interior. Disponer todos los demás bucles en sus valores mínimos.

II. Realizar pruebas de bucle simple con el bucle más interior mientras se mantienen los bucles exteriores con valores mínimos. Añadir pruebas de fuera de rango o de valores excluidos.

III. Progresar hacia fuera realizando pruebas para el siguiente bucle, manteniendo los demás bucles exteriores en valores mínimos y los demás bucles anidados con valores "típicos".

IV. Continuar hasta que todos los bucles hayan sido probados.

2.3.1.3.5.3 Bucles concatenados

Pueden ser probados como bucles simples si cada bucle es independiente de los otros. Si no es así, se puede utilizar el método de bucles anidados.

2.3.1.3.5.4 Bucles no estructurados

Bajo ningún concepto son aceptables. Los bucles no estructurados se deben rediseñar, no se deben probar.

2.3.1.3.6 Prueba de Definición de Datos

Estas pruebas son realizadas con el objetivo de encontrar posibles contradicciones o redundancias en la definición de los datos utilizados en el software. Para ello, se realiza un análisis del comportamiento de cada uno de los datos o cada una de los flujos de ejecución.

2.3.2 PRUEBA DE CAJA NEGRA

Las pruebas de Caja Negra, también llamada pruebas funcionales, de comportamiento o a gran escala, se centran en los requisitos funcionales del software ignorando la estructura de control del mismo. Esta permite al Ingeniero de Software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

Este tipo de pruebas concentra la atención en generar casos de prueba que permitan verificar los requisitos funcionales de un programa. A diferencia de las pruebas de caja blanca, que se basan en la lógica interna del software, este tipo de pruebas se concentra en su funcionalidad, por lo que mucho del trabajo se realiza interactuando con la interfaz del software. Los casos de prueba generados bajo este enfoque, se diseñan a partir de valores entrada y salida. De esta forma, se puede determinar la validez de una salida para un conjunto de entradas proporcionadas.

La aplicación de pruebas de caja negra permiten detectar errores como:

I. Funciones incorrectas o ausentes.

II. Errores de interfaz.

III. Errores en estructuras de datos o en accesos a bases de datos externas.

IV. Errores de rendimiento.

V. Errores de inicialización y de terminación.

Las pruebas se diseñan para responder, entre otras, las siguientes preguntas:

I. ¿Cómo se prueba la validez funcional?

II. ¿Qué clases de entrada compondrán buenos casos de prueba?

III. ¿Es el sistema particularmente sensible a ciertos valores de entrada?

IV. ¿De qué forma están aislados los límites de una clase de datos?

V. ¿Qué volúmenes y niveles de datos tolerará el sistema?

VI. ¿Qué efectos sobre la operación del sistema tendrán combinaciones específicas de datos?

En el ciclo de vida del software, la prueba de la Caja Blanca debería realizarse al principio del proceso de prueba, mientras que la prueba de la Caja Negra debería realizarse en etapas posteriores.

Estas son algunas de las pruebas (métodos) más conocidas en este contexto:

2.3.2.1 Partición Equivalente

La partición equivalente intenta definir un caso de prueba que descubra clases de errores y por tanto reducir el número de casos de prueba necesarios. Ésta

se basa en una evaluación de las clases de equivalencia para una condición de entrada.

La idea de esta técnica, consiste en dividir los valores válidos y no válidos para entradas y salidas en un número reducido de particiones, de forma que, el comportamiento del software sea el mismo para cualquier valor contenido en una partición particular. El propósito principal de una partición es reducir la cantidad de casos de prueba generados en el proceso. Para llevar a cabo los mismos, se procede dividiendo el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

Las clases de equivalencia se pueden definir de acuerdo con los siguientes criterios o directrices:

I. Si una condición de entrada especifica un rango, se definen una clase de equivalencia válida (dentro del rango) y dos inválidas (por debajo del rango y por encima del rango).

II. Si una condición de entrada requiere un valor específico, se definen una clase de equivalencia válida (el valor específico) y dos inválidas (por debajo del valor y por encima del valor).

III. Si una condición de entrada especifica un miembro de un conjunto, se definen una clase de equivalencia válida (dentro del conjunto) y una inválida (fuera del conjunto).

IV. Si una condición de entrada es lógica, se definen una clase válida (verdadero) y una inválida (falso).

De igual forma se construyen las particiones de equivalencia para los valores del dominio de salida.

2.3.2.2 Análisis de los Valores Límite

La generación de casos de prueba en esta técnica, se enfoca en los valores límites bajo la consideración de que existe una tendencia a fallar precisamente cuando el software trabaja con valores extremos de la /s variable /s de entrada /s. Generalmente, los valores establecidos para generar los casos de prueba son el mínimo, valores un poco arriba del mínimo, valor máximo y valores un poco arriba del máximo.

Este método complementa la partición equivalente dado que selecciona casos de prueba en los "bordes" de una clase. Además de centrarse en las condiciones de entrada, también deriva casos de prueba para el dominio de salida. Las directrices para derivar casos de prueba son:

I. Para una condición de entrada que especifica un rango limitado por los valores ***a*** y ***b***, los casos de prueba deben incluir los valores ***a*** y ***b*** y los valores justo por debajo de ***a*** y justo por encima de ***b***.

II. Para una condición de entrada que especifica un número de valores, los casos de prueba deben incluir el valor mínimo, el máximo y los valores justo por debajo del mínimo y justo por encima del máximo.

III. Aplicar las directrices I y II a las condiciones de salida.

IV. Si las estructuras de datos internas tienen límites preestablecidos, debe diseñarse un caso de prueba que ejercite la estructura en sus límites.

2.3.2.3 Pruebas Según la Experiencia

En este tipo de testing la generación de casos de prueba se realiza a partir de la intuición y la experiencia. La idea básica es redactar una lista de las posibles fallas o de las posibles situaciones en las cuales suele ocurrir algún problema y así desarrollar casos de prueba basados en la información contenida en estas listas.

2.3.2.4 Tablas de Decisión

Este tipo de prueba permite describir el comportamiento de un programa a partir de un conjunto de acciones que este realiza cuando se opera bajo determinadas condiciones. En este enfoque, las condiciones pueden ser interpretadas como entradas de un programa y las acciones como las salidas producidas. Para ello se pueden utilizar conectores lógicos **y** (and), **o** (or) y **no** (not). Al involucrar aspectos de lógica este tipo de prueba se hace más rigurosa y permite además transformar una especificación en lenguaje natural en una especificación más formal.

2.3.2.5 Técnicas de Grafos de Causa-Efecto

Las técnicas de grafos de causa-efecto proporcionan una concisa representación de las condiciones lógicas y sus correspondientes acciones. Las mismas siguen cuatro pasos:

I. Se listan para un módulo las causas (condiciones de entrada) y los efectos (acciones), asignando un identificador a cada uno de ellos.

II. Se desarrolla un grafo de causa-efecto.

III. Se convierte el grafo en una tabla de decisión.

IV. Las reglas de la tabla de decisión se convierten a casos de prueba.

2.3.3 PRUEBA DE REQUERIMIENTOS NO FUNCIONALES

El testing de requerimientos no funcionales tiene que ver con aquellas pruebas que se realizan al entorno sobre el cual se ejecuta el nuevo software o sistema desarrollado (sistema operativo, hardware, software subyacente, etc.) y que no tienen que ver con las funcionalidades del mismo. Abarca entre otros:

2.3.3.1 Prueba de Seguridad

La prueba de seguridad intenta verificar que los mecanismos de protección incorporados en el sistema protegerán al mismo de ataques o intromisiones inapropiadas. La seguridad del sistema debe ser probada tanto en su invulnerabilidad frente a ataques frontales como a ataques por los flancos y retaguardia. Para vulnerar el sistema "vale todo", desde obtener claves de acceso por cualquier medio externo, atacar al mismo con software diseñado

para romper cualquier defensa que se haya construido, etc.; se debe bloquear el sistema, negando así el servicio a otros usuarios, se deben producir errores a propósito, curiosear datos públicos, etc.

Con suficiente tiempo y recursos, una buena prueba de seguridad terminará por penetrar en el sistema. Lo que se debe lograr hacer es que el coste de penetración sea mayor que el valor de la información obtenida mediante la misma.

En la prueba de seguridad se validan la disponibilidad, integridad y confiabilidad de datos y servicios.

2.3.3.2 Prueba de Performance

En la prueba de performance se validan los tiempos de acceso y respuesta del sistema. Para llevar a cabo las mismas se requiere tanto de instrumentación de software como de hardware.

2.3.3.3 Prueba de Stress

En la prueba de stress se valida el uso del sistema en sus límites de capacidad y se verifica las reacciones que se producen. Para llevar a cabo las mismas, al igual que la prueba de performance, se requiere tanto de instrumentación de software como de hardware.

2.4 NIVELES DE PRUEBA DEL SOFTWARE

Un nivel de prueba permite especificar el alcance de la prueba de software que se realiza, se pueden identificar principalmente dos niveles:

2.4.1 BAJO NIVEL

En este nivel están consideradas todas aquellas pruebas que se realizan a componentes individuales de un programa. Las pruebas que se pueden realizar en este nivel son:

2.4.1.1 Pruebas de Unidad

Como su nombre lo indica, este tipo de prueba se aplica a elementos de software individualmente, excluyendo todos aquellos casos en los que se considere la interacción con otras unidades. El propósito fundamental de una prueba de unidad es descubrir diferencias entre la especificación del módulo de la interfaz y el comportamiento efectivo.

2.4.1.2 Pruebas de Integración

Las pruebas de integración se realizan con el propósito de verificar la arquitectura de un sistema. Una vez que ya se ha probado que las unidades funcionan de forma correcta en forma aislada, se procede a probar cómo funcionan al integrarlas con otras, de forma que se llegue a probar el comportamiento de un sistema final.

La forma en que se puede organizar la integración de las unidades para la prueba puede realizarse siguiendo enfoques tales como el ascendente, descendente, u otros.

El enfoque ascendente parte de los módulos atómicos (las unidades o módulos del nivel inferior en la estructura del programa) hacia los módulos subordinados de los niveles intermedios hasta llegar al módulo principal. Este parte de lo particular a lo general.

Mientras que el enfoque descendente parte del módulo principal hacia los distintos módulos subordinados y así sucesivamente en cada nivel hasta llegar a los módulos atómicos. Es decir, parte de lo general a lo particular.

2.4.2 ALTO NIVEL

En éste nivel las pruebas se orientan a un producto completo; para ello se proponen las siguientes alternativas:

2.4.2.1 Pruebas de Sistema

Este tipo de pruebas permiten probar el sistema como un todo así como también aspectos relacionados con la integración del producto a otros sistemas.

2.4.2.2 Pruebas de Usabilidad

El diseño de pruebas de usabilidad, requiere considerar a los usuarios que trabajan con el producto, con el propósito de observar sus respuestas hacia éste. De ésta forma se puede observar discrepancias existentes entre las interfaces implementadas y los requerimientos de los estilos de trabajo de los usuarios finales.

2.4.2.3 Pruebas de Función

Este tipo de pruebas tiene como objetivo detectar inconsistencias entre la especificación funcional de un programa y su comportamiento actual.

2.4.2.4 Pruebas de Aceptación

Realizando estas pruebas se puede comparar el producto final con las necesidades finales de los usuarios.

2.4.2.5 Pruebas de Regresión

Las pruebas de regresión son recomendables cuando partes del software se modifican, ya que permiten verificar que los cambios realizados no generan comportamientos no deseados.

2.5 DISEÑO DE CASOS DE PRUEBA

El diseño de pruebas para el software puede requerir tanto esfuerzo como el propio diseño inicial del producto. Sin embargo, muchas veces, no se presta demasiada importancia a los mismos y al final se termina o realizando una mala planificación de las pruebas o realizado un doble trabajo por no haber diseñado correctamente las mismas.

Se deben diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

Cualquier producto de ingeniería puede comprobarse de una de estas dos formas:

I. Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa, y al mismo tiempo, buscar errores en cada función. (Prueba de caja negra)

II. Conociendo el funcionamiento del producto, se pueden desarrollar pruebas que aseguren que "todas las piezas encajan", o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. (Prueba de caja blanca)

La definición del plan, es un aspecto determinante para la efectividad de la prueba de software, ya que los resultados obtenidos, después de haberse realizado este proceso, permitirán tener ciertas directrices para calificar el comportamiento del elemento evaluado.

El diseño de casos de prueba debe tener en cuenta dos alternativas para la generación de las mismas: casos generados de forma automática y casos administrados por el usuario.

Entre las generalidades de la generación automática de casos de prueba, se describen:

- Automatizar la realización de pruebas, tanto como sea posible, permitirá ejecutar programas con casos de prueba asociados. Con esto se pretende ofrecer al usuario una directriz sobre la calidad de la componente.
- La prueba que la herramienta debe preparar tendrá que consistir de un conjunto de *casos de prueba*, los cuales se aplicarán a las funcionalidades implementadas en cada componente.

Dado que la prueba depende de la ejecución controlada del código debemos proveer valores para que el mismo se ejecute con éstos. Cada conjunto de **datos** creado para la ejecución del código, junto con el **resultado esperado** de una ejecución correcta, es lo que se denomina ***caso de prueba***, como se resume a continuación:

```
Datos + ResultadoEsperado = CasoDePrueba
```

Es necesario crear un conjunto suficientemente grande y amplio para encontrar tantos errores como sea posible y asegurar la presencia de todas las funciones que fueron especificadas originariamente.

Existen muchas técnicas para el desarrollo de casos de prueba. El procedimiento más simple es permitir a la persona que prueba crear los datos a

su gusto y usarlos. Esto puede funcionar más o menos bien, dependiendo del grado de conocimiento que se tenga sobre el sistema, pero no es adecuado para una prueba completa del producto.

A continuación se listan las posibles formas de determinar casos de prueba:

- creación intuitiva
- conjunto exhaustivo de todos los datos posibles
- datos que representen todos los datos disponibles
- datos que representen el uso real del sistema
- datos que causen un efecto deseado
- datos que causen la ejecución de todas las sentencias
- datos que causen la ejecución de todos los caminos
- datos diseñados para ser improcesables

Evidentemente, no es posible realizar todas los puntos mencionados, por lo cual es necesario definir un objetivo de la prueba. Sin un objetivo, podría ser suficiente seleccionar datos distribuidos aleatoriamente sobre el conjunto de datos posibles. Sin embargo, estudios sobre la utilización de esta técnica han demostrado que descubre solo el 60% de los errores.

2.6 PRUEBA DE SOFTWARE ORIENTADO A OBJETOS

La metodología *orientada a objetos* es actualmente una de las nuevas alternativas para desarrollo de software con mayor aceptación. Esto es debido a las facilidades y beneficios que ofrece.

Si bien es cierto que este nuevo enfoque ha sido adoptado por muchos, es también cierto que estas personas han tenido que enfrentarse y adaptarse a cambios bastante significativos. Conceptos como objeto, herencia y encapsulación marcan una nueva perspectiva que contrasta con los modelos de desarrollo anteriormente utilizados.

Esta nueva percepción ha impedido que áreas de vital importancia, como la prueba de software, no hayan alcanzado el desarrollo y la robustez que requieren. Es importante destacar que mucho del software que se está fabricando hoy es orientado a objetos, por lo que resulta evidente la necesidad de contar con herramientas o al menos guías de acción para realizar esta actividad.

Esta falta de trabajos quizá sea debida a la suposición y afirmación inicial de algunos autores de que el software orientado a objetos, al tener como una característica esencial el reuso, está propenso a presentar menos errores que el software construido usando las metodologías tradicionales. Lamentablemente, las experiencias obtenidas han demostrado que no es así.

2.6.1 GENERALIDADES DEL MODELO DE DESARROLLO ORIENTADO A OBJETOS

En cierto grado, algunos de los conceptos de la orientación a objetos han estado presentes en los lenguajes de programación que se han utilizado; sin embargo, la programación orientada a objetos aparece formalmente durante la década de los '80, como una alternativa de desarrollo de software que ofrece características muy atractivas, tales como:

- Mejor modularidad en los programas.
- Mayor flexibilidad.
- Reuso.

Esta última característica no es nada despreciable en ambientes de trabajo donde se requiere construir software en tiempos muy cortos.

La programación orientada a objetos puede definirse como la *descripción de tareas en términos de los objetos que la misma tarea envuelve, así como de las propiedades de dichos objetos.*

A pesar de lo simple de la definición, se puede apreciar que un elemento muy representativo en este modelo es el *objeto*. Profundizando sobre este tema, es posible decir que la programación orientada a objetos gira en torno a dos conceptos: el *objeto* y la *clase*.

En los lenguajes de programación imperativos, los elementos más significativos son las estructuras de datos y los procedimientos, los cuales aparecen en forma desligada. En el modelo orientado a objetos, existe una unidad que junta

estos elementos, esta unidad es el objeto. La definición formal de un objeto está contenida en una clase a la que pertenece. Esta definición contiene las características del objeto, las cuales se modelan en términos de los atributos, datos que el objeto utiliza, y de los métodos, que son las funciones que manipulan a estos datos.

A continuación se presentan descripciones más amplias de clase y objeto, que permitirán comprender mejor lo mencionado anteriormente:

Un objeto tiene un *estado y un comportamiento*. El estado del objeto lo determina un conjunto de atributos. El comportamiento de un objeto está definido por el conjunto de métodos del objeto.

Una clase describe el comportamiento común de un grupo único de objetos. Describe todos los métodos y los atributos de esos objetos. Los objetos son creados mediante la instanciación de clases.

Un *objeto* es una entidad instanciada la cual tiene:

I. un *conjunto de operaciones* a las cuales responde,

II. un *estado,* el cual es afectado por un subconjunto de operaciones, y

III. la *habilidad de enviar mensajes* a otros objetos para invocar operaciones.

Una *clase* especifica las propiedades de un objeto y contiene:

I. una *interfaz,* la cual detalla como se accede a las propiedades públicas de la clase,

II. un *cuerpo de código* que implementa las operaciones definidas en la interfaz, y

III. *variables de instancia*, las cuales implementan el estado del objeto.

Además de clase y objeto, existen otros conceptos relacionados con esta forma de programación; *encapsulación*, *polimorfismo* y *herencia*, son algunos de los más destacables. En el punto siguiente, se abordan cada uno de ellos, dentro del contexto de prueba de software.

2.6.2 CONSIDERACIONES EN LA PRUEBA DE SOFTWARE ORIENTADO A OBJETOS

Al igual que el software construido utilizando otros enfoques de desarrollo, todo producto construido a partir del modelo orientado a objetos requiere ser sometido a pruebas con el propósito de garantizar su calidad. En términos generales, se puede decir que los dos enfoques más representativos en materia de pruebas, de caja blanca y de caja negra (ver puntos 2.3.1 y 2.3.2 respectivamente), son aplicables al software orientado a objetos en cierta medida. Sin embargo, existen algunas características del software orientado a objetos que generan problemas adicionales no cubiertos por las técnicas tradicionales de prueba.

Existe una referencia común acerca de que la unidad básica para la prueba de software orientado a objetos es la clase. A pesar de ello, cuando se prueba un desarrollo de software orientado a objetos, no es posible realizar una prueba

para una clase por sí misma, sino que hay que realizarla para una instancia de ésta, es decir para un objeto.

Una característica importante del enfoque orientado a objetos es la *encapsulación.* Un objeto encapsula su estado y sus funciones asociadas. La *abstracción,* concepto que define la capacidad de solo destacar las características esenciales de un objeto de forma tal que se puede separar su comportamiento esencial de su implementación, va estrechamente ligada con la encapsulación. Indudablemente la posibilidad de poder encerrar dentro de un contenedor físico o lógico elementos como arreglos, registros o incluso objetos parece ser muy atractiva para ciertas actividades, sin embargo en el contexto de prueba esto no es realmente una ventaja.

El ocultar todos los detalles del objeto que no contribuyen a sus características esenciales, por ejemplo su estructura y la implementación de sus métodos, hace que parte de un objeto sea inaccesible para el mundo. Naturalmente, esto obstaculiza la eficiencia de las pruebas, ya que para realizarlas, en algún momento, se requiere monitorear el estado de un objeto. Esto es difícil de realizar con características como la encapsulación y la abstracción, pues la dificultad de visualizar el estado interno del objeto impide consultar información que podría requerirse para el desarrollo de la prueba.

Por otra parte, la *herencia* es otra de las características que han venido a facilitar en gran medida el desarrollo de sistemas, la posibilidad de que una clase pueda ser escrita en términos de variaciones de otras clases es una ventaja significativa. Puede pensarse que esto apoya la prevención de fallas al construir software. Desgraciadamente, se ha comprobado que. mediante esta

práctica, se tienen muchas posibilidades de cometer errores, porque generalmente los elementos heredados son sometidos a algún tipo de refinamiento o redefinición y en algunos casos eliminación de componentes. Cuando un objeto es creado a partir de características heredadas, se genera un nuevo contexto que puede ser diferente al que pudiera crearse por la clase antecesora si esta trabajara de manera aislada y, por tanto, presentarse problemas ante situaciones no previstas en el nuevo contexto. Todas estas situaciones hacen que muchos autores afirmen que realizar una prueba a los métodos heredados debe ser una regla más que una excepción.

La herencia en cierta medida trae como consecuencia *reuso*, lo que genera una interrogante común con respecto a la formulación de las pruebas: las subclases de una clase que ya ha sido probada ¿Deben de ser probadas nuevamente?. Si la respuesta es sí, se habla de diferentes niveles de herencia lo que incrementa el número de pruebas a realizar. Así mismo, se afirma que el reuso no garantiza que el software esté exento de errores, ya que no es posible determinar si un número suficiente de rutas y estados hayan sido probados.

Otro aspecto que determina la dificultad de las pruebas que se realizan al software orientado a objetos es el *polimorfismo*. Cada vez que se realiza una instancia diferente de un objeto como producto del polimorfismo en los métodos, se requiere una prueba separada. Realizar una prueba separada para cada una de las formas de un método es una tarea difícil, la complejidad y el tiempo requerido crece considerablemente cuando se tienen que definir todos los posibles errores y obstáculos que pueden presentarse.

En los sistemas orientados a objetos, el flujo de control se lleva a cabo mediante el *paso de mensajes* entre objetos. Cuando un mensaje es enviado de un objeto u otro, la consecuencia es, generalmente, que el objeto receptor ejecute alguna operación para examinar o alterar su estado. El paso de mensajes es un punto fundamental al realizar la prueba.

Si bien estos son algunos de los aspectos más significativos y de referencia común en las distinta referencias bibliográficas, de igual forma pueden mencionarse la dependencia de estado, el desarrollo incremental, el manejo de excepciones, las invocaciones implícitas, el cambio de tipo, las peculiaridades del lenguaje o la generosidad.

2.6.3 MÉTODOS DE PRUEBA DE SOFTWARE ORIENTADO A OBJETOS

Se ha mencionado que muchas de las generalidades de los métodos de prueba tradicionales han sido adaptadas considerando las características del modelo orientado a objetos, con el propósito de que puedan ser aplicables en este nuevo contexto.

Actualmente, existen muy pocos trabajos producto del estudio de prueba de software orientado a objetos; de hecho, por sí mismo, el área de prueba de software es bastante compleja y dentro de este marco de objetos existe una carencia de métodos robustos para garantizar la realización de las pruebas de forma eficaz.

A continuación se presenta un panorama del estado actual en cuanto a prueba de software orientado a objetos:

2.6.3.1 Pruebas de Unidad

En el software orientado a objetos la menor unidad a considerar para realizar una prueba es la clase. La prueba de clases en el ámbito de software orientado a objeto es equivalente a la prueba de unidad realizada al software tradicional. Esta prueba está fundamentalmente dirigida a las operaciones encapsuladas por la clase, así como al estado y comportamiento del objeto que se implementa en ella. El énfasis de la prueba de unidad es verificar que esta pequeña unidad trabaje correctamente en forma aislada, antes de proceder a integrarla en el sistema.

Los métodos contenidos en una clase pueden ser muchos y una operación en particular de ese conjunto, a consecuencia de la herencia, puede existir como parte de varias clases diferentes. Por lo tanto el significado de prueba de unidad cambia en muchos sentidos y es importante diseñarla bajo ciertas consideraciones.

Actualmente, no existe una práctica única de cómo realizar pruebas de unidad al software orientado a objetos, pues el construir casos de prueba para este tipo de sistemas no es una tarea sencilla. Además existen muchas posibilidades de que estos casos caduquen muy rápido, por lo que resulta imposible reutilizarlos muchas veces. Las pruebas de unidad se pueden llevar a cabo considerando aspectos como:

- el orden en que los métodos son sometidos a la prueba,
- el orden en que una jerarquía de clases puede ser probada,

- el ejercicio del flujo de datos, y
- el análisis del estado del objeto.

Pero ¿Qué aspectos deben considerarse para construir casos de prueba para una clase?. Se puede verificar que ésta proporcione los servicios que ofrece, que responda correctamente a las condiciones esperadas y, más aún, ante las inesperadas. Aspectos adicionales pueden verificar si la clase contiene y permite disponer de todas las funciones asociadas a ella o que cada método de la clase ejecute su responsabilidad especificada.

Estas son algunas de las técnicas más populares para realizar pruebas de unidad:

2.6.3.1.1 Pruebas Estructurales

Si se tiene la disponibilidad de código fuente, pueden realizarse pruebas estructurales a las unidades sometidas a la prueba. Las acciones de esta actividad pueden diseñarse con el propósito de verificar todas las rutas del código, las condiciones establecidas o bien los ciclos definidos en el programa.

2.6.3.1.2 Prueba de Valores Límite

Mediante esta técnica se prueba la unidad bajo situaciones inusuales o extremas, con el propósito verificar cómo son manejadas por el software. Para ello, los casos de prueba suministrados son diseñados considerando valores

frontera, es decir los valores mínimo y máximo que la unidad puede aceptar, así como también aquellos valores cercanos a las fronteras identificadas.

2.6.3.1.3 Prueba Basada en Estados

Para esta técnica, se generarán casos de prueba para un contexto en donde una clase se modela como una máquina de estados con secuencias de transiciones, con esto se pretende analizar el estado de los objetos de acuerdo a su comportamiento. Una vez que se ha establecido un modelo de estados con base en los atributos del objeto, se deben considerar en la prueba los métodos necesarios para poder observar los cambios de estado. La aplicación de esta técnica permite observar alguna de las siguientes situaciones:

- se produce un cambio a un estado correcto,
- se produce cambio a un estado incorrecto,
- no hay cambio de estado,
- se produce un estado indefinido correcto, y
- se produce un estado indefinido incorrecto.

2.6.3.1.4 Prueba Incremental

La prueba incremental dirige su atención a las subclases generadas como consecuencia de la herencia, siendo la clase padre una clase previamente probada. Aunque existen situaciones en las que éste tipo de pruebas se descarta, se pueden identificar algunas en las que no estarían de más:

- cuando se han agregado o modificado propiedades y / o métodos,
- cuando existen propiedades y métodos que se han heredado y no se han alterado, pero que realizan algún tipo de interacción con elementos nuevos o modificados.

2.6.3.2 Pruebas de Integración

Cuando se aplican pruebas de integración al software orientado a objetos, se pretende demostrar que las unidades que ya han sido sometidas a un proceso de prueba y funcionan correctamente, lo hacen de igual forma cuando interactúan y se integran con otras unidades del sistema. Prácticamente, el trabajo de esta prueba se concentra en la interacción de métodos en diferentes unidades.

Existe una coincidencia en los dos enfoques para realizar este tipo de pruebas: el basado en hilos de ejecución y el basado en uso. En el primero, pretende que todas las clases respondan a sencillas entradas externas, provenientes de otra unidad. De esta forma, se realizan casos de prueba para cada clase en la unidad, con lo cual un hilo de ejecución de este conjunto se prueba.

En el enfoque basado en uso, se realizan pruebas para clases las cuales usan servicios de otras clases.

A continuación se presentan algunos métodos para realizar pruebas de integración:

2.6.3.2.1 Método de Caminos de Mensajes

Este método se concentra principalmente en probar aquellos caminos que se generan por un evento de entrada y terminan con un evento de salida.

2.6.3.2.2 Método de Overbek

En este método se prueban las clases por pares, donde una hace el papel de cliente y otra el de servidor, estableciéndose para éstas dos conjuntos de pruebas. El primer conjunto, son pruebas orientadas a verificar si los mensajes de entrada y de salida generados son correctos; es decir si se usa correctamente cualquier clase servidora y si todas las secuencias de operaciones son correctas. En el segundo conjunto se verifica además de lo anterior, si la clase cliente siempre satisface las precondiciones de la clase servidora, así como también si satisface las salidas esperadas por la clase servidora.

2.6.3.2.3 Método de Kung

Este método emplea una estrategia de ingeniería en reversa sobre el código de las unidades con el propósito de generar un diagrama de relaciones entre objetos. A partir de este diagrama se propone un orden para las pruebas que minimiza el uso de cabos. El diagrama se convierte en un grafo acíclico, que puede contener varios grupos (clusters) de objetos y los ordenan topológicamente. Su método involucra las etapas de pruebas de unidad y de integración y puede usarse también para pruebas de regresión.

2.6.3.3 Pruebas de Sistema

Las pruebas de unidad se concentran en verificar si las funcionalidades descritas en las especificaciones o en los requisitos iniciales corresponden a las que se presentan en el producto final. En esta área, al igual que la de pruebas de integración, se han generado pocos trabajos, por lo que se emplean muchos de los métodos tradicionales.

2.6.3.3.1 Prueba de Función

La prueba de función se orienta a confirmar que la aplicación alcanza los requerimientos y la funcionalidad especificadas por el usuario.

2.6.3.3.2 Pruebas de Aceptación (beta)

En este tipo de pruebas, las versiones que aún no han sido liberadas en el mercado, son ofrecidas a ciertos grupos de usuarios con el propósito de que las utilicen. El propósito de esto es que los usuarios reporten defectos que pudieran presentarse.

2.6.3.3.3 Prueba Bajo Stress

Para realizar esta prueba, el sistema se somete a condiciones extremas de trabajo, como pueden ser un alto volumen de transacciones o un gran número de usuarios.

Aplicando este enfoque, se puede verificar si el sistema se comporta como se espera aún ante este tipo de escenarios.

2.6.3.4 Patrones de Diseño de Pruebas

El concepto de *patrones de diseño* consiste en identificar problemas típicos y sus soluciones adecuadas mediante *patrones:* cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de su solución, de forma que puede utilizarse continuamente sin hacer dos veces lo mismo.

Este concepto se empezó a aplicar en el contexto de la Ingeniería de Software y más concretamente en el enfoque de programación orientada a objetos, buscando soluciones generales y por tanto reutilizables a problemas similares. Se trata de reutilizar las *ideas* pensadas por una persona para solucionar un problema, llevando el concepto de reutilización a un nivel de abstracción mayor que el de la mera reutilización del código de programación: al nivel del diseño orientado a objetos.

Un patrón de diseño se puede definir como "una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular".

En el diseño de software se han clasificado gran cantidad de patrones según su utilidad:

- Patrones sobre creación: sobre cómo crear instancias de objetos.

- Patrones estructurales: sobre cómo agrupar y organizar objetos.
- Patrones de comportamiento: sobre cómo se relacionan los objetos entre sí en tiempo de ejecución.

Otra de las ventajas del uso de patrones de diseño además de la reutilización es en la ayuda en la documentación de un sistema.

Los problemas principales del trabajo con patrones de diseño es que debido a su grado de abstracción y generalidad muchos patrones no son comprensibles y que las soluciones a veces no son evidentes.

2.6.3.4.1 Diseño de Pruebas Basado en Patrones

Una plantilla de patrones de diseño de pruebas debe contener entre otras cosas:

- Nombre: una palabra o frase que identifica el patrón y sugiere su enfoque general.
- Objetivo: ¿Qué problema de diseño de pruebas resuelve este patrón?, ¿Cuál es la estrategia de pruebas? Dar una descripción muy breve.
- Contexto: ¿Bajo qué circunstancias se aplica este patrón?, ¿A qué clase de entidades software?, ¿A qué alcance? Esta sección se corresponde con las de "motivación", "aplicabilidad" de los patrones de diseño.
- Modelo de Fallo: ¿Qué clase de defectos busca este patrón? El fallo debe alcanzarse a partir de los datos de entrada de la prueba y el estado del sistema, debe producir resultados incorrectos y ha de ser propagado

a la salida de tal forma que sea observable por la persona que realiza la prueba.

- Estrategia: esta sección aclara como se ha de generar e implementar la colección de casos de prueba. Tiene cuatro subsecciones obligatorias:
  - Modelo de pruebas: define una representación de las responsabilidades y / o implementación que son el objetivo del diseño de pruebas.
  - Procedimiento de prueba: define un algoritmo, técnica o heurística por la que se generan los casos de prueba del modelo.
  - Comprobación: define el algoritmo, la técnica o heurística por la cual los resultados actuales de un caso de prueba se evalúan como pasado con éxito / no pasado.
  - Automatización: discute los enfoques automáticos para la generación de casos de prueba, ejecución de pruebas y evaluación de la ejecución de las pruebas. Se suele mostrar con ejemplos.
- Criterio de entrada: lista de precondiciones para una prueba efectiva y eficiente con este patrón.
- Criterio de salida: criterio objetivo que ha de alcanzarse para dar por completa la prueba con este patrón.
- Consecuencias: prerrequisitos generales, costes, beneficios, riesgos y consideraciones en el uso de este patrón.

- Usos conocidos: ¿Cuáles son los usos conocidos de este patrón de diseño?, ¿Cuáles son los usos conocidos de los modelos de prueba y estrategias incorporadas en este patrón?, ¿Cuáles son la eficiencia y la efectividad de este patrón o estrategias similares establecidas por estudios empíricos?
- Patrones relacionados: patrones de diseño similares o complementarios.

Esta plantilla de patrones de diseño de pruebas se centra en los aspectos fundamentales del diseño de pruebas:

- Elección de la estrategia de pruebas apropiada al caso particular.
- Tipos de defectos que se buscan en la fase de pruebas.
- Cómo construir una colección de casos de prueba.
- Automatización.

2.6.3.4.2 Uso de Patrones de Diseño de Pruebas

El uso de patrones de diseño de pruebas no modifica sustancialmente las tareas básicas de los procesos de prueba. Los pasos básicos para utilizar un patrón de pruebas son los siguientes:

I. En las fases iniciales del proceso de desarrollo, seleccionar los patrones de pruebas que se correspondan con el campo de aplicación y la estructura del sistema en desarrollo y sus partes.

II. Desarrollar el modelo de pruebas para la implementación a probar.

III. Instrumentar el trabajo de desarrollo y prueba de tal manera que la colección de casos de prueba sea aplicada a una implementación que haya alcanzado el umbral mínimo de operatividad tal y como se haya establecido en el criterio de entrada.

IV. Generar la colección de casos de prueba aplicando el procedimiento de pruebas al modelo de pruebas.

V. Desarrollar la implementación de la automatización de las pruebas.

VI. Ejecutar y evaluar las pruebas. Si no se alcanza la cobertura recomendada, revisar la colección de casos de prueba.

2.7 PRUEBA DE SOFTWARE BASADO EN COMPONENTES

La construcción de software a partir de componentes es una práctica relativamente nueva, por lo que no es extraño que sea escasa la existencia de trabajos generados al respecto.

En términos formales un componente es una pieza de software que cumple con dos características:

- no depende de la aplicación que la utiliza, y
- se puede emplear en diversas aplicaciones.

Puesto que el desarrollo basado en componentes presenta algunas similitudes con el enfoque orientado a objetos, para un componente pueden ser aplicables algunas de sus consideraciones, incluso en materia de prueba.

Es evidente que para hacer las pruebas de componentes más robustas será necesario considerar las características propias del enfoque de componentes.

En la mayoría de los casos, los criterios de prueba de caja negra son los más aplicados a los componentes, puesto que la disponibilidad del código fuente es nula en la mayoría de las veces. Debido a que un componente es una unidad concreta, con una función bien definida, no basta realizar pruebas para su evaluación; de igual forma se requieren procesos de prueba para su selección y para su integración.

Durante la etapa de construcción de un componente, el equipo de testing puede aplicar las técnicas de prueba de unidad y de integración tradicionales del modelo orientado a objetos, sin embargo en lo que respecta a la selección y evaluación, considerar el punto de vista del usuario es un aspecto vital para la realización de la prueba. Finalmente en el marco de pruebas de integración, consideraciones como la arquitectura de la aplicación, el software intermediario y los modelos de los componentes, deben agregarse a los criterios de evaluación.

2.7.1 MÉTODOS DE PRUEBA DE SOFTWARE BASADO EN COMPONENTES

Con el propósito de organizar algunas de las estrategias de prueba de componentes más comunes, se presenta una descripción de las mismas en forma similar a los términos que se presentaron para el enfoque orientado a objetos, a nivel de unidad y nivel de integración.

2.7.1.1 Pruebas de Unidad

Aunque la realización de pruebas de unidad es una actividad que en algún momento es llevada a cabo por el desarrollador, existe un marco de trabajo adicional a considerar: el de la persona que se interesa en el componente con el fin de integrarlo en sus sistemas. No por esto el equipo de testing debe descartarla, sino más bien debe probar el componente de manera tal que este realice correctamente las funciones específicas para las cuales fue construido.

Las pruebas que se aplican a los componentes se construyen bajo la combinación de varios enfoques como diseño por contrato, como pruebas matemáticas, métricas, validación exhaustiva en proyectos prácticos y manejo de cambios rigurosos.

2.7.1.2 Pruebas de Integración

Si las pruebas de nivel de unidad para componentes muestran severas carencias, en el nivel de integración, al igual que en otros enfoques de desarrollo, las carencias son aún más notables. Sin embargo, existen coincidencias en cuanto a las problemáticas comunes al integrar componentes.

2.8 TESTING DE SISTEMAS WEB

El proceso de prueba de un sistema web presenta algunos aspectos diferentes a considerar y se expondrán en esta sección.

Antes de que la aplicación web sea puesta en producción, se requiere un proceso de prueba riguroso donde es importante incluir pruebas de seguridad, de confiabilidad de hardware y software y de compatibilidad entre todos los elementos del sistema.

La puesta en producción de sistemas web luego de una etapa de testing pobre puede acarrear algunos de los siguientes problemas:

- Errores en el sistema de transacciones debido a que las funcionalidades de transacciones del sistema pueden no haber sido probadas.
- Pérdida de datos críticos almacenados en la base de datos que puede resultar en un pobre, o inexistente, seguimiento de transacciones.
- Transacciones inseguras que pueden provocar pérdidas a los usuarios y a la organización en general.
- Caídas del sistema, si no se le ha hecho una prueba de carga adecuada. Este tipo de errores provocan pérdidas debido al tiempo de restauración del servicio. La reparación de estos errores aumenta los costos en forma directa e indirecta.

2.8.1 EQUIPO DE TESTING

El equipo de testing o equipo de prueba de software debe ser cuidadosamente seleccionado. La experiencia de los integrantes de este equipo es de gran importancia. Un equipo de novatos en ejecución de pruebas puede no cubrir todas las alternativas de error que presenta cada sistema.

Cada miembro del equipo debe estar familiarizado con la tecnología, poseer conocimiento del negocio, de los requerimientos del cliente y cómo se interrelaciona la tecnología. Por ejemplo, para el testing de sistemas de tres capas, los miembros del equipo deberían estar familiarizados con cada una de ellas (ver Figura 3).

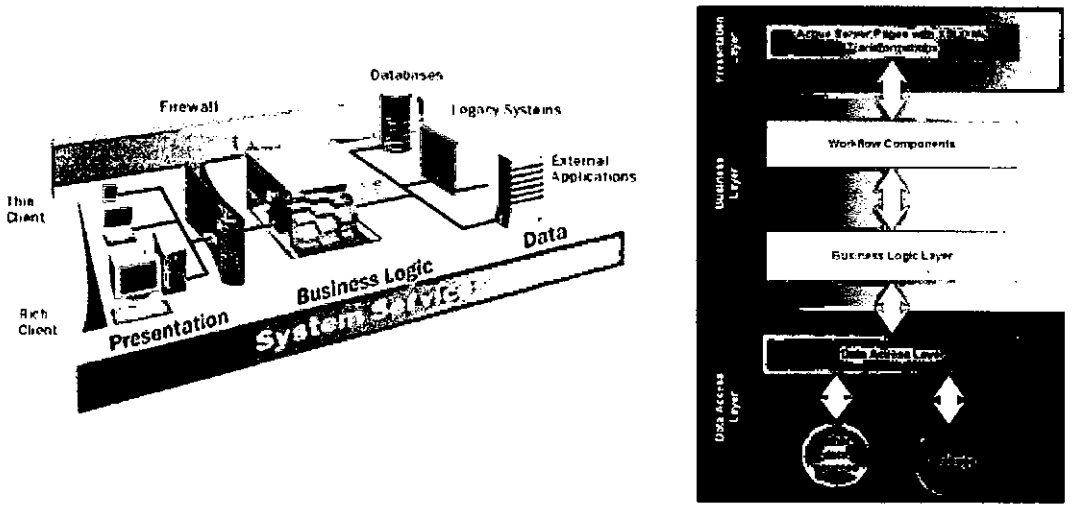


Figura 3 – Capas de las Aplicaciones WEB

De este modo, para hacer pruebas sobre la capa de presentación (*presentation layer* o *web tier*), el equipo de testing debe comprender el funcionamiento de los principales navegadores (*browsers*), tal como el Internet Explorer, Netscape, AOL, etc. así como entender ASP, HTML, DHTML, Java y VBScript debido a su relevancia.

El testing en la capa intermedia (*business layer* o *middle layer*) requiere conocimiento de la lógica del negocio. Esto incluye el servidor de aplicaciones web utilizado, mecanismos y funcionalidades utilizados.

Para realizar el testing de la capa de datos (*data access layer* o *data tier)* se debe estar familiarizado con el administrador de bases de datos utilizado, así como de sus comandos y sistemas de almacenamiento y archivo.

2.8.2 PRUEBA DEL SISTEMA

En la etapa de testing de un sistema, debe considerarse una serie de pautas a cumplir.

Al principio del proceso de testing se debe planificar la cantidad de pruebas a realizar al sistema. Esta planificación debe estar basada en suposiciones acerca de la calidad actual del sistema a probar. Frecuentemente, el proceso de testing termina cuando el tiempo que se estableció para el mismo concluye; sin embargo, un plan de testing eficiente debería permitir terminar el proceso cuando no se han encontrado más errores en las pruebas existentes o cuando la cantidad de errores encontrados es mayor que el umbral de aceptación.

El uso de estándares adecuados ayuda a tener una buena documentación y guía el proceso en general.

El proceso de testing se realiza en niveles o etapas. Para cada uno de estos niveles deben prepararse especificaciones detalladas, o sea, un plan de prueba y sus correspondientes casos de prueba. A esto se le debe agregar los datos de entrada y el registro de su ejecución en la documentación.

Los ambientes de prueba especifican las precondiciones requeridas para llevar a cabo cada uno de los casos de prueba. Éstos también deben estar documentados y definidos en forma adecuada antes de realizar la prueba.

La **cobertura del testing** es el grado en el que los casos de prueba específicos abordan los requerimientos del sistema o componente. Especificar la cobertura que se intenta lograr como parte del plan de pruebas.

Los datos a utilizar en cada uno de los casos de prueba tienen particular importancia. Se deben generar u obtener datos que permitan la ejecución de los mismos, simulando tanto la actividad normal del sistema como situaciones específicas que requieran ser probadas. Considerar que no se deben utilizar datos confidenciales para realizar las pruebas sin previamente tener autorización escrita, especialmente en el caso de testing de aceptación[1].

Los casos de prueba deben ser diseñados para ayudar a probar las restricciones del sistema, también llamado testing de stress, tal como tamaños de archivo y de bases de datos.

Una vez ejecutadas las diversas pruebas y para asegurar que no se han introducido nuevas anomalías en el sistema, producto de los cambios realizados, luego de la detección de errores se debe realizar testing de regresión sobre cada componente del sistema (ver 2.8.4.4.3).

2.8.3 DESAFÍOS EN EL TESTING DE SISTEMAS WEB

El testing de sistema WEB no es sencillo. Con los rápidos cambios en tecnologías, tanto en hardware como software, el equipo de prueba encuentra

[1] Esto se debe a que esta prueba la realizan los usuarios finales u operadores del sistema. Si se utilizan datos confidenciales, éstos serán accedidos por los encargados del testing de aceptación.

dificultades en conseguir herramientas o técnicas estándar para este tipo de aplicaciones.

- Los cambios rápidos de la tecnología hacen que se vuelque al mercado nuevo hardware y software que proporcionan más funcionalidades, cambiando las necesidades de testing. Esto provoca que los miembros del equipo de prueba deban crear cada vez nuevos ambientes de prueba.
- Los perfiles de usuario o visitantes de un sitio pueden variar desde principiantes a clientes sofisticados. El equipo de prueba necesita simular las posibles acciones de todo tipo de clientes.
- Se deben crear constantemente ambientes de prueba debido a la complejidad que pueden tener las aplicaciones y sus interacciones con el mundo. Estos incluyen la configuración del cluster WEB, componentes de la capa del negocio, servidores de base de datos, firewalls[2], etc. El sistema, incluso, puede tener interacción con compañías de tarjetas de crédito, consultas a servicios WEB, etc. Así, la simulación de cada posible acción que realice un cliente en línea es imposible debido a que el encargado de la prueba no puede predecir las acciones de un visitante del sitio. La aplicación web puede ser probada sobre ciertas funciones estándar pero no puede ser probada sobre todas las contingencias posibles debido a que el futuro del sitio es desconocido.

[2] Sistema diseñado para prevenir el acceso no autorizado de intrusos a los sistemas principales ya sea desde una red privada o Internet.

- La magnitud del tráfico en un sitio puede ser desconocida al momento de la fase de prueba. Un sitio puede haber sido diseñado para un cierto grupo de usuarios y, dadas las posibilidades provistas por Internet, convertirse luego en un sitio muy concurrido con demandas inesperadas por los desarrolladores.

- Los hackers no poseen métodos estándar para atacar sitios WEB (y las aplicaciones existentes en ellos). Tampoco hay métodos estándar para realizar las pruebas de seguridad del sitio. Lo más común es que se utilicen herramientas para hacer pruebas estándar de aspectos de seguridad (ver 2.8.4.4.1).

2.8.4 TIPOS DE TESTING PARA APLICACIONES WEB

El proceso de testing de una aplicación web se extiende a las tres capas de la arquitectura del sistema (ver Figura 4).

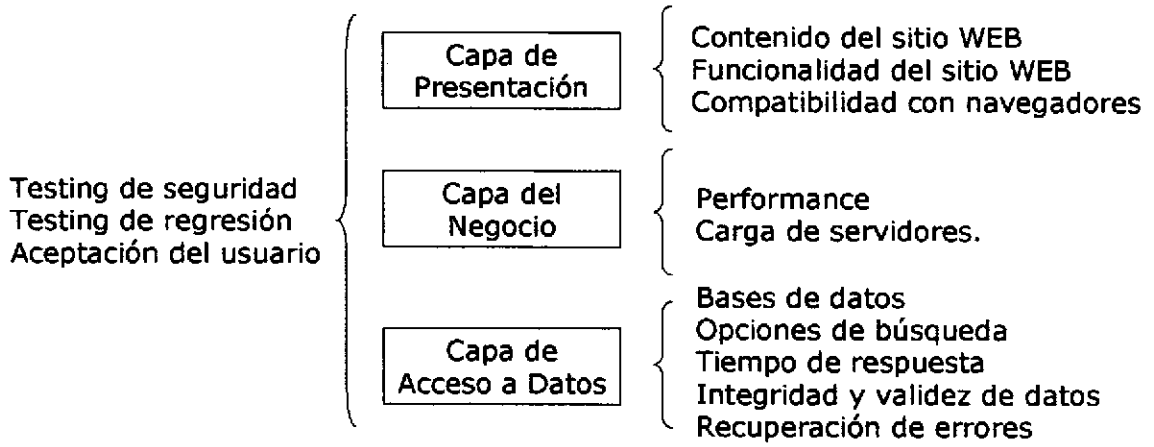


Figura 4 – Tipos de Testing de Aplicaciones WEB

2.8.4.1 Testing de la Capa de Presentación

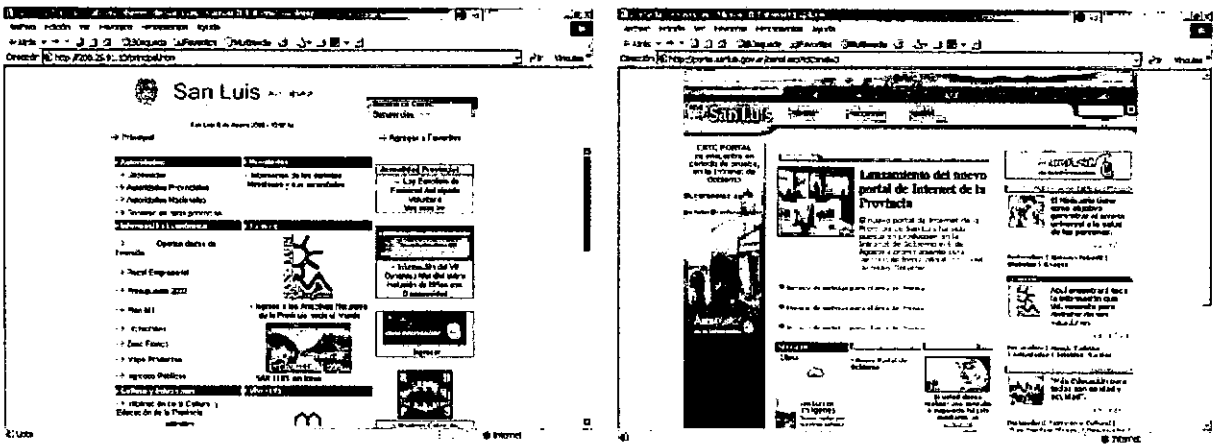


Figura 5 – Diferentes Diseños del Sitio Web

2.8.4.1.1 Contenido del Sitio Web

La prueba de contenidos es importante para poder presentar un sitio de calidad y libre de errores. Antes de que la aplicación web sea finalmente presentada a los clientes en línea se debe seguir un proceso de planificación del sitio, realización de las pruebas de detección de errores y haber hecho la corrección de los mismos.

La apariencia visual del sitio WEB es importante. Más allá de la elección del color, fuente o gráficos, las pruebas deben validar la apariencia del sitio y detectar áreas problemáticas (pantallas, páginas de catálogos, etc.).

2.8.4.1.1.1 Testing de GUI

GUI la sigla de *Graphic User Interface* o, en español, Interfaz Gráfica de Usuario. La GUI es la interfaz visual que permite al usuario utilizar sus

programas a través de menús e iconos representativos, haciendo uso de dispositivos de hardware que facilitan las operaciones y apelan a la intuición del usuario. En la GUI se debe evaluar el uso de fuentes, colores (intensidad, combinación, etc.), gráficos, audio y video, etc.

Deben realizarse pruebas de compatibilidad de estilos de fuentes debido a que no todas las variedades existentes en editores de HTML pueden ser mostradas por todos los navegadores, especialmente aquellos de versiones viejas. Textos escritos con estas fuentes pueden ser mostrados con caracteres ilegibles en algunas versiones de navegadores.

Es recomendable que las diferentes páginas del sitio tengan un estándar visual, lo normal es tener títulos o encabezados con tamaño de fuente entre 18 y 24 y texto normal entre 10 y 14. Considerar la combinación de colores del primer plano y del fondo a lo largo del sitio (por ejemplo, puede resultar difícil leer texto color amarillo sobre fondo blanco).

El tiempo de carga de una página depende, en gran medida, del contenido gráfico que posea. Se recomienda que, en donde sea posible, se utilicen imágenes pequeñas (thumbnails) en lugar de fotografías completas. Los desarrolladores deben evaluar tiempo de descarga de las páginas con alto contenido gráfico.

Existen sitios en Internet, como http://www.sitecritique.net que asesoran en la creación de diseños de página.

A continuación se presenta una lista, no exhaustiva, de pruebas de interfaz de usuario:

- Probar cada barra de herramientas, ítem de menú, navegación de las ventanas para comandos utilizando mouse y teclado.
- Probar que las máscaras de formato sean las adecuadas. Por ejemplo, todos los campos drop-down (combo) deben estar ordenados de forma adecuada. El ingreso de fechas debe estar formateado acorde a la región; el usuario debe identificar fácilmente el formato de fecha a utilizar.
- Ver que los colores, fuentes y tamaño de fuente sean estándares y coherentes a lo largo de la aplicación.
- Asegurarse que las barras de desplazamiento horizontal o vertical no aparezcan, a menos que sea necesario.
- Ver que los diferentes objetos de la pantalla estén correctamente alineados.
- Controlar la redimensión de las ventanas; hay ventanas cuyo tamaño debe ser constante.
- Verificar que todos los campos alfanuméricos tengan alineación a la izquierda y que los campos numéricos estén alineados a la derecha.
- Evaluar las opciones que aparecen por omisión.
- Probar que cada comando de menú tenga una secuencia de teclas alternativas (hot keys), que funcionen correctamente y no estén duplicadas.

- Probar el orden de tabulación. Debe ser desde arriba a la izquierda hacia abajo a la derecha. También chequear que los objetos de sólo lectura, deshabilitados y ocultos deben ser salteados en la secuencia de tabulación y que el cursor se posicione en el primer campo de entrada cuando se abra la ventana.
- Asegurarse que funcionen correctamente los botones definidos por omisión y que los botones de comando no aparezcan (o estén deshabilitados) cuando no estén en uso.
- Asegurarse que las ventanas se activen como "modal". Esto previene que el usuario acceda otras funciones cuando se activa la ventana.
- Chequear que puedan abrirse múltiples instancias de la aplicación al mismo tiempo.
- Asegurarse que exista un menú de ayuda.

2.8.4.1.1.2Uso de imágenes

Las imágenes incrustadas en páginas web son en gran parte responsable de los cuellos de botella existentes en sitios WEB.

El proceso de prueba, en este punto, debe ser suficientemente crítico como para conseguir un compromiso aceptable entre el número y tamaño de las imágenes y la velocidad de acceso al sitio.

Los dos formatos de imagen que utilizan los navegadores son GIF y JPEG. El primero es adecuado para imágenes o gráficos sencillos y pocos niveles de

color. El segundo, en cambio, es el óptimo para imágenes complejas o fotografías (con millones de colores). Utilizando el formato adecuado para cada tipo de imagen se optimiza la velocidad de acceso.

Chequear que se incluya en el HTML en el tag <IMG> los parámetros WIDTH y HEIGHT, es decir, ancho y alto de la imagen. HTML toma por defecto las dimensiones reales de la imagen, pero si se incluyen estos valores, la página HTML se presentará mucho más rápido en pantalla (el texto), mientras se acaban de transmitir las imágenes.

Chequear que, en donde sea posible, se utilicen animaciones con imágenes sin recurrir a complicados applets Java o cualquier otro sistema. Los GIFs animados están soportados por la gran mayoría de navegadores (ver 2.8.4.1.3). El uso de animaciones debe ser moderado para no marear al visitante. En el caso que el navegador no soporte los GIFs animados se presentará en pantalla el GIF inicial de la animación.

2.8.4.1.1.3<u>Formularios</u>

Las aplicaciones WEB requieren con frecuencia que el usuario ingrese información. Los campos de formularios creados con HTML también deben tener controles de contenidos de datos, como por ejemplo, no permitir un ingreso de edad inferior a 10 ni superior a otro valor.

2.8.4.1.1.4 Legibilidad

La legibilidad engloba aspectos diferentes (lo que en inglés se denomina *legibility* y *readability*).

- Legibilidad a nivel de caracteres: Se debe a las características del diseño de la fuente y depende de propiedades tales como la altura del caracter "x", los espacios interiores, la optimización para la pantalla (hinting) y otros detalles.
- Legibilidad y aspecto de las fuentes en pantalla:
    - Antialiasing: es el suavizado de los contornos de los tipos (o de los objetos) que se puede conseguir en la mayoría de programas de gráficos. Tiene la ventaja de hacer los contornos de las letras más suaves, de modo que no quedan tan recortados sobre el fondo. Como regla general, es totalmente recomendable suavizar los tamaños mayores, no así tamaños muy reducidos debido a que puede perjudicar el aspecto de muchas fuentes en pantalla (apenas quedan píxeles para difuminar).
    - Fuentes optimizadas para la pantalla: para tamaños reducidos, muchas tipografías tienen un aspecto que deja mucho que desear. Existen, sin embargo, fuentes especialmente preparadas para su visualización en pantalla, incluso a tamaños reducidos. Este tipo de fuentes es recomendable para el uso normal de texto en la WEB y para textos especiales de reducidas dimensiones.

- Legibilidad de bloques de texto: La facilidad con que se puede leer el cuerpo de texto de un documento, depende sobre todo de una correcta composición:
    - Ancho de las líneas.
    - Tamaño del texto proporcionado al tipo de documento.
    - Correcto interlineado.
    - Separación adecuada entre párrafos (con un espacio extra, con sangría de la primera línea).
    - Relación clara entre subtítulos y el párrafo que va asociado a éstos.

Las especificaciones de los estándares CSS (*Cascade Style Sheet* u hojas de estilo en cascada) permiten un fino control de los atributos tipográficos del texto. El problema principal está en la falta de compatibilidad con navegadores anteriores (versiones 3 o previas) y en el deficiente (o muy deficiente) soporte por parte de los diferentes navegadores (ver 2.8.4.1.3).

2.8.4.1.1.5 Ortografía y Gramática

Son inaceptables los errores ortográficos a lo largo de la aplicación. Utilizar una herramienta de chequeo de ortografía, teniendo en cuenta que, muchas veces, sus resultados no proveen una confiabilidad total. Se recomienda hacer una lectura completa y de corrido del texto para detectar errores gramaticales y asegurar legibilidad. Es frecuente encontrar errores gramaticales relacionados

con pronombres relativos, conjunciones y pronombres interrogativos (que / qué, como / cómo, etc.).

El equipo de testing debe controlar la escritura de todo texto mostrado en las ventanas: título, opciones en la barra de estado, campos de texto, textos emergentes y mensajes de error.

2.8.4.1.1.6Contenido

Verificar todas las afirmaciones, imágenes y relaciones entre éstas que se encuentren presentes en el sitio WEB con las áreas legal, marketing y negocio.

2.8.4.1.2 Testing de Funcionalidad del Sitio WEB

Los hipervínculos en sitios WEB deben ser probados debido a que pueden estar rotos, ausentes o impropiamente asignados.

Los vínculos rotos pueden ser producto de descuidos del desarrollador, tal vez porque en forma inadvertida, cambió de directorio un archivo o página o modificó su nombre. Por ejemplo, si el desarrollador cambia el nombre de la página de productos de "Producto.htm" a "Productos.htm" se romperán los vínculos y esta página quedará inaccesible.

Los vínculos ausentes son aquellos que están especificados pero aún no han sido creados.

Los vínculos incorrectos causan errores, trayendo al usuario la página errónea o ejecutando una acción incorrecta o no deseada. Por ejemplo, a través de un

vínculo erróneo, puede realizar acciones catastróficas; por ejemplo un intercambio de vínculos entre las opciones "aceptar" y "cancelar".

En los tres casos, el usuario no puede acceder a la página apropiada; más aún, en una aplicación WEB el usuario puede realizar acciones inesperadas y que provoquen serios problemas.

En el parámetro HREF del tag <A>, para crear los hipervínculos, se debe especificar el URL o dirección a la cual apunta el enlace (dirección y archivo en el caso de las imágenes). Estas direcciones pueden ser absolutas o relativas. En el caso de las direcciones absolutas se especifica la dirección completa a la que apunta el enlace, por ejemplo:

```
<A HREF=http://www.rotativo.com/timagazine/test/mi_pagina.html>
Enlace </a>
```

Las pruebas sobre el sitio deben verificar que, por ejemplo, si el enlace apunta a una página dentro del mismo servidor, se deben utilizar direcciones relativas, es decir, sin hacer referencia a la dirección Internet. En el ejemplo anterior sería:

```
<A HREF="../test2/ejemplo.html"> Enlace 2 </a>
```

En el ejemplo anterior las palabras "Enlace 2" constituyen un enlace a la página ejemplo.html, situada en el directorio test2, al mismo nivel en el árbol de directorios que la página donde se encuentra el enlace. Para ascender niveles en una estructura de árbol de directorios o carpetas se utilizan los dos puntos.

Las direcciones absolutas y relativas se pueden aplicar del mismo modo en el caso de las imágenes.

Comprobar que no se utilicen direcciones absolutas de imágenes residentes en cualquier servidor de Internet. Cualquier modificación en éste repercutirá de inmediato en el sitio.

Existen herramientas que ayudan a chequear vínculos rotos o incorrectos, sin embargo, los vínculos ausentes deben ser chequeados en forma manual. En los siguientes sitios pueden hacerse pruebas en línea de vínculos rotos:

- http://netmechanic.com
- http://www.linkalarm.com/index.html
- http://www.Websitegarage.com

2.8.4.1.3 Compatibilidad con Navegadores

Un visitante puede acceder a un sitio WEB con uno de los numerosos navegadores, con versión moderna o vieja, con resolución de pantalla y color variada, con una u otra demanda del sistema, etc.

El navegador es el encargado de interpretar el código HTML. Por tanto, la apariencia final del documento dependerá directamente de la interpretación que realice el nevegador.

Una de las mejores características del lenguaje HTML es que es compatible bidireccionalmente; esto quiere decir que, un navegador antiguo, ante tags HTML de una nueva especificación que no sepa interpretar, no producirá un

error, sino que simplemente omitirá esa información, compatibilidad hacia delante[3]. En cambio, el navegador de última generación siempre será capaz de interpretar todas las especificaciones HTML anteriores a él, compatibilidad hacia atrás.

Otro grave problema es el de la utilización de tags HTML propietarios. Los dos navegadores que abarcan la mayor parte del mercado son Netscape Navigator y Microsoft Internet Explorer. Cada uno de ellos ha ido introduciendo tags que no son interpretables por el producto competidor, lo que provoca, en principio, un problema. Por regla general, los nuevos tags que introduce una empresa en su navegador son introducidos en la siguiente especificación de HTML que dicta el W3C (organismo que vela por la estandarización del HTML).

Otro aspecto a tener en cuenta es la plataforma o sistema operativo sobre la cual se ejecuta el navegado que puede influir sobre la apariencia final del documento. Un ejemplo claro son las fuentes. Si se utiliza el tag <FONT> con el parámetro FACE (tipografía) y se elige un tipo de fuente que no está instalado por omisión en el sistema operativo de otra plataforma, la apariencia variará. En este caso particular lo más aconsejable es asignar una lista de fuentes al parámetro FACE, separadas por comas. Así, si no se encuentra disponible una de ellas, se buscará una alternativa de la lista. También es importante considerar que no todos los usuarios trabajan con la misma resolución. Una web fantástica a 800x600 de resolución puede deformarse enormemente a 640x480.

[3] Esta característica no es muy frecuente en el mundo del software.

En el caso de Internet Explorer existen versiones: para plataformas Windows (3.x, 9x, XP, 2000, NT), Mac OS e incluso se espera una versión para Unix. El caso contrario es Netscape Communicator con versiones para casi cualquier plataforma (Windows, Mac OS, OS/2, Unix). Exceptuando el caso de OS/2, Netscape Communicator está actualizado a la última versión en el resto de plataformas.

Por lo tanto, a la hora probar un sitio WEB, se debe establecer un compromiso entre las especificaciones HTML que se utilicen y la compatibilidad con navegadores más antiguos o de otra empresa. Se debe pensar que no todo el mundo está siempre con la última versión de sus navegadores.

El testing de compatibilidad debe garantizar que la página se vea bien en el mayor número de navegadores posible. La compatibilidad con los navegadores más usados es obligatoria; la compatibilidad con navegadores menos usados y modelos arcaicos también será de utilidad. Cada página debe verse bien en un navegador de texto.

Sin embargo, debido a los diferentes estándares, modelos de navegadores y diversidad de sistemas, no es posible crear una página que sea compatible con todos los sistemas de navegación. Pero el sitio debe ser respetuoso hacia todos sus visitantes y explicarles por qué tienen problemas y cómo pueden resolverlos.

Los analistas y desarrolladores deben determinar el grado de compatibilidad del sitio con la mayoría de los navegadores populares, tal como Internet Explorer, Netscape, AOL, etc.

Muchos desarrolladores optan por preparar dos o más versiones alternativas de la misma página WEB para evitar los problemas de compatibilidad con navegadores. Existen editores como Dreamweaver, Fusion o GoLive donde es extremadamente simple convertir tablas en capas o viceversa. Una vez que se tienen las dos versiones alternativas de las páginas WEB, pueden subirse ambas al servidor y colocar en las páginas un script que reconoce el browser del visitante, de manera que dirige la página, si es preciso, a la versión soportada por el navegador.

2.8.4.2 Testing de la Capa del Negocio

2.8.4.2.1 Performance del Software

El objetivo es asegurar que el software actúa de acuerdo a las especificaciones operacionales en cuanto a tiempos de respuesta, costos de procesamiento, uso de almacenamiento y salidas por impresora.

La información de performance se recolecta durante las siguientes operaciones:

- Transacciones normales actuales y esperadas.
- Transacciones pico actuales y esperadas.
- Volúmenes de transacciones mínimo.

2.8.4.2.1.1 Captura correcta de datos

La captura correcta de datos se refiere al uso de scripts CGI (Common Gateway[4] Interface) o ASP (Active Server Pages) para capturar datos del cliente web, tales como formularios, números de tarjeta de crédito, y detalles de pagos. Cualquier error en la captura de estos datos resultará en el procesamiento incorrecto de las ordenes del cliente.

Sobre este aspecto se deben realizar pruebas con técnicas de caja negra y caja blanca, análisis de condiciones o valores límite, etc. (ver punto 2.3.1 y 2.3.2).

2.8.4.2.1.2 Completitud de transacción

La completitud de las transacciones es el aspecto más importante para transacciones de sistemas en ambiente WEB y distribuido. Cualquier error en esta fase de la operación puede generar acciones legales debido a que la parte afectada puede estar en riesgo de perder dinero o información valiosa debido a transacciones incompletas.

2.8.4.2.1.3 Compatibilidad de Gateway

Muchas transacciones que se llevan a cabo en aplicaciones WEB de relevancia incluyen validaciones contra otros sistemas utilizando servicios WEB que

[4] Máquina virtual que realiza una conversión de protocolos de comunicación entre diferentes tipos de redes o aplicaciones para que estos puedan interactuar.

requieren un gran marco de compatibilidad. En este aspecto pueden destacarse transacciones de validación de tarjetas de crédito, de identidad de personas, etc.

Estas transacciones se realizan a través de gateways, lo que hace que las mismas agreguen complejidad debido a problemas de compatibilidad que pueden surgir entre diferentes gateways. Esto hace que las transacciones no sean confiables. Se recomienda que antes de poner el sitio en producción se hagan pruebas piloto on-line para probar la confiabilidad de los gateways.

En estos dos últimos puntos, se deben hacer pruebas funcionales utilizando datos de usuarios simulados.

2.8.4.2.2 Prueba de Carga de Servidores

La capacidad de carga de los servidores es un factor importante en la productividad de un sitio WEB. El sitio WEB debe soportar volúmenes de carga que aseguren que el servicio no será interrumpido debido a que el número de usuarios conectados ha superado la cantidad esperada. Además de esto, un sitio WEB exitoso es víctima frecuente de ataques de hackers que utilizan el método de sobrecarga de transacciones para lograr sus objetivos.

La capacidad es la cantidad de usuarios simultáneos que el servidor puede manipular exitosamente. La carga excesiva en el servidor web provoca una grave degradación en la performance hasta que se reduzca la carga. El objetivo de las pruebas de carga es determinar el número óptimo de usuarios simultáneos.

Para prevenir caídas y sobrecarga de tráfico en servidores en horarios pico, los administradores deben invertir en equipos, sobredimensionando la capacidad para el tráfico normal esperado. La capacidad de un servidor para manejar sobrecarga en horas pico depende de la velocidad de la red, el poder de procesamiento, memoria y espacio de almacenamiento del servidor.

2.8.4.2.2.1 Prueba de stress

Algunos sistemas están diseñados para manipular un volumen de carga específico. Por ejemplo, un sistema de procesamiento de transacciones de banco puede estar diseñado para procesar hasta 100 transacciones por segundo, un sistema operativo puede estar diseñado para controlar hasta 200 terminales separadas. Generalmente, esta prueba incluye la planificación de una serie de exámenes donde la carga se incrementa gradualmente para reflejar el patrón de uso esperado. La ejecución del sistema en un ambiente de stress crea altas demandas de recursos y ayuda a probar el comportamiento del sistema en ambientes donde hay interferencia continua.

Las pruebas de stress aumentan constantemente la carga del sistema más allá de la carga máxima para la que fue diseñado el sistema hasta que se produce un fallo. Este tipo de prueba tiene doble función:

- Prueba el comportamiento del sistema ante fallas. Pueden surgir situaciones imprevistas debido a la combinación aleatoria de eventos en los que la carga a la que el sistema es expuesto excede la carga

máxima prevista. Las pruebas de stress determinan si la sobrecarga del sistema resulta en pérdida de datos o servicios para el usuario.

- Alborota el sistema y puede revelar ciertos defectos, que, normalmente, no se manifestarían como errores.

Las pruebas deben estar diseñadas para asegurar que el sistema pueda procesar la carga esperada.

La prueba de stress es particularmente relevante para aplicaciones WEB que utilizan servicios de bases de datos donde frecuentemente se observa degradación severa cuando se satura la red con llamadas al sistema operativo.

2.8.4.2.2.2 Software para pruebas de carga

En el mercado existe gran variedad de software para realizar pruebas de carga. La técnica utilizada por la mayoría de estos productos consiste en simular múltiples conexiones de usuario. Luego de diversas ejecuciones, el software calcula el factor de carga óptimo para el servidor WEB. Los datos provistos por el software son luego utilizados para configurar el servidor WEB. Finalmente, si el tráfico en el servidor se incrementa por encima de la capacidad de carga, el servidor dejará de aceptar nuevos requerimientos de usuarios on-line.

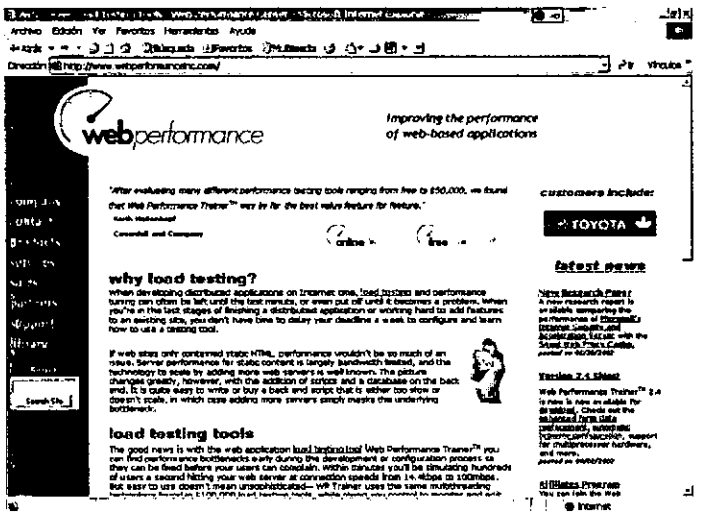


Figura 6 – Página www.Webperfcenter.com

La herramienta *Web Application Stress* (WAST) de Microsoft está diseñada para simular múltiples navegadores requiriendo páginas al servidor WEB. Esta herramienta puede simular muchos requerimientos con relativamente pocas máquinas cliente (Asegurarse que se está utilizando el número adecuado de máquinas cliente).

2.8.4.3 Testing de la Capa de Acceso a Datos

2.8.4.3.1 Prueba de la Base de Datos

Una de las partes importantes de las aplicaciones es la base de datos. Las aplicaciones basadas en WEB no son la excepción debido a que deben almacenar registros de transacciones, perfiles de usuarios, catálogos, etc.

Luego de la creación de una base de datos se debe almacenar una copia de la misma (*mirror*) en el mismo u otro servidor. La base de datos original se la mantiene tal cual está y las pruebas se realizan sobre copia de la base de datos. El proceso de pruebas se realiza sobre la copia de la base de datos y continúa hasta que las pruebas resultan satisfactorias. Luego, las modificaciones son implementadas en la base de datos original.

2.8.4.3.1.1 Ubicación de una base de datos

Las bases de datos no deben que estar en el mismo servidor que las capas intermedia y la capa WEB. El servidor de la bases de datos debería estar separado del servidor WEB por un firewall. Esto agrega complejidad al proceso de testing.

2.8.4.3.1.2 Objetivos del testing de la capa de datos

A diferencia del resto de las capas, el testing sobre la capa de datos es un proceso continuo. Esto se debe a que las bases de datos no son estáticas; los datos cambian con el tiempo.

Las bases de datos deben ser verificadas en busca de errores de:

- Relevancia de resultados de búsquedas.
- Tiempo de respuesta de las consultas.
- Integridad de datos.
- Validez de datos.

- Recuperación.

2.8.4.3.2 *Opciones de Búsqueda*

La opción "Búsqueda" es una de las más utilizadas en bases de datos on-line. Los resultados de las búsquedas proveen acceso directo a otras páginas, etc.

Si la aplicación contiene esta funcionalidad, se debe controlar que los resultados de las mismas sean relevantes para el usuario.

Esta prueba debe ser llevada a cabo por un grupo de personas ajeno al equipo de desarrollo. Este equipo asume el rol del cliente on-line y ejecuta búsquedas aleatorias con diferentes palabras clave. Los resultados de las búsquedas serán registrados de acuerdo a un porcentaje de relevancia con la palabra clave. Al final del proceso de pruebas, el equipo estará en condiciones de hacer una serie de recomendaciones que pueden ser implementadas como consultas a la base de datos.

2.8.4.3.3 *Tiempos de Respuesta*

El tiempo de respuesta ante consultas es esencial en transacciones on-line. El tiempo de respuesta para atender consultas en una base de datos debe ser corto. Los resultados de ese testing pueden ayudar a identificar problemas, tal como cuellos de botella en redes, consultas específicas, la estructura de la base de datos o el hardware.

2.8.4.3.4 Integridad de Datos

El proceso de pruebas debe verificar la correctitud de los datos almacenados.

Ejemplos de pruebas de integridad:

- Evaluar que la creación, modificación y eliminación de datos en tablas se realice tal como se lo especifica en las funcionalidades.
- Probar qué sucede cuando la base de datos devuelve como respuesta un valor nulo o un conjunto vacío.
- Verificar que, cuando un conjunto particular de datos es guardado en la base de datos, cada valor se almacene completamente. En otras palabras, que no se trunquen strings ni se redondeen valores numéricos.
- Chequear que los valores por omisión sean almacenados en la base de datos si el usuario no los especifica.
- Evaluar la compatibilidad con datos y hardware antiguo, así como versiones del sistema operativo e interfaces con otros software.

2.8.4.3.5 Validez de los Datos

Los errores más comunes en los datos son los producidos por el ingreso incorrecto. A estos errores se los denomina *errores de validez de datos*. Realmente, son los más difíciles de detectar y presentan mayor frecuencia cuando se ingresa un gran volumen de datos en poco tiempo. Por ejemplo, un valor de $67 puede haber sido ingresado en forma inadvertida como $76, siendo inválido este dato.

Los errores de validez pueden reducirse mediante el uso de reglas de validación de datos en los campos de ingreso. Por ejemplo, en el ingreso de fecha con formato DD/MM/AAAA, se puede incorporar reglas de validación donde el número ingresado en la parte MM sea entre 01 y 12 o que el rango del número ingresado en DD dependa del mes MM.

En muchos casos, las reglas simples de validación en los campos de ingreso no pueden detectar errores de validez. En estas situaciones puede hacerse la validación de la entrada utilizando consultas sobre datos confiables.

2.8.4.3.6 Pruebas de Recuperación de Errores

Esta prueba tiene como objetivo hacer que el sistema falle de diversas maneras y así asegurar que:

- El sistema tolera fallas, lo que significa que las fallas en el procesamiento no paralizan el funcionamiento global del sistema.
- El sistema se recupera de fallas y continúa procesando dentro del período de tiempo establecido en las especificaciones.
- La recuperación de datos y el reinicio del sistema con autorecuperación son correctos. Si la recuperación requiere intervención humana, entonces el tiempo medio para reparar la base de datos se encuentra dentro de los límites aceptables predefinidos.

2.8.4.4 Otras Pruebas

2.8.4.4.1 Pruebas de Seguridad

La confianza en la aplicación es fundamental para el éxito de cualquier sistema; los usuarios tienen que sentir que su información está segura en el sistema. Esta tarea no es sencilla, requiere mucho tiempo y esfuerzo.

La técnica principal para realizar pruebas de seguridad es intentar sobrepasar los controles de seguridad de la aplicación. Esta técnica cerciora que los mecanismos de protección del sistema controlan accesos no permitidos.

Una de las formas de realizar esto es que el equipo encargado de las pruebas doblegue al sistema con solicitudes continuas y, de ese modo, denegando servicios a otros usuarios. Otra forma puede ser la provocación de errores para penetrar en el sistema durante el proceso de recuperación del mismo o navegar por datos inseguros hasta encontrar la clave de ingreso al sistema.

Hay dos áreas en la seguridad de aplicaciones WEB: seguridad de la red y seguridad de las transacciones. En estas áreas se debe considerar:

- Anonimato.
- Autenticación.
- No repudiación.
- Control de integridad.

2.8.4.4.1.1 Seguridad de las redes

El sistema operativo de red, junto con el *firewall*, se encargan de la seguridad de la red. Si alguno de estos dispositivos falla, usuarios no autorizados pueden intentar acceder a información confidencial o dañar los datos almacenados en el servidor. Este tipo de hueco puede deberse a medidas insuficientes de seguridad en las redes.

El sistema operativo de red debe estar configurado para permitir el acceso sólo a usuarios autenticados, previniendo en forma efectiva el acceso de hackers.

Por ejemplo, si un hacker accede por el puerto inseguro de FTP (puerto 25) de un servidor WEB, puede luego utilizarlo como punto de acceso a toda la red y así acceder a los datos del servidor. El hacker puede incluso acceder a cualquier máquina conectada a ese servidor.

Las pruebas de seguridad indicarán cuáles son estas áreas vulnerables e incluso ayudará a configurar la red para mayor seguridad.

La seguridad de la conexión de red a Internet se puede chequear utilizando programas, como Kane Security Analyst (KSA) que complementa el testing de la red con el de los sistemas operativos utilizados. KSA además verifica:

- Permisos de usuario.
- Ubicación de discos removibles.
- Fortaleza de las políticas de contraseñas.
- Uso de scripts de conexión y fechas de expiración de contraseñas.
- Almacenamiento de contraseñas en texto plano o encriptado.

2.8.4.4.1.2 Seguridad de transacciones específicas

No todas las transacciones en un sistema tienen igual importancia. Algunas, como las que efectúan pagos o movimientos de dinero, deben tener seguridad suficiente como para crear confianza en los usuarios y brindar un procesamiento predecible.

Las pruebas a realizarse en este sentido deben asegurar que la información es transmitida y almacenada en forma segura y que los datos importantes de usuarios (información de tarjetas de crédito, por ejemplo) se almacenan con una fuerte encriptación y sólo se permite acceso limitado y autorizado.

2.8.4.4.2 Pruebas de Aceptación

Es el paso final en el proceso de testing antes de que el sistema sea aceptado para uso operacional.

La pruebas de aceptación son llevadas a cabo en un ambiente de producción y utilizando un conjunto de funcionalidades del negocio. Se realizan luego de que se han completado las pruebas de funcionalidad. En estas pruebas se incluyen pruebas del sistema con datos reales provistos por el cliente, en lugar de los datos simulados que hayan sido desarrollados como parte del proceso de verificación.

Estas pruebas revelan frecuentemente errores y omisiones en la definición de los requerimientos del sistema que pueden no reflejar las facilidades actuales y

performance requerida por el usuario. Este nivel de pruebas confirma que el sistema está listo para producción.

En esta etapa se puede tener apoyo en la búsqueda de errores a través de la ejecución de funcionalidades piloto sobre un conjunto selecto de clientes o realizando encuestas sobre amigabilidad con el usuario, conveniencia, atractivo visual, relevancia, respuesta, etc.

2.8.4.4.3 Pruebas de Regresión

Los errores en las componentes del sistema pueden surgir en forma tardía en el proceso de pruebas, por lo que el proceso debe ser iterativo. Debido a que la reparación de los defectos puede incorporar nuevos defectos, las pruebas de regresión tienen como objetivo analizar nuevamente todas las funcionalidades y componentes del sistema que hayan sido modificados como consecuencia del proceso de pruebas, asegurando que el sistema funciona correctamente, incluso luego de los cambios realizados en partes del sistema.

Lineamientos para las pruebas de regresión:

- Probar cada modificación hecha al sistema para asegurar que no se han introducido nuevos problemas y que la performance operacional no se ha visto degradada como resultado de las modificaciones.
- Cualquier cambio en el sistema luego de completar cualquier fase del proceso de pruebas debe estar sujeto a una nueva prueba de regresión. Esto se hace para asegurar que los efectos de los cambios son

transparentes a otras áreas del sistema y a otros sistemas que tengan interfaces con éste.

- El equipo del proyecto debe crear datos de prueba basándose en las especificaciones.

3 CONCLUSIÓN

En este informe se describen los conceptos teóricos a tener en cuenta para plantear una estrategia de testing de software capaz de maximizar la cantidad de errores encontrados de forma sistemática, ordenada y sobre todo fundamentada sobre la base del conocimiento científico.

En la próxima etapa de desarrollo del presente proyecto (actividades 2 y 3) se elaborarán los requisitos mínimos, estrategia de testing, procedimientos y lineamientos necesarios y la generación de las plantillas para documentar los resultados de los procedimientos de prueba que la S.E.T.I. adoptará para el testing de software, tanto como para los desarrollos internos como para los desarrollos contratados, llevados a cabo por terceros.

INFORME DEL DESARROLLO DE LA ACTIVIDAD N° 2

1 ENUNCIADO DE LA ACTIVIDAD

Elaboración de requisitos mínimos, estrategias y procedimientos a utilizar por la S.E.T.I. para el testing de aplicativos adquiridos en el marco de la Autopista de la Información y demás contrataciones, sobre la base de los desarrollos y definiciones técnicas de la actividad anterior.

2 DESARROLLO

En la actividad anterior del presente proyecto se realizó un desarrollo de los conceptos teóricos generales de testing cubriendo varios aspectos, entre ellos, métodos de prueba de software, niveles de prueba, estrategias de pruebas de software, diseño de casos prueba, etc.

La presente actividad amplía estos conceptos en el sentido específico de testing de aplicativos y, además, determina la forma de ponerlos en práctica tanto en sistemas desarrollados por la Secretaría de Estado de Tecnologías de la Información (S.E.T.I.) como por terceros. Así mismo, en el desarrollo de esta actividad se dan lineamientos, requisitos mínimos, procedimientos, instrucciones e información de utilidad que permite, a los líderes de proyectos de aplicaciones o al grupo de testing, crear planes de prueba para los distintos niveles de testing, llevar a cabo la ejecución de los mismos y determinar si el

software probado tiene una fiabilidad alta para poder ser aceptado y puesto en producción.

2.1 PLAN DE PRUEBAS

La planificación es un elemento crucial para que el esfuerzo realizado en la etapa de testing resulte beneficioso.

El plan de pruebas depende generalmente de tres factores:

- **El tipo de proyecto** (tradicional, iterativo, mantenimiento, software comprado, etc.)
- **El tipo de sistema** (batch, a eventos, manejo de base de datos, procesamiento de imágenes, transaccional, etc.)
- **El alcance del proyecto** (interfaces con otros sistemas, tareas del negocio que se ven afectadas por el sistema, etc.)

El documento de plan de pruebas que se presente a la S.E.T.I. o se desarrolle por ésta tendrá que describir como mínimo: el alcance, el enfoque o estrategia, los responsables, los recursos requeridos, la planificación, el análisis de riesgos de un proceso de pruebas y las actividades necesarias. Identificar los elementos de prueba, las características que deben probarse, las tareas de prueba, lo que hará cada tarea y cualquier planificación necesaria para la contención de riesgos"

El documento de plan de pruebas podrá ser construido a partir de documentos menores agrupados.

La especificación de requerimientos en la documentación de un producto de software no solo determina los objetivos y lista las características, éste también provee las bases para toda la actividad de testing sobre el producto. El equipo de testing está encargad, entre otros, crear, causar o provocar comportamientos no esperados por la descripción de las especificaciones. Incluso más, se recomienda que el equipo de testing forme parte, también, de la especificación de requerimientos para que posean un conocimiento mayor del producto deseado.

El plan de pruebas completamente enmarca al proceso de testing e incluye los casos de prueba individuales. Para desarrollar un plan de pruebas sólido, se debe explorar sistemáticamente el software para asegurar una cobertura completa del mismo, pero que no sea innecesariamente repetitiva. Un plan de pruebas formal establece un proceso de testing que no dependa de pruebas aleatorias ni accidentales.

El testing, así como el desarrollo del producto, puede fácilmente llegar a convertirse en una tarea que se extienda en gran medida en el tiempo. De este modo, las especificaciones de la aplicación y, subsecuentemente, el plan de pruebas, deberían definir la calidad mínima esperable para aceptar la aplicación.

Un plan de pruebas incluye:

1. **Identificador del plan:** Preferiblemente de alguna forma mnemónica que permita relacionarlo con su alcance, por ejemplo: PG-ProcTest (plan global del proceso de Testing), PV-ReqSeg (plan de verificación

de requerimientos de seguridad), PP-Mod-NC-SisEdu (plan de prueba del módulo del nivel central del sistema de educación) Como todo artefacto del desarrollo, el plan de pruebas está sujeto a control de configuración, por lo que debe distinguirse adicionalmente la versión y fecha del plan.

II. **Alcance:** Indica el tipo de prueba y las propiedades / elementos del software a ser probado.

III. **Ítem a probar:** Indica la configuración a probar y las condiciones mínimas que deben cumplir para comenzar a aplicarse el plan. Se debe tener en cuenta que, por un lado, es difícil y riesgoso probar una configuración que aún reporta fallas; aunque por otro lado, si se espera a que todos los módulos estén depurados (libre de errores), puede que se detecten fallas graves demasiado tarde.

IV. **Estrategia:** Describe la técnica, patrón y / o herramientas a utilizar en el diseño de los casos de prueba. Por ejemplo, en el caso de pruebas de unidad de un procedimiento (ver 2.1.1.3.2) se podría indicar: "Se aplicará la técnica caja negra de fronteras de la precondición" o "Ejercicio de los caminos ciclomáticos válidos". En lo posible la estrategia debe precisar el número mínimo de casos de prueba a diseñar, por ejemplo, 100% de las fronteras, 60% de los caminos ciclomáticos, etc. La estrategia también explicita el grado de automatización que se exigirá, tanto para la generación de casos de prueba como para su ejecución.

V. **Categorización de la configuración:** contiene las condiciones bajo las cuales el plan de pruebas debe ser suspendido, repetido o culminado.

En algunas circunstancias el proceso de prueba debe suspenderse debido a los defectos o fallas detectados. Al corregirse éstos, el proceso de prueba previsto por el plan puede continuar, debiendo indicarse que primero, si es el caso, se deben ejecutar los casos de pruebas correspondientes a los niveles inferiores[5] y luego se continúa con la ejecución de los mismos del nivel en que se detectaron los errores (ver punto 2.2).

Los criterios de culminación (ver punto 2.1.4) pueden ser tan simples como aprobar el número mínimo de casos de prueba diseñados o tan complejo como tomar en cuenta no sólo el número mínimo, sino también el tiempo previsto para las pruebas y la tasa de detección de fallas.

VI. **Tangibles:** Explicita los documentos a entregarse al culminar el proceso previsto por el plan de pruebas, por ejemplo, subplanes, especificación de pruebas, casos de prueba, resumen gerencial del proceso, listado de funciones probadas, listado de sistemas operativos compatibles, etc.

[5] . Esto se debe a que es necesario repetir algunas pruebas porque las modificaciones realizadas pueden afectar a otras partes del programa que hacen uso del código corregido.

VII. **Procedimientos especiales:** Identifica el grafo de las tareas necesarias para preparar y ejecutar las pruebas, así como cualquier requerimiento especial.

VIII. **Recursos:** Especifica las propiedades necesarias y deseables del ambiente de prueba, incluyendo las características del hardware, el sistema operativo y cualquier otro software necesario para llevar a cabo las pruebas, así como la instalación específica del software a probar (qué módulos se colocan en qué máquinas de una red local) y la configuración del software de apoyo.

La sección incluye una estimación de los recursos humanos necesarios para el proceso. También se indica cualquier requerimiento especial del proceso: actualización de licencias, tiempo en la máquina de producción, seguridad, etc.

IX. **Cronograma:** Describe los hitos del proceso de prueba y el grafo de dependencia en el tiempo de las tareas a realizar.

X. **Manejo de riesgos:** Explicita los riesgos del plan, sus acciones mitigantes y de contingencia.

XI. **Responsables:** Especifica quién es el responsable de cada una de las tareas previstas en el plan.

2.1.1 Estrategia de las Pruebas

2.1.1.1 Objetivos

Los objetivos del plan de pruebas, en forma más detallada, son:

- Facilitar las tareas técnicas de las pruebas.
    - Aumentar la cobertura de las pruebas.
    - Evitar repeticiones innecesarias y no olvidar otras.
    - Analizar el programa y descubrir buenos casos de prueba rápidamente.
    - Suministrar una estructura para la prueba final.
    - Aumentar la eficiencia de las pruebas.
    - Comprobar su completitud.
- Mejorar la comunicación sobre las tareas y los procesos de las pruebas.
    - Comunicar el pensamiento detrás de la estrategia de prueba.
    - Provocar reentrada sobre la precisión y cobertura.
    - Comunicar el tamaño del trabajo de prueba.
    - Provocar reentradas sobre la profundidad y temporización.
    - Dividir el trabajo.
- Suministrar estructura para la organización, planificación y gestión del proyecto de prueba.
    - Alcanzar un acuerdo sobre las tareas de prueba
    - Identificar las tareas.
    - Estructurar.

- Organizar.
- Coordinar.
- Aumentar la responsabilidad personal.
- Medir el estado del proyecto y mejora la responsabilidad del proyecto.

2.1.1.2 Alcance de las Pruebas

Los métodos de pruebas que deben cubrir los documentos del plan de prueba son: pruebas de caja blanca, pruebas de caja negra y pruebas no funcionales (ver punto 2.3 de la Actividad 1 del presente proyecto).

En el caso de que se prueben sistemas desarrollados a medida, se requiere la aplicación en forma estricta de todas las metodologías. En el caso de que se adquieran sistemas ya desarrollados o como se los llama comúnmente "enlatados", como no se cuenta con los códigos fuentes y documentación de diseño, entre otras cosas, no se puede aplicar la metodología de caja blanca. Sólo se puede hacer una prueba de sistema muy estricta haciendo énfasis en las funcionalidades provistas y en los resultados obtenidos.

NOTA:

Se deberá tener en cuenta que las pruebas de caja blanca no nos brindan información sobre:

- Errores relacionados con el tiempo.

- Condiciones de error no detectadas.
- Condiciones especiales de los datos.
- Invalidez de la información mostrada por pantalla.
- Inconsistencias en la interfaz de usuario.
- Interacción con tareas en segundo plano.
- Fallos de configuración / compatibilidad.
- Incapacidad de soportar el volumen de carga o fallos hardware.

Las pruebas de caja blanca deben planificarse en la etapa de codificación.

2.1.1.3 Niveles de Testing

2.1.1.3.1 Etapas del Testing

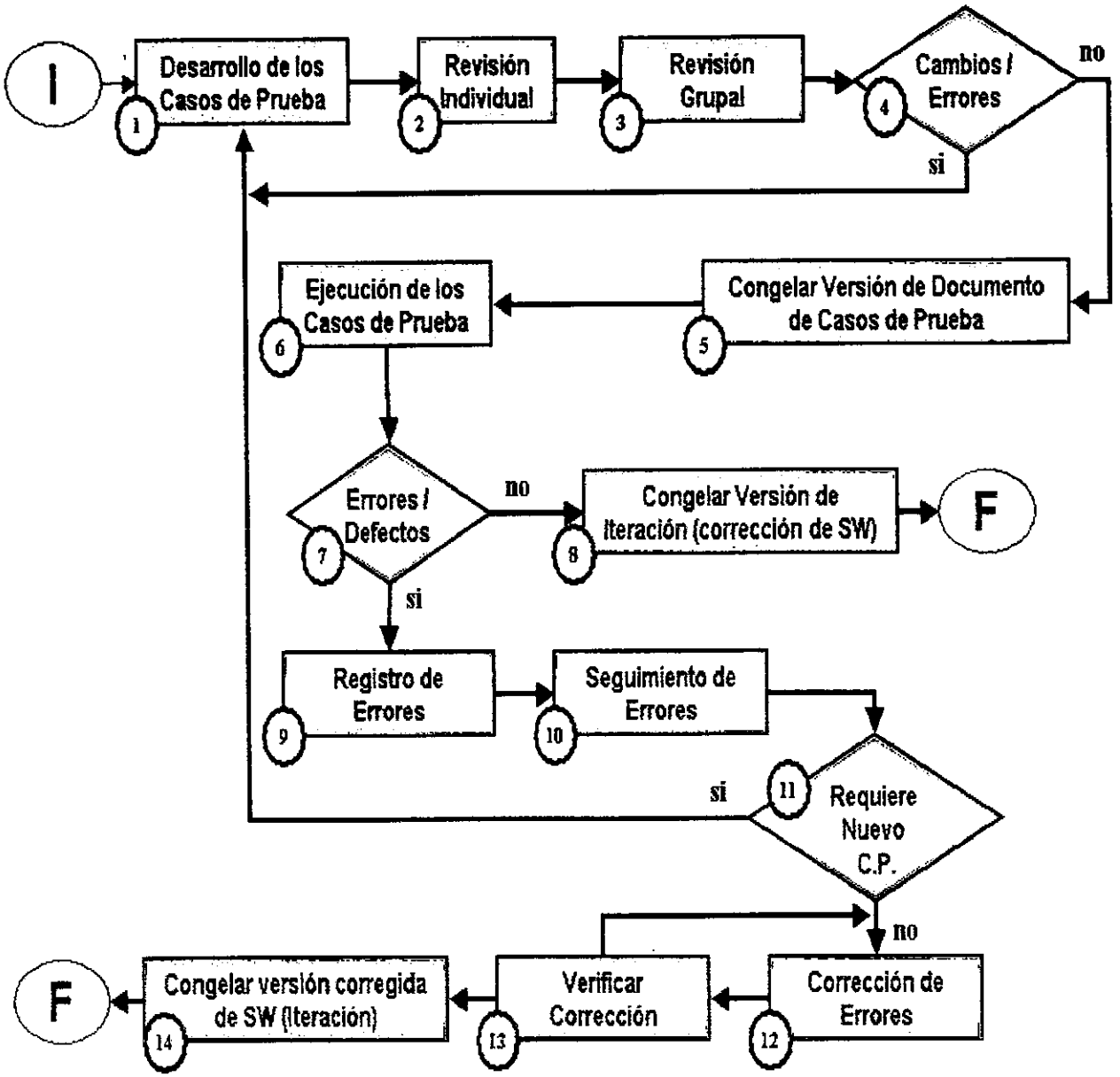


Figura 7 - Esquema de Etapas Comunes a Todos los Niveles de Testing.

Las actividades de Definición, Ejecución, Análisis y Evaluación y Corrección del testing de los distintos niveles deberán basarse en el siguiente esquema de etapas (ver Figura 7):

1) Desarrollo de los Casos de Prueba.

En esta etapa se definen y documentan los lineamientos de los casos de prueba a ejecutar en cada nivel de testing. El diseño y construcción de los casos de prueba dependen en gran medida de nivel de testing.

Aquí también se define el entorno de datos y se corrigen los errores encontrados en la etapa 3) y se verifican las justificaciones de los no corregibles.

2) Revisión Individual (de los Casos de Prueba).

En esta etapa, cada integrante del grupo de testing verificará que los casos de prueba cubren los objetivos planteados para cada nivel, si estos fueron diseñados correctamente y si no se han contemplado algunos casos. También se registrarán los defectos encontrados y aquellas sugerencias que surjan.

3) Revisión Grupal (de los Casos de Prueba).

En esta etapa se exponen y discuten los resultados obtenidos en la revisión individual en forma grupal. Se obtienen los reportes individuales, se identifican los errores potenciales, se registran los errores confirmados como defectos y se justifican la no corrección de aquellos que parecían defectos. También se verificará si los errores se repiten en otras partes.

4) Cambios / Errores.

SI se producen cambios o se encuentran errores ir a 1).

NO ir a 5).

5) **Congelar la Versión del Documento de Casos de Prueba.**

Esto se realiza una vez que se aprueban todas las correcciones (y justificaciones de no corrección). Aquí se aprueba y se registra la versión de casos de prueba que se ejecutarán.

6) **Ejecución de los Casos de Prueba** (ver punto 2.2).

7) **Errores / Defectos**

En esta etapa se realizará un análisis para detectar los errores o defectos en los resultados obtenidos de la ejecución de los casos de prueba y se deberá determinar el tipo de error o defecto y su criticidad.

Por cada caso de prueba se comparará el resultado esperado con el resultado obtenido. También se deberá realizar un análisis del error (ver punto 2.3.4)

SI se encuentran errores o defectos ir a 9)

NO ir a 8)

8) **Congelar versión de iteración (corrección de SW[6]).**

En este punto el sistema ha superado la prueba de testing y cumple con el / los criterio / s de finalización acordados para el mismo.

9) **Registro de Errores.**

[6] SW = Software.

Verificar que se registraron los resultados de la ejecución de los casos de prueba. En la Actividad 3 del presente proyecto se presenta un modelo de plantilla para registro de errores.

10)Seguimiento de Errores.

En esta etapa se evaluarán los errores determinando si los mismos debieron ser detectados en niveles de testing inferiores y si estaban asociados a algún caso de prueba (ver punto 2.3.5).

11)Requiere Nuevo C. P. [7]

SI es necesario el desarrollo de nuevos casos de prueba ira a 1).

NO ir a 12).

12)Corrección de Errores.

La corrección tiene como objetivo eliminar los errores detectados durante el Testing, este es llevado a cabo por los desarrolladores y programadores

13) Verificar Corrección.

Una vez corregidos los errores se deben ejecutar los casos de prueba correspondientes a los mismos (iteraciones del nivel), ver punto 2.2.

14)Congelar versión corregida de SW (iteración).

En este punto el sistema ha superado la prueba de testing y cumple con el / los criterio / s de finalización acordados para el mismo.

[7] C.P. = Casos de Prueba.

2.1.1.3.2 Pruebas de Unidad

Es el nivel de testing más básico. Se focaliza en los bloques más pequeños de un componente o sistema en forma separada. Es el proceso de ejecutar cada componente unitaria para confirmar que cada una realiza la función asignada (ver punto 2.4.1 de la Actividad 1 del presente proyecto).

Se probarán:

- La interfaz.
- Las estructuras de datos locales.
- Las condiciones límite.
- Los caminos independientes.
- Los caminos de manejo de errores.

Se aplican las técnicas vistas de Caja Negra y Caja Blanca.

La prueba de unidad comienza cuando se han terminado de escribir y corregir las sintaxis del código fuente correspondiente al módulo. Dado que el módulo no es un programa independiente es necesario crear cierto entorno para que pueda ejecutarse.

Un **conductor o módulo impulsor**, es un programa que simula la actividad de los módulos de nivel superior en una aplicación, éste acepta o provee los datos de entrada de la prueba, llaman al módulo que esta siendo probado, pasa esos datos y, después de la prueba, recibe los resultados mostrando los datos de salida.

Un **resguardo o módulo auxiliar**, es un programa esqueleto que simula algunas funciones simples del módulo completo para permitir el testing de los módulos superiores de la estructura, éste usa la interfaz del módulo a ser probado, recibe los datos de aquel, muestra la verificación de la entrada, devuelve el o los datos necesarios por el módulo y termina.

Los resguardos típicamente realizan:

- Un retorno inmediato.
- Retornan una constante.
- Retornan una respuesta tabulada.
- Retornan una respuesta interactiva.
- Realizan un cálculo mínimo.

Esta táctica se usa mientras sea posible construir resguardos y conductores simples. Si el módulo requiere conductores y resguardos complejos, la prueba total del módulo se pospone hasta la etapa de la prueba de integración.

Los objetivos de la prueba de unidad son:

- Verificar que el código de cada componente unitaria de la aplicación trabaja de acuerdo con sus especificaciones.
- Validar la lógica del componente probado.

La prueba de unidad logra estos objetivos a través de la identificación precisa de las discrepancias entre lo que una componente unitaria de software hace y lo que se supone que debe hacer.

En el desarrollo de los casos de prueba se deberá definir y documentar los lineamientos de los mismos por cada componente. Para ello, los casos de prueba se deben diseñar de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba.

Los aspectos a considerar para la definición de los lineamientos de los casos de prueba son, entre otros, los siguientes:

- Rutinas de excepción.
- Rutinas de error.
- Manejo de parámetros.
- Validaciones.
- Valores válidos e inválidos.
- Valores límites.
- Rangos.
- Mensajes posibles.

Los controles que se deben realizar son la verificación de que las componentes de prueba unitaria hayan sido probadas.

El listado de verificación asegura que el código de cada componente unitaria de una aplicación haya sido probado individualmente. Este listado servirá de base para la transferencia a la prueba de integración.

2.1.1.3.3 Pruebas de Integración

La Prueba de Integración es la etapa en la que se identifican los errores introducidos por la combinación de componentes probadas unitariamente. Para esto se deberá construir la estructura de un programa para descubrir defectos que surgen de la interacción e interfaces entre los distintos componentes, que no pueden ser detectadas durante la prueba de unidad al mismo tiempo que se lleven a cabo las pruebas para detectar errores de integración de módulos.

A menudo existe una tendencia a realizar una **integración no incremental**: se combinan todos los módulos por anticipado y se prueba todo el programa en conjunto. En este caso la corrección se hace difícil debido a que aparecen muchos errores juntos y complicados de aislar debido a la extensión del programa.

En contraposición al método anterior se propone el de **integración incremental**: el programa se construye y se prueba en pequeños segmentos en donde los errores son más fáciles de aislar y corregir.

Existen dos formas de realizar la integración incremental:

- Descendente
- Ascendente

Integración Descendente

Se integran los módulos, moviéndose hacia abajo en la jerarquía de control, comenzando por el módulo de control principal (programa

principal). Los módulos subordinados se incorporan a la estructura de dos formas:

- Primero en profundidad: los módulos de una misma rama de la estructura.
- Primero en anchura: los módulos que se encuentran en un mismo nivel dentro de la estructura global.

Pasos del Proceso de integración descendente:

I. Se usa el módulo principal como conductor de la prueba haciendo resguardos para todos los módulos directamente subordinados al principal.

II. Dependiendo de la aproximación elegida se sustituyen los resguardos subordinados, uno a uno, por los módulos originales.

III. Se llevan a cabo las pruebas cada vez que se integra un nuevo módulo.

IV. Luego de terminar cada conjunto de pruebas se reemplaza otro resguardo por el módulo real.

V. Se hacen pruebas de regresión (repetición de las mismas pruebas que ya se habían realizado: todo tendría que funcionar igual que antes) para asegurar que no se introdujeron errores en la integración. Y se vuelve al paso II hasta haber integrado toda la estructura del programa.

La principal desventaja de la integración descendente es la necesidad de los resguardos. Y tiene como ventaja que se pueden probar de antemano las distintas funciones del software.

Integración Ascendente

Se comienza la integración con los módulos inferiores de la estructura. Esta técnica tiene como ventaja sobre la anterior que evita la necesidad de crear resguardos.

Se siguen los siguientes pasos:

I. Se combinan los módulos de igual nivel en grupos que realicen una subfunción específica de software.

II. Se escribe un conductor para coordinar la entrada y la salida de los casos de prueba.

III. Se prueba el grupo

IV. Se eliminan los conductores y se combinan los grupos moviéndose hacia arriba por la estructura del programa.

La principal desventaja de esta técnica es que el programa como entidad no existe hasta que se haya añadido el último módulo.

Posee como ventaja la mayor facilidad de diseño de casos de prueba y la falta de resguardos.

La prueba de integración tiene tres objetivos principales:

- Verificar que las interfaces entre los componentes de software de la aplicación funcionan correctamente.

- Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.
- Verificar que las especificaciones de diseño son alcanzadas.

La prueba de integración acompaña estos objetivos a través de la integración de las componentes de prueba definidas en los diferentes niveles de funcionalidad identificados.

En la prueba de integración participarán el grupo de testing conjuntamente con el grupo de desarrollo y la responsabilidad será compartida.

Una vez definidos y documentados los caso de prueba y definido el entorno de datos correspondiente, los casos de prueba deberán diseñarse considerando el orden de ejecución de los mismos y los objetivos de la prueba de integración

En la realización de la prueba de integración se debe tener en cuenta la manera en la cual se combinan los distintos componentes unitarios para constituir un componente funcional mayor, es decir la mecánica del testing (ver punto 2.1.1.5).

Los controles que se deben realizar son la verificación de que los casos de prueba de integración estén completos y registrados formalmente

Tener en cuenta para la integración de las componentes, si es necesario, la creación de módulos impulsores y módulos auxiliares para simular las interfaces con otras componentes funcionales.

2.1.1.3.4 Pruebas de Sistema

La prueba de sistema es el punto más difícil entre las pruebas en el proceso de desarrollo, esto se debe a que es una prueba multifacética que cubre múltiples aspectos del sistema, para lo cual varias pruebas separadas deben ser llevadas a cabo.

En este punto, donde el software es incorporado a otros elementos del sistema (hardware, datos), se deben realizar una serie de pruebas diferentes para verificar y validar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

El objetivo es verificar que el sistema alcanza sus objetivos y requerimientos originales y que funciona dentro de los límites definidos (requerimientos no funcionales).

La prueba de Sistema deberá incluir como mínimo las siguientes pruebas:

- Prueba Final de Requerimientos
- Pruebas de Usabilidad
- Pruebas de Performance
- Pruebas de Documentación y Procedimientos
- Pruebas de Seguridad y Controles
- Pruebas de Volumen
- Pruebas de Esfuerzo (Stress)
- Pruebas de Recuperación

- Pruebas de Múltiples sitios
- Pruebas de Resistencia
- Pruebas de Rendimiento
- Pruebas de Capacidad de Servicio o Mantenimiento
- Pruebas de Configuración de Equipos
- Pruebas de Campo
- Pruebas de Confiabilidad
- Pruebas de Compatibilidad y Conversión
- Pruebas de Instalación

La prueba de sistema es compleja porque intenta validar un número de características al mismo tiempo, a diferencia de otras pruebas que sólo se centran en uno o dos aspectos del sistema al mismo tiempo.

2.1.1.3.4.1 Prueba Final de Requerimientos

Es un chequeo final de cuán bien el sistema alcanza los objetivos del negocio originales y satisface los requerimientos del sistema.

2.1.1.3.4.2 Pruebas de Usabilidad

Es una prueba para determinar cuán bien el usuario podrá usar y entender la aplicación. Identifica las áreas de diseño que hacen al sistema de difícil uso para el usuario.

La prueba de usabilidad detecta problemas relacionados con la conveniencia y practicidad del sistema desde el punto de vista del usuario.

Entre los problemas de usabilidad más comunes encontramos:

- El sistema es demasiado complejo y difícil de usar.
- El sistema es difícil instalar y entender.
- La recuperación de errores es pobre y los mensajes de error no tienen significado.
- La sintaxis de los comandos es difícil de aprender y recordar.
- El sistema fuerza al usuario a recordar formatos y secuencias fijas.
- Los procedimientos no son simples ni obvios.
- El sistema no tiene instrucciones de ayuda por computadora y tiene manuales pobres.
- Los diagramas, pantallas, reportes y gráficos son de calidad y apariencia pobre.
- La lógica y conveniencia de los botones, pantallas y mensajes de ayuda no es la solicitada.

2.1.1.3.4.3 Pruebas de Performance

El propósito de esta prueba es medir objetivos de servicio específicos definidos previamente. Las principales actividades son:

- Comparar la performance del sistema actual con la de los requerimientos.
- Poner a punto el sistema para mejorar las métricas de performance.
- Proyectar la capacidad futura de carga del sistema.

Los objetivos de nivel de servicio definidos deben guiar la prueba de performance. Entre las consideraciones de performance que se tomen no deberían faltar:

- Errores lógicos
- Procesamiento ineficiente
- Diseño pobre: muchas interfaces, instrucciones y entradas / salidas
- Cuellos de botella en discos, CPU ó canales de entrada / salida
- Salidas del sistema
- Tiempos de respuesta
- Capacidad de almacenamiento
- Ratio de entrada / salida de datos
- Número de transacciones que pueden ser manejadas simultáneamente

2.1.1.3.4.4 Pruebas de Documentación y Procedimiento

El objetivo de esta prueba es evaluar la exactitud y claridad de la documentación del usuario y determinar si el manual de procedimientos trabajará correctamente como una parte integral del sistema.

Muchos defectos son identificados cuando se chequean los manuales y documentación del usuario, esto se debe, entre otras cosas, a una descoordinación interna de los grupos de trabajo en el desarrollo del sistema, a que un programa no se lo ve como parte integrante de un sistema mayor, etc.

Aquellos programas que conforman sistemas mayores y que no están completamente automatizados, es decir, que contienen procedimientos realizados por operadores, administradores de la base de datos o el usuario de terminal, (cualquier procedimiento humano), deberán ser probado durante el testing de sistemas.

2.1.1.3.4.5Pruebas de Seguridad y Controles

El objetivo de esta prueba es evaluar y verificar el funcionamiento correcto de los mecanismos y controles de seguridad del sistema para asegurar la integridad y confidencialidad de los datos. El foco principal es probar la vulnerabilidad del sistema frente a accesos o manipulaciones no autorizadas.

Para encontrar casos de prueba para este tipo testing es recomendable estudiar problemas conocidos de seguridad en sistemas similares y lo que se deberá buscar es tratar de mostrar la existencia de dichos problemas en el sistema que se examina.

Se deberán tener en cuenta mínimamente las siguientes consideraciones:

- Controles de acceso físico.

- Acceso a estructuras de datos específicas a través de los programas de aplicación y otros utilitarios y / o programas.
- Seguridad en sitios remotos.
- Existencia de datos confidenciales en reportes y pantallas.
- Acceso diferenciado a funciones del sistema por rol / perfil de usuario.
- Controles manuales, incluyendo aquellos para autorización y aprobación, formularios, documentación numerada, transmisión de datos, balances y conversión de datos.
- Controles automáticos, incluyendo aquellos para edición de datos, chequeo de máquinas, errores del operador, acceso a datos elementales y archivos, acceso a funciones, auditoría, entre otros.

El encargado de esta prueba debe ubicarse en el lugar del que intenta penetrar al sistema en forma no autorizada. Todo recurso es válido para este tipo de prueba: tratar de conseguir las claves, producir errores intencionales en el sistema para entrar en la etapa de recuperación, etc.

Evidentemente si se le da a una persona el tiempo y los recursos suficientes, terminará por penetrar el sistema; por lo tanto, el objetivo es lograr que el costo de penetrar el sistema sea mayor que el valor de la información.

2.1.1.3.4.6 Pruebas de Volumen

Las pruebas de volumen hacen referencia a grandes cantidades de datos para determinar los límites en que se provocan fallas en el sistema. También

identifican la carga máxima o volumen que el sistema puede manejar en un momento dado.

El objetivo de la prueba es verificar que la aplicación funciona adecuadamente y someterla a grandes cantidades de datos para determinar si el mismo puede manejar el volumen de datos especificado. Se prueba bajo los siguientes escenarios de volumen:

- Máximo (actual o físicamente posible) número de clientes conectados (o simulados), todos ejecutando la misma función (peor caso de desempeño) por un período extendido.
- Máximo tamaño de base de datos (actual o escalado) y múltiples consultas ejecutadas simultáneamente

Para realizar esta prueba se deben utilizar los scripts diseñados para las pruebas de desempeño. Deben usarse múltiples clientes, ya sea corriendo las mismas pruebas o pruebas complementarias para producir el peor caso de volumen (ver punto 2.3.3.3 y 2.1.1.3.4.7) por un período extendido. Se utiliza un tamaño máximo de datawarehouse (actual, escalado o con datos representativos) y múltiples clientes para correr consultas simultáneamente para períodos extendidos.

La tarea finaliza cuando todas las pruebas planeadas han sido ejecutadas y los límites especificados en el sistema se han conseguido o excedido sin que el sistema falle.

La prueba de volumen es costosa, tanto en tiempo de máquina como en personal.

2.1.1.3.4.7 Pruebas de Esfuerzo (Stress)

Esta prueba no debe confundirse con la prueba de volumen: un esfuerzo grande es un pico de volumen de datos que se presenta en un corto período de tiempo. Puesto que la prueba de esfuerzo involucra un elemento de tiempo, no resulta aplicable a muchos programas, por ejemplo a un compilador o a una rutina de pagos. Es aplicable, sin embargo, a programas que trabajan bajo cargas variables, interactivos, de tiempo real y de control de proceso.

El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos.

Aunque muchos pruebas de esfuerzo representan condiciones que el programa encontrará realmente durante su utilización, muchas otras serán en verdad situaciones que nunca ocurrirán en la realidad. Esto no implica, sin embargo, que estas pruebas no sean útiles. Si se detectan errores durante estas condiciones "imposibles", la prueba es valiosa porque es de esperar que los mismos errores puedan presentarse en situaciones reales, algo menos exigentes.

2.1.1.3.4.8 Pruebas de Recuperación

Estas pruebas aseguran que una aplicación o sistema se recupere de una variedad de anomalías de hardware, software o red con pérdidas de datos o fallas de integridad.

Las pruebas de tolerancia a fallas aseguran que, para aquellos sistemas que deben mantenerse corriendo, cuando una condición de falla ocurre, los sistemas alternos o de respaldo pueden tomar control del sistema sin pérdida de datos o transacciones.

Las pruebas de recuperación son contrarias a las pruebas en que la aplicación o sistema es expuesto a condiciones extremas (o condiciones simuladas), tales como fallas en dispositivos en entrada/salida o llaves o punteros inválidos de base de datos. Esta prueba evalúa las características de contingencia construidas en el sistema para procesar interrupciones y para volver a puntos específicos en el ciclo de procesamiento del sistema. Los procesos de recuperación se invocan y la aplicación es monitoreada y/o inspeccionada para verificar que éstos mecanismos se han ejecutado en forma apropiada.

La recuperación debe ser considerada en el proceso de diseño.

Errores de programación o de datos pueden ser incorporados en un sistema para determinar si se puede recuperar de ellos. Las fallas de equipo (por ejemplo errores de paridad en memoria, errores en dispositivos de entrada / salida) pueden ser simuladas.

Si la recuperación es automática se debe evaluar el correcto funcionamiento de los casos de uso que la componen (reinicialización, recuperación de datos de arranque, etc.).

Si la recuperación requiere de intervención humana se deberá evaluar los tiempos medios de reparación para determinar si está dentro de los límites aceptables y su complejidad.

Los siguientes tipos de condiciones deben incluirse en la prueba:

- Interrupción de electricidad en el cliente.
- Interrupción de electricidad en el servidor.
- Interrupción en la comunicación hacia el servidor (caídas de red).
- Interrupción en la comunicación con los controladores de disco.
- Ciclos incompletos (procesos de consultas interrumpidos, procesos de sincronización de datos interrumpidos).
- Llaves o punteros de base de datos inválidos.
- Elementos corruptos o inválidos en la base de datos.

Se deben utilizar las pruebas creadas para la funcionalidad del sistema y procesos de negocios para crear una serie de transacciones. Una vez se alcanza el punto inicial de las pruebas de recuperación, se deben realizar o simular las siguientes acciones:

- Interrupción de electricidad en el cliente.
- Interrupción de electricidad en el servidor: simular o iniciar procedimientos de pérdida de energía para el servidor.
- Interrupción de la comunicación en la red. (desconectar físicamente los cables o apagar los hubs o routers).
- Interrupción de la comunicación con los controladores de disco: simular o eliminar físicamente la comunicación con uno o mas controladores o dispositivos. Una vez se realizan estas acciones, se deben ejecutar

series de transacciones, y luego, una vez alcanzado el segundo punto de pruebas, se deben invocar los procedimientos de recuperación.

2.1.1.3.4.9 Pruebas de Múltiples Sitios

El propósito de esta prueba es evaluar el correcto funcionamiento del sistema o subsistema en múltiples instalaciones.

Mínimamente debe considerarse:

- Consistencia de las opciones de configuración para el sistema a través de los sitios.
- Sincronización de datos entre sitios.
- Comunicación de datos entre sistemas en diferentes sitios.
- Consistencia de controles y seguridad a través de los sitios.

2.1.1.3.4.10 Pruebas de Resistencia

Está diseñada para probar el sistema con situaciones anormales, es decir una mayor demanda de recursos, tanto en cantidad como en frecuencia. Esto tiene por objeto responder a la pregunta: ¿con qué carga se puede poner a funcionar un sistema antes de que falle?

2.1.1.3.4.11 Pruebas de Rendimiento

Las pruebas de rendimiento miden tiempos de respuesta, índices de procesamiento de transacciones y otros requerimentos sensibles al tiempo. El objetivo de las pruebas de rendimiento es verificar y validar los requerimientos de rendimiento que se han especificado (en este caso, el rendimiento ofrecido por el proponente).

Las pruebas de rendimiento usualmente se ejecutan varias veces, utilizando en cada una, carga diferente en el sistema. La prueba inicial debe ser ejecutada con una carga similar a la esperada en el sistema. Una segunda prueba debe hacerse utilizando una carga máxima.

Adicionalmente, las pruebas de rendimiento pueden ser utilizadas para perfilar y refinar el rendimiento del sistema como una función de condiciones tales como carga o configuraciones de hardware.

Las pruebas de rendimiento pueden hacerse con las siguientes técnicas:

- Utilizar los procedimientos de prueba desarrollados para las pruebas del modelo del negocio (Pruebas del Sistema).
- Modificar archivos de datos (para incrementar el número de transacciones) o los scripts para incrementar el número de veces que ocurre cada transacción.
- Los scripts pueden ser ejecutados en una máquina y deben ser repetidos con múltiples clientes (virtuales o actuales).

Las pruebas de desempeño integrales incluyen tener una carga en background en el servidor. Hay varios métodos que pueden ser utilizados para hacer esto:

- Transacciones dirigidas directamente al servidor, usualmente en forma de sentencias SQL.
- Creación de usuarios virtuales para simular muchos clientes (usualmente varios cientos). Se utilizan herramientas de emulación de terminales remotas para obtener esta carga. Esta técnica también puede ser utilizada para cargar de tráfico la red.

Use múltiples clientes físicos, cada uno corriendo los scripts de prueba. Las pruebas de rendimiento deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas. La base de datos utilizada para pruebas de rendimiento debe ser de un tamaño real o proporcionalmente más grande que la diseñada.

2.1.1.3.4.12 Pruebas de Capacidad de Servicio o Mantenimiento

El sistema puede tener, también, objetivos de capacidad en cuanto se refiere al servicio o mantenimiento. Deben probarse todos los objetivos de esta naturaleza. Estos objetivos podrían definir la ayuda al servicio provistas con el sistema (por ejemplo, programas de vuelcos de memoria, diagnósticos), el tiempo medio para encontrar problemas aparentes, los procedimientos de mantenimiento y la calidad de la documentación de la lógica interna.

2.1.1.3.4.13 Pruebas de Configuración de Equipos

Estas pruebas verifican la operación del sistema en diferentes configuraciones de hardware y software. En la mayoría de los ambientes de producción, las especificaciones para las estaciones de trabajo, equipos de red y servidores pueden variar. Las estaciones pueden tener diferentes versiones de software instaladas (Sistemas Operativos, Drivers, etc) y en cualquier momento, pueden llegar a utilizarse diferentes combinaciones.

Con frecuencia, el número de configuraciones posibles es demasiado grande para intentar una prueba de cada una de ellas, pero el programa debe probarse al menos con cada tipo de dispositivo y con las configuraciones mínima y máxima posibles.

2.1.1.3.4.14 Pruebas de Campo

El objetivo es correr el sistema en el ambiente real para encontrar errores y validar el producto contra sus especificaciones originales.

2.1.1.3.4.15 Pruebas de Confiabilidad

El objetivo es predecir cuando ocurrirá una falla de sistema, o el tiempo esperado entre fallas.

2.1.1.3.4.16 Pruebas de Compatibilidad y Conversión

El propósito es demostrar que los objetivos de compatibilidad no han sido logrados y que los procedimientos de conversión no funcionan.

La mayoría de los programas que se desarrollan no son completamente nuevos; con frecuencia son reemplazos de partes deficientes, ya sea de sistemas de procesamiento de datos, o sistemas manuales. Como tales, los programas tienen a menudo objetivos específicos con respecto a su compatibilidad y a sus procedimientos de conversión con el sistema existente.

2.1.1.3.4.17 Pruebas de Instalación

Las pruebas de instalación tienen dos propósitos. El primero es asegurar que el sistema puede ser instalado en todas las configuraciones posibles, tales como nuevas instalaciones, actualizaciones, instalaciones completas o personalizadas, y bajo condiciones normales o anormales; estas últimas incluyen insuficiente espacio en disco, falta de privilegios para algunas tareas, etc. El segundo propósito es verificar que, una vez instalado, el sistema opera correctamente. Esto usualmente implica correr un número significativo de pruebas de Funcionalidad.

El sistema debe instalarse apropiadamente en cada cliente, bajo las siguientes condiciones:

- Instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el sistema.

- Actualizar máquinas previamente instaladas con el sistema.
- Instalar versiones viejas en máquinas previamente instaladas con el sistema.

2.1.1.3.5 Pruebas de Regresión

El objetivo es probar el sistema después de los cambios realizados durante el debugging[8], mantenimiento o desarrollo de nuevas versiones del sistema. Es una prueba que intenta determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes.

Se debe tener en cuenta que la prueba de regresión es una nueva corrida de pruebas previas, que requiere políticas para decidir qué casos de prueba incluir.

Estas pruebas pueden ser automatizadas puesto que las mismas son repetidas una y otra vez; aquí las herramientas para minimizar el esfuerzo del trabajo son útiles.

Las pruebas de regresión se debería realizar regularmente después de haber probado profundamente un área del programa.

NOTA

[8] Proceso de depuración de errores.

Aquellos casos de prueba que descubren defectos tempranamente deben ser incluidos en la prueba de regresión.

2.1.1.3.6 Pruebas de Validación

Se dice que el software está validado cuando éste funciona de acuerdo a las expectativas razonables del cliente. Las expectativas razonables quedan definidas en las especificaciones de requerimientos de software, donde se explicitan los atributos de software visibles al usuario.

La prueba de validación se lleva a cabo con una serie de pruebas de caja negra y una vez terminada la misma nos encontraremos ante estas dos condiciones:

- Las características de funcionamiento o de rendimiento están de acuerdo con las especificaciones y son, por lo tanto, aceptables
- Se descubre alguna desviación de las especificaciones y se crea una lista de deficiencias. En este caso los errores descubiertos raramente se pueden corregir de inmediato sin afectar la calidad, por lo que es necesario negociar con el cliente.

En este punto la mayor parte del sistema está terminado y es aquí donde algunas compañías utilizan este como disparador de actividades. (planificación, ejecución de pruebas, manuales, etc.)

El jefe del grupo de pruebas, debe probar el programa y determinar si las características perdidas son lo suficientemente importantes como para no poder llamar alfa al programa (ver el punto siguiente).

Dado un programa casi alfa, la declaración del hito alfa se hace conjuntamente por los jefes de proyecto y pruebas.

2.1.1.3.7 Pruebas Alfa y Beta

Son técnicas de prueba de aceptación para permitir que el cliente valide los requerimientos, cuando se cree que existen errores que sólo el usuario final puede descubrir.

Esta técnica se utiliza cuando se tiene un producto de software a ser utilizado por muchos clientes.

- **Prueba Alfa**: es la conducida por el cliente en el lugar de desarrollo bajo la supervisión del jefe de desarrollo.

  La definición de alfa varia mucho. Las principales son:

  - La mayoría de las funcionalidades están, pero pueden faltar algunas o ser inestables. El programa muestra claramente su naturaleza y estilo.
  - Todas las funciones están codificadas, aun cuando tengan errores importantes. Funcionan todos o algunos de los tipos de dispositivos. No existe riesgo de codificación.

- Las funciones del núcleo están terminadas (modelo evolutivo). Todas las características críticas están: producto mínimo aceptable. Se puede usar el producto y ver su aspecto.

La prueba alfa deberá, entre otras cosas, probar los manuales del sistema y atacar puntos del diseño,

- **Prueba Beta**: se lleva a cabo en los lugares donde se correrá el sistema, con los clientes, bajo condiciones de operación naturales. Los clientes registran todos los problemas que encuentran durante la prueba e informan a intervalos regulares al equipo de desarrollo.

  Al igual que alfa, la definición varía y entre ellas podemos encontrar:

  - El programa está preparado para enviarse a los clientes designados para la prueba beta que usarán el producto y registrarán sus experiencias. El producto:
    - Debe ser usable.
    - Soporta la mayoría de dispositivos.
    - Puede tener algunos problemas serios, de los que se deberán alertar.
    - La mayoría de los temas de diseño (desarrollo) están resueltos.
    - En algunos modelos todo se codifica en alfa, en otros falta algo. Beta significa que se puede evaluar, pero no se encuentra implementado completamente.

- Método en Cascada:
  - Completadas y probadas todas las características, sin errores fatales pero con alguno serio.
  - Archivos de datos no esenciales al 50% y los demás casi terminados.
  - El producto cumple con los requisitos iniciales.
- Método Evolutivo:
  - El corazón del producto está completo junto con las características mínimas aceptables (totalmente probadas) y se pueden haber agregado otras características deseables (cada vez que se añade una característica o grupo de ellas se inicia un nuevo ciclo de pruebas).
  - En ocasiones se puede utilizar un tercer hito (¿gamma?), similar a las anteriores definiciones de beta, donde se tiene un producto "vendible".

2.1.1.3.8 *Pruebas de Aceptación*

La Prueba de Aceptación verifica la habilidad de la aplicación para alcanzar los objetivos del negocio y los requerimientos del sistema. El Propósito de la misma es certificar que el software satisface sus objetivos de negocio

originales, alcanza los requerimientos de usuarios definidos y opera dentro de los límites identificados.

La prueba de aceptación es ejecutada por el usuario y el equipo de testing. Para la misma, se define un protocolo de aceptación que identifica el criterio de aceptación de la prueba y los casos de prueba que serán ejecutados.

En la definición de la prueba de aceptación se deberá identificar los casos de prueba a ejecutar durante la misma y definir el entorno de datos correspondiente.

Una vez ejecutada y aprobada la prueba por parte del usuario, se firmará una acta de aceptación del producto. Este documento habilita el inicio de la fase de transición para el software correspondiente a la prueba. A partir de aquí el sistema puede ponerse en producción.

La prueba de aceptación puede dividirse en prueba pre-final y final:

2.1.1.3.8.1 Prueba Pre-Final

Cualquier rutina de instalación, archivo de datos, u otra cosa se encuentra completo.

Al entrar en pre-final no hay errores abiertos.

Probablemente se encontrarán algunos errores más. Después de ser corregidos y no encontrar otros, el programa va a la prueba final.

En pre-final el grupo de testing tiene la última oportunidad para encontrar errores que paralicen la puesta en producción. Se tienen tres retos:

- Mantener la búsqueda de problemas terribles.
- Asegurar que la corrección de nuevos errores no tiene efectos colaterales. No se puede ejecutar el plan en cada versión, se trabaja sobre él a lo largo de versiones.
- Atar cabos sueltos. Ultima oportunidad para asegurarse de que todas las pruebas críticas hayan sido ejecutadas en algún momento durante el proyecto.

Los errores serios encontrados se corregirán tan pronto como sean informados y se obtiene una nueva versión de programa.

En esta prueba se debe:

- Realizar la última ronda de pruebas de dispositivos.
- En lo posible (si se tiene tiempo), se deberá volver a probar todos los errores corregidos.
- Publicar una lista final de errores aplazados. Esto da una última oportunidad a los jefes para reconsiderar la calidad del producto.
- Evaluar la fiabilidad del producto:

Una vez terminadas las pruebas el sistema se someterá finalmente a las pruebas de aceptación.

Antes de que el sistema entre en producción se deberá evaluar su calidad. La calidad de un producto es su ajuste a las necesidades. Cuando se tengan que dar las medidas de calidad, lo que se deberá proporcionar son las medidas de fiabilidad, quizás suplementadas por comentarios de diseño.

La fiabilidad es alta si la probabilidad de que los usuarios o clientes no encuentren ningún error en el sistema es alta.

La fiabilidad puede clasificarse en cuatro niveles:

- **Fiabilidad baja**: el producto tiene errores serios que el usuario o cliente encontrará.
- **Fiabilidad media**: en algún lugar entre baja y alta.
- **Fiabilidad alta**: el producto ha sido probado adecuadamente y no se pueden encontrar errores serios.
- **Fiabilidad desconocida**: no se ha probado adecuadamente el programa, no se ha encontrado errores groseros o serios, pero se tiene la certeza de que permanecen errores serios.

La S.E.T.I. no debería permitir poner en producción sistemas que no alcancen una fiabilidad alta. Siguiendo los pasos y recomendaciones que se han detallado del presente proyecto se puede realizar un testing de muy buena calidad que nos permitirá saber si el sistema probado cumple con una fiabilidad alta.

2.1.1.3.8.2 Prueba Final

El producto está terminado. Se le pasa una última ronda de pruebas.

La mayoría de compañías paran las pruebas al finalizar el estado pre-final, otras compañías ejecutan una última ronda de pruebas que deben ser realizadas por los usuarios finales o por un grupo de testing diferente.

Los objetivos de la prueba final son:

- Evaluar la fiabilidad del primer día.
- Predecir comentarios de los revisores.

2.1.1.4 Diseño de Casos de Prueba

El diseño de los casos de prueba (ver punto 2.5 de la Actividad 1 del presente proyecto), como cualquier otra actividad, debe comenzar con la definición de sus objetivos.

Es primordial que las pruebas diseñadas detecten sistemáticamente diferentes clases de errores, con la menor cantidad de tiempo y esfuerzo. Los casos de prueba deberán ser diseñados de acuerdo a los métodos de prueba de software (ver punto 2.3 de la Actividad 1 del presente proyecto) y deberá contemplar dónde añadir profundidad a los mismos según lo requiera el plan de pruebas. La profundidad de la prueba puede realizarse, entre otros, sobre los siguientes criterios:

- Errores más probables.
- Errores más visibles.
- Áreas del programa usadas más frecuentemente.
- Áreas más criticas.
- Áreas más propensas a contener errores.
- Áreas distintivas del programa.

- Áreas más difíciles de corregir.
- Áreas más entendidas.

Los mecanismos para añadir profundidad al plan de pruebas se basan en la creación o extensión de los componentes del plan de pruebas (listas, árboles de decisión, lista de funciones, gráficos de borde, matrices de prueba, etc.).

Se deben evitar los casos de prueba desechables, a menos que el programa sea verdaderamente desechable, es decir, los casos de prueba también podrían ser útiles para otros modelos o sistemas.

Es una práctica habitual sentarse frente a la computadora e improvisar un caso de prueba y ejecutarlo inmediatamente sobre el programa. El problema está en la pérdida de tiempo, ya que cuando se tenga que hacer la prueba de regresión (por haber corregido algún error) se tendrá que reinventar la prueba.

2.1.1.5 Mecánica

Los programas grandes requieren tratamientos especiales en cuanto a la forma en que el proceso de testing es estructurado. Es lo que se denomina mecánica del testing.

A través de una buena mecánica, se puede lograr un producto de calidad y dentro de los límites de tiempo.

Para mantener el proyecto bajo control, el jefe del proyecto debe mantener el equilibrio entre:

- Fiabilidad del producto.

- La cantidad y profundidad de sus características.
- Fecha de distribución.

La metodología usada en el proyecto tendrá una gran influencia en la flexibilidad cuando se aproxime la fecha límite de entrega.

Algunas restricciones comunes a toda metodología son:

- **Fiabilidad**. El jefe de proyecto siempre puede vender el producto antes, a un coste de desarrollo menor y con muchos errores.
- **Características**. Otra manera de acortar un proyecto es simplificarlo. Cuando una característica se ha codificado o diseñado mal, o su coste se planificó mal, el jefe de proyecto puede quitarla del producto. Si la característica era importante, perjudicará la satisfacción del usuario.
- **Dinero**. El jefe de proyecto puede apurar el proyecto gastando dinero (nuevas herramientas, consultores, plantillas, etc.)
- **Fecha de venta**. Si el proyecto sobrepasa la planificación, siempre se puede retrasar la distribución. Los costes de retrasar la venta pueden ser enormes:
    - Coste directo de continuar el desarrollo. Se puede estimar multiplicando el sueldo semanal que se paga a cada persona, y multiplicarlo por dos o tres para incluir beneficios, créditos, etc.
    - Pueden existir multas por incumplimiento de contrato. Límites temporales. Un producto que podría haber sido rentable puede fallar si no se distribuye para la temporada alta o porque se

desarrolla después que los competidores lo hayan lo hayan realizado.

- Coste gastado en marketing. La publicidad preventa y los esfuerzos son inútiles si la venta se retrasa mucho después de los anuncios o los artículos.
- Ausencia de ingresos actuales. Si la compañía necesita ingresos líquidos y no los tiene, se estará frente a un problema grave.

En la Tabla 1 se exponen los criterios para determinar el grado y tipo de testing que puede requerirse según el tipo de aplicación.

Aplicación	**Criterio de Prueba**
Aplicación nueva que no reemplaza ninguna aplicación existente	El Gerente de Aplicaciones debería tener gran participación en el desarrollo del plan de pruebas de la nueva aplicación. El plan de pruebas de aceptación debería ser desarrollado en un marco de comunicación amplia con el grupo desarrollador y con cantidad de información suficiente tomada de la documentación del sistema.
Aplicación nueva que reemplaza una aplicación existente	El plan de pruebas debe ser desarrollado usando los aspectos requeridos por el sistema que está siendo reemplazado. Los casos de prueba para cualquier mejora existente en el nuevo sistema deberían ser desarrollados tal como si se tratara de una nueva aplicación, basándose en la información de requerimientos y documentos de diseño. El escenario de pruebas óptimo para las funcionalidades duplicadas en el nuevo sistema corresponde con ejecuciones en paralelo con el sistema viejo y comparación de los resultados obtenidos.

Cambios en la base de datos (sin cambios en la aplicación, por ejemplo actualización de versiones del motor de base de datos)	Se recomienda un testing paralelo con el sistema que utiliza la versión anterior del motor de base de datos. También en este caso el escenario de prueba óptimo es la comparación de los resultados obtenidos para asegurar que la nueva base de datos de la aplicación y los controladores producen los mismos resultados que el antiguo sistema. Se deberían incluir pruebas de performance.
Mejoras en una aplicación (una aplicación existente es mejorada con nuevas funcionalidades)	Se deben desarrollar casos de prueba basados en formularios para cada pantalla agregada a la aplicación existente para asegurar que se comportan correctamente y que se satisfacen los requerimientos. Los casos de prueba de los procesos de negocio deberían ser desarrollados y probados para asegurar que la nueva funcionalidad está correctamente integrada con el resto de la aplicación original y para asegurar que ninguna funcionalidad previamente existente se haya perdido en el proceso de mejora del producto. Los casos de prueba deberían ser desarrollados basándose en los requerimientos específicos de cualquier nuevo reporte. En este caso, las pantallas que no hayan sido modificadas pueden no requerir testing.
Mejoras en una aplicación (la aplicación existente está siendo mejorada para cambiar la funcionalidad existente)	Deben desarrollarse casos de prueba basados en formularios teniendo en cuenta los documentos de requerimientos para asegurar que la aplicación cumpla con los requerimientos de mejoras. Los casos de prueba existentes pueden ser reformados y reutilizados. Los casos de procesos de negocios deberían ser desarrollados y probados para asegurar que la mejora funcional está integrada correctamente con la aplicación existente, y para asegurar que ninguna funcionalidad existente se ha perdido en el proceso de mejora. Los casos de prueba deberían ser desarrollados basándose en los requerimientos específicos de cada uno de los reportes mejorados. En este escenario, cualquier pantalla que ha sido cambiada o es afectada por una modificación, nuevos datos o nuevas funcionalidades deben ser probadas para asegurar que no se ha perdido la funcionalidad existente. Esto asegurará que los nuevos requerimientos están siendo satisfechos.

Cambios en la infraestructura (la aplicación no es modificada, pero se requiere una prueba para transportarla a un nuevo ambiente, servidor o nuevo modo de uso)	En este caso se requiere una prueba de regresión completa de los casos de prueba basados en formularios y de los procesos del negocio; deberían ser repetidos como existan nuevos ambientes que puedan crear problemas inesperados para una aplicación existente.

Tabla 1 Tipo de Aplicación / Criterio de Prueba

2.1.1.5.1 Pruebas Incrementales

En lugar de probar cada módulo en forma aislada, el próximo módulo a ser probado se combina primero con los módulos que ya han sido probados previamente.

Dentro de las pruebas incrementales existen dos métodos o estrategias (ver punto 2.1.1.3.3):

- Descendentes
- Ascendentes

2.1.1.5.1.1 Estrategia descendente

Con esta estrategia de prueba, se requiere la creación de módulos auxiliares (ver punto 2.1.1.3.2) para simular los módulos invocados. De los módulos auxiliares se espera que provean el mismo resultado que si el módulo real fuese invocado.

La estrategia comienza con el módulo superior o inicial del programa. Después de esto no existe ningún procedimiento correcto para seleccionar el próximo módulo que ha de probarse en forma incremental.

Cuando un módulo es probado, los módulos que invoca son representados por módulos auxiliares, que devuelven el control al módulo invocador con un resultado simulado. A medida que la prueba progresa hacia abajo en la estructura del programa, cada módulo auxiliar es reemplazado por el código que representa.

2.1.1.5.1.2 Estrategia ascendente

Esta estrategia requiere la creación de módulos impulsores (ver punto 2.1.1.3.2). Un módulo impulsor es un programa temporal que invoca al módulo que está siendo probado.

Comienza con los módulos terminales del programa (aquellos que no llaman a otros). Después que estos módulos han sido probados, no existe "el mejor" procedimiento para seleccionar el módulo siguiente a ser probado incrementalmente; la única regla es que para ser elegido como módulo siguiente, todos los módulos del módulo subordinado deben haber sido probados previamente.

En la Tabla 2 se muestra una comparación entre ambas estrategias.

Pruebas Descendentes	
Ventajas	**Inconvenientes**
La estrategia es ventajosa si aparecen fallas grandes en la parte superior del programa. Una vez incorporadas las funciones de entrada / salida resulta fácil la representación de casos de prueba. La estructura previa del programa permite demostraciones y ayuda a mantener la moral.	Se requieren módulos auxiliares. Los módulos auxiliares son a menudo más complicados de lo que al principio parece. Antes de Incorporar las funciones de entrada / salida, la representación de los casos de prueba es difícil. Las condiciones de prueba pueden ser imposibles, o muy difíciles de crear. La observación de la salida de la prueba es más difícil. Permite pensar qué prueba y diseño pueden superponerse. Induce a diferir la terminación de la prueba de ciertos módulos.
Pruebas Ascendentes	
Ventajas	**Inconvenientes**
La estrategia es ventajosa si aparecen fallas grandes en la parte interior del programa. Las condiciones de la prueba son más fáciles de crear. Es más fácil la observación de los resultados de la prueba.	Se requieren módulos impulsores. El programa, como una entidad, no existe hasta que es agregado el último módulo.

Tabla 2 Comparación de Estrategias: Ascendente y Descendente

2.1.1.5.1.3 Pruebas de Hilo

Es una técnica que consiste en combinar unidades individuales en hilos de funcionalidad que forman una función o conjunto de funciones y que resultan en una única salida o conjunto relacionado de salidas. Es comúnmente utilizada en la prueba de integración. La prueba de hilo y la incremental pueden ser ejecutadas simultáneamente.

Un hilo se define seleccionando una transacción de negocio particular y determinando qué actividades contribuyen a esa transacción de negocio. Las actividades son probadas separadamente y luego, incorporadas de a una al hilo. El hilo (transacción de negocio) es entonces probado. Los hilos son integrados en turno e incrementalmente probados como subsistemas. Estos subsistemas son entonces integrados en el sistema total.

2.1.1.5.2 Pruebas No Incrementales

Las características de la prueba no incremental son:

- **Requiere más trabajo:** como cada módulo es probado separadamente, el equipo de testing debe crear muchos módulos impulsores y módulos auxiliares.
- **Los errores de interfaces entre componentes son detectados en forma tardía:** se debe a que el programa como, un todo, se lo prueba posteriormente.

- **El debugging requiere más esfuerzo:** se supone que existen errores relacionados con las interfaces entre los módulos. Estos errores no se encontrarán hasta que el programa entero haya sido combinado. A esta altura será difícil señalar el lugar de donde proviene el error, puesto que puede estar en cualquier parte del programa.
- **Los errores de diseño son detectados tarde:** los errores de diseño son generalmente detectados cuando el programa se prueba como un todo, que ocurre muy tarde en esta prueba no incremental.
- **Funciona con programas pequeños y bien diseñados:** se debe a que no existen muchas interdependencias y conexiones cuando tenemos un número pequeño de módulos.
- **Permite el procesamiento en paralelo.**

2.1.1.5.3 Pruebas a Realizar por el Usuario Final

Se recomienda que el usuario del dominio encargado del testing tenga una charla introductoria de la actividad que realizará así como la forma de completar la documentación.

Se debe hacer énfasis en su responsabilidad como representante del Gobierno de la Provincia de San Luis que hace pruebas para aceptar un producto.

2.1.1.5.3.1 Pruebas en el Ambiente de Desarrollo

La verificación (prueba alfa – ver punto 2.1.4.4 –) involucra la ejecución de partes o todo del sistema en ambientes simulados, con el fin de encontrar errores. Se selecciona un grupo de usuarios para la prueba alfa y se les pide que trabajen con el sistema como parte de las pruebas.

La retroalimentación de esta fase produce cambios en el software para resolver los errores y fallas que se descubren.

Las pruebas alfa se llevan a cabo en el lugar en donde fue desarrollado el sistema, en un ambiente controlado, en el cual el / los desarrollador / es está / n presente / s.

2.1.1.5.3.2 Pruebas en el Ambiente de Producción

La validación (prueba beta, ver punto 2.1.4.4) involucra el uso del software en un ambiente real. Se selecciona un grupo de usuarios que ponen a trabajar el sistema en un ambiente real. Usan el sistema en sus actividades cotidianas, procesan transacciones y producen salidas normales del sistema.

Las transacciones y personas que usan el sistema son reales y trabajan en su área de trabajo real. El desarrollador no está presente. Los usuarios están advertidos de que están usando un sistema que puede fallar.

Las pruebas Alfa y Beta se usan cuando el software desarrollado es un producto a ser usado por muchos clientes. Las pruebas de aceptación con un cliente específico son impracticas en estos casos.

2.1.1.6 Análisis de Riesgos

Para identificar los riesgos más importantes se deberá determinar el tipo de sistema de software (transaccional, por lotes, dirigido a eventos, gestor de base de datos, etc.) y el alcance del proyecto (sistema nuevo, sistema modificado, etc.). Con estos datos podemos identificar riesgos estructurales (asociados a la aplicación y a los métodos de construcción), riesgos técnicos (asociados a la tecnología utilizada) y riesgos de tamaño (en todo los aspectos del software)

Los riesgos más comunes asociados a la actividad de testing pueden ser resumidos en:

I. **Desvíos**: causados por el tiempo, el dinero o los recursos humanos. La definición de criterios de terminación de las pruebas (ver punto 2.1.4) ayuda a prevenir este riesgo.

II. **Baja Satisfacción del Cliente**: esto se previene con la definición de casos de prueba en etapas tempranas del desarrollo, validaciones con usuarios y la realización de pruebas de aceptación.

III. **Calidad**: si no se definen los objetivos y se actúa en el testing según ellos, es probable que el testing no logre encontrar errores.

IV. **Detección de problemas antes de entrar en producción**: esto implica un riesgo económico alto.

La realización correcta de un plan de pruebas como el propuesto en este proyecto previene los riegos planteados anteriormente.

2.1.2 Roles

2.1.2.1 Desarrolladores

Los siguientes son algunos roles, dentro de un proyecto de desarrollo de sistemas, que interactúan con el grupo de testing en la etapa de prueba de sistemas:

Director del proyecto

- Es responsable de obtener un producto de alta calidad a tiempo.
- Balancea costos, responsabilidades, características del producto, y planificación.
- Decide qué problemas se solucionarán, con que prioridad y cuáles no (con apelación).
- Debe reconocer qué o quién necesita ayuda observando el progreso de los informes. Debe suministrar esa ayuda.

Analistas, Arquitectos, Diseñadores y Programadores.

- Leen los informes y los responden.
- Tienen problemas cuando: el informe no está claro, no es simple, no se sabe lo que el testeador alega, el error es irreproducible, se pide más información y no se obtiene, el caso de prueba es demasiado complejo y no se dispone de material auxiliar, los términos del informe pueden interpretarse como personales o cuando las estadísticas reflejan una baja productividad.

- Resuelven los problemas informados.

2.1.2.2 Grupo de Testing

El grupo de testing debería estar formado por personas que reúnan los siguientes perfiles: programador, analista funcional, testeador profesional, arquitecto y diseñador de software.

A continuación mencionamos un conjunto posible de roles dentro del grupo de testing:

Director de pruebas

- Es responsable del plan pruebas, la estrategia de pruebas, el diseño de los casos de uso, de la ejecución de las pruebas, de los métodos de prueba a utilizar, la coordinación de la planificación, en pocas palabras, del testing de sistemas en general.
- Designa los roles dentro del equipo.
- Aprueba los documentos e informes, las iteraciones, etc.
- Soluciona problemas de comunicación, productividad, etc.

Testeador Principal

- Responsable de la calidad y el informe de errores.
- Puede ser la única persona que puede cerrar los informes de errores.
- Revisor de los informes cuestionables, aquellos marcados como aplazados o "cómo se diseñó", etc.

Otros testeadores

- Informan de problemas y monitorean su resolución.

Moderador

- Persona experta en testing independiente del proyecto.

2.1.3 Planificación de las Pruebas

La planificación de las pruebas comienza junto con el análisis de la captura de requerimientos y continúa a lo largo en cada fase del desarrollo, en la Figura 8 se muestra en que etapa del desarrollo de software se debe comenzar a preparar los niveles de prueba más generales y el orden en que deben ser ejecutados (ver punto 2.2).

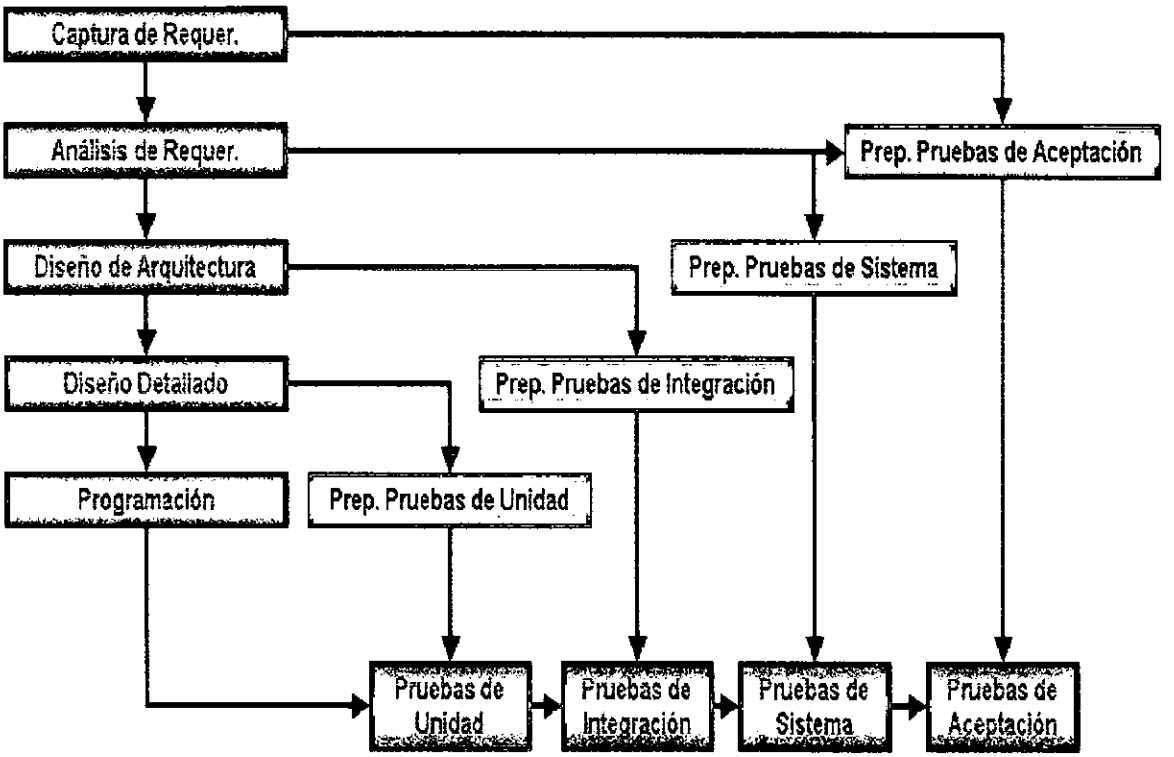


Figura 8 - Etapas de Preparación y Orden de Ejecución de los Niveles de Prueba

La planificación de las pruebas de sistemas desarrollados a medida está sujeta a los cronogramas de cada etapa de desarrollo del proyecto. En este caso el cronograma del plan de pruebas se convierte en un lineamiento definido por el director del proyecto.

En sistemas enlatados, la planificación la realizará el jefe del grupo de testing de acuerdo al tipo y tamaño de sistema que se trate, sin dejar de lado el tiempo máximo disponible para entregar el resultado de la prueba.

2.1.4 Criterios de Finalización de las Pruebas

La prueba es la última tarea dentro de las actividades de garantía de calidad. Aunque la prueba no produce calidad directamente, nos asegura que el producto será mejorado hasta alcanzar los niveles de calidad fijados en las etapas anteriores.

A esta altura del proyecto puede surgir la tentación de apresurar la prueba para lograr una rápida liberación del producto, descuidando la calidad de la prueba en sí misma.

La utilización de métricas relativas a la etapa de prueba nos puede ayudar a determinar en qué momento finalizar la prueba de un producto de software.

Entre las técnicas para determinar el momento de finalización de la prueba se pueden mencionar:

I. Se alcanzó un determinado nivel de cobertura.

Este método es el más cuantitativo de todos. Nos permite establecer el nivel de cobertura deseado que puede estar determinado por:

- Ejecución de un n% de las sentencias del módulo.
- Ejecución de un n% de todos los caminos.
- Ejecución de todos los ciclos condicionales 0,1,...,k veces.

II. Se ha encontrado el número predicho de errores.

Esta técnica requiere contar con una forma de calcular el probable número de errores a ser encontrados en el software.

La prueba finalizaría cuando se ha alcanzado una cifra de errores similar a la predicha.

La forma de calcular el número posible de errores es empírica, es decir basada en datos de proyectos anteriores.

III. La proporción de descubrimiento de errores ha caído

Esta técnica se basa en la suposición que el gráfico de errores descubiertos tiene un dibujo como el mostrado en la Figura 9.

Se decidirá terminar la prueba cuando la curva alcance su punto de inflexión en su porción descendente, pues a partir de este punto el costo y tiempo para descubrir un nuevo error será cada vez mayor.

En la gráfica se ve que la proporción de descubrimiento de errores ha caído y se ha estabilizado, por lo tanto podría optarse por finalizar la prueba.

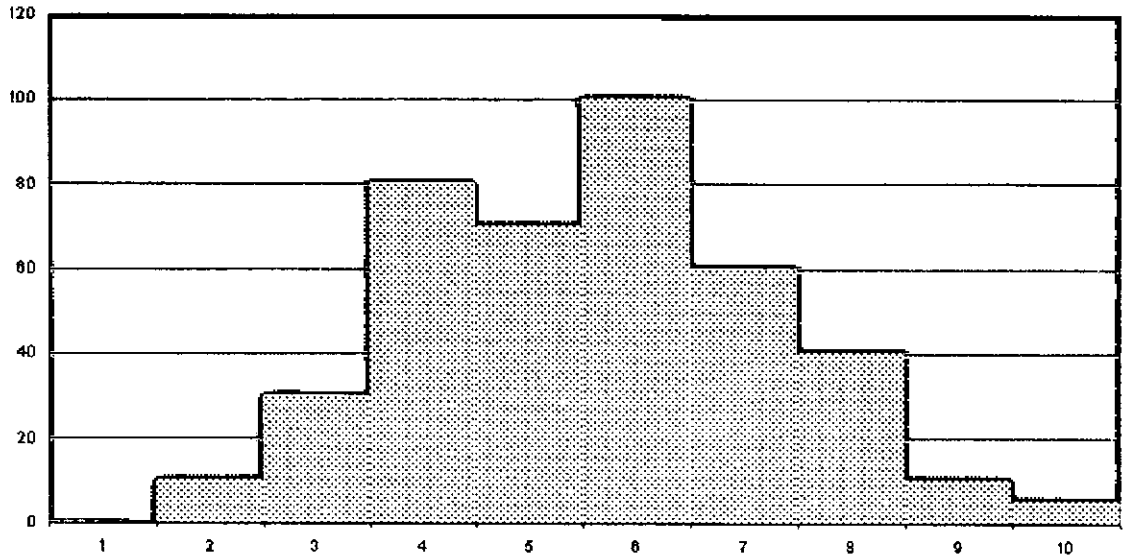


Figura 9 - Distribución de Errores Descubiertos en la Etapa de Testing de un Sistema

IV. Se ha descubierto determinado porcentaje de los errores sembrados

Esta aproximación se basa en el supuesto de que la completitud de la prueba pueda medirse a través de la introducción artificial de errores (sembrado de errores). Los errores sembrados que faltan por descubrir es una indicación de los errores reales que faltan por descubrir.

Mills propuso la siguiente fórmula para determinar el total de errores:

$$\text{TotalErrores} = \frac{\text{ErroresEncontradosNoSembrados} * \text{TotalErroresSembrados}}{\text{ErroresSembradosEncontrados}}$$

V. Se ha alcanzado un determinado nivel de confianza

Se puede definir la confianza como la probabilidad de no tener fallas durante un especificado período de tiempo.

Podemos usar como medida del tiempo entre fallas.

Se podría terminar la prueba cuando el tiempo medio entre fallas haya alcanzado una determinada longitud.

VI. Se ha acabado el tiempo asignado a la prueba

Este enfoque es sin duda el más aplicado, aunque no recomendado.

Su efectividad depende por supuesto de la cantidad de tiempo asignada, la que puede ser mayor o menor de la que es realmente necesaria (generalmente es *mucho menor*), y no tiene en cuenta la calidad del producto.

VII. Se han encontrado errores de tipo no aceptable

Esta técnica clasifica los errores y determina cuales no son aceptables, tanto para continuar con la iteración en un nivel de prueba como para pasara un nivel superior. También define la suma de cuantos errores "aceptables" equivalen a los de un tipo no aceptable y cuanto es la cantidad máxima de errores por tipo que se pueden aceptar.

VIII. Se ejecutaron todos los casos de prueba planeados sin detectares ningún error.

2.1.4.1 Pruebas de Unidad:

Los requisitos mínimos para superar las pruebas de unidad y avanzar a las pruebas de integración son:

- Debe comportarse esperadamente ante valores límites o extremos.

- Debe comportarse esperadamente ante valores erróneos.
- Las pruebas de caminos deben dar los resultados esperados.

2.1.4.2 Pruebas de Integración:

Los requisitos mínimos que se deben alcanzar para superar la prueba de integración son los siguientes:

- Las interfaces entre los componentes de software de la aplicación funcionan correctamente.
- Las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.
- Las especificaciones de diseño han sido alcanzadas.

2.1.4.3 Pruebas de Sistema

Para las Pruebas de Sistema se deben realizar pruebas a muchas características al mismo tiempo (ver punto 2.1.1.3.4). Las pruebas de Sistema concluirán cuando han superado exitosamente todas las pruebas que conforman las Pruebas de Sistema. Si algunas de las pruebas realizadas no se supera, deberá iniciarse una nueva iteración donde deberán permanecer los resultados positivos de las pruebas de la iteración anterior y la corrección en las pruebas donde falló.

Los requisitos mínimos para superar las Pruebas de Sistema son:

- Pasar correctamente cada una de las pruebas que forman las Pruebas de Sistema.
- No debe contener errores en comportamiento respecto a especificación de requerimientos.
- No de contener fallas de robustez.

2.1.4.4 Pruebas Alfa y Beta

Durante las Pruebas Alfa y Beta intervienen los usuarios finales integrando el grupo de testing (ver punto 2.1.1.3.7). Se comienza con la prueba alfa y se finaliza con la prueba beta. Si no se superara la prueba alfa, no se tiene que seguir con la prueba beta. En este caso se tiene que seguir iterando dentro de la prueba alfa hasta cumplir con los requisitos y una vez alcanzados, se avanza a la prueba beta.

La prueba beta sigue el mismo patrón de ejecución que la prueba alfa: se debe iterar mientras no se hayan superado los requisitos mínimos de la prueba.

Los requisitos mínimos a superar por las pruebas alfa y beta son los siguientes:

- Tanto el usuario final y el grupo de testing no deben detectar errores funcionales en la aplicación.
- La cantidad de errores poco relevantes debe ser igual a los pactados con el desarrollador del producto.

2.1.4.5 Prueba de Aceptación

El objetivo de estas pruebas es que se acepte un sistema libre de errores, por ello, se debe iterar hasta que los desarrolladores corrijan los errores pendientes.

Entre los requisitos mínimos que se han de alcanzar se pueden mencionar:

- La ejecución exitosa de los casos de prueba diseñados para este nivel.
- El tipo y cantidad de errores no supera los convenidos por criterio de aceptación (aunque los mismos deben ser corregidos para la puesta en producción).

2.2 EJECUCIÓN DEL PLAN DE PRUEBAS

La ejecución de las pruebas se lleva a cabo en lo que se denomina ambiente de prueba, este representa las condiciones bajo las cuales la prueba se realiza. Este incluye tecnología, software de sistema operativo, software de comunicaciones, software de conectividad, sistemas de administración de bases de datos, herramientas de testing y otros soportes de software y de hardware necesarios para las pruebas.

La ejecución del Plan de Pruebas se realizará mediante la iteración de los casos de prueba en cada una de los niveles de testing hasta que cumplan con los requisitos necesarios para pasar al siguiente nivel. La Figura 10 muestra dichas iteraciones con sus evoluciones:

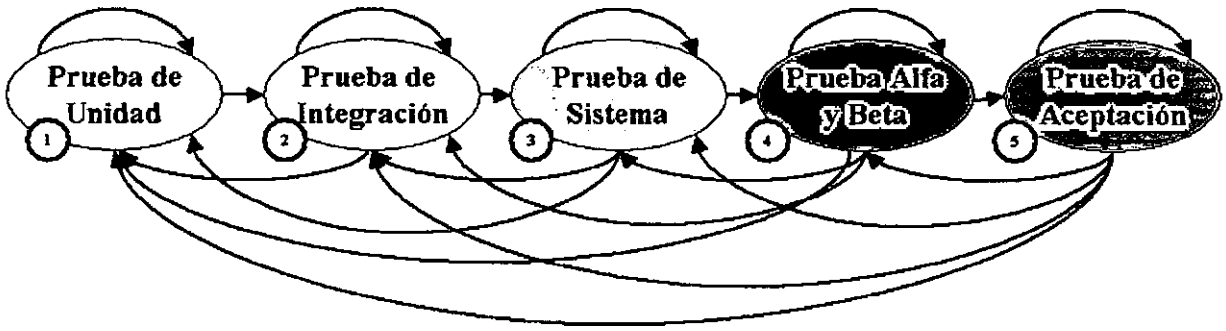


Figura 10 - Ejecución de los Niveles de Testing

En todos los niveles los casos de prueba se ejecutarán en forma iterativa hasta que los mismos no registren errores. Una iteración representa una corrida de ejecución de los casos de prueba definidos en un plan de pruebas específico.

Concluidas las iteraciones y cumpliéndose los requisitos mínimos que forman la base para el próximo nivel de testing se procede a ejecutar los casos de prueba del nivel de testing superior y así sucesivamente hasta finalizar con todos los niveles.

No se debería pasar a un nivel de testing superior sin haberse ejecutado todos los casos de prueba diseñados para el nivel inferior ni sin haberse corregido los errores detectados.

Se deberá tener en cuenta que en los niveles superiores se podrán encontrar errores, que al evaluarlos, tendrían que haberse detectado en niveles de testing inferiores (esto se debe a que se obviaron o no se tuvieron en cuenta ciertos casos de prueba o los mismos no fueron ejecutados correctamente).

Los componentes probados que no pasaron la prueba, es decir, a los que se le detectaron errores, volverán al área de desarrollo para su corrección (programación, diseño, análisis).

La próxima corrida de los casos de prueba deberá hacerse sobre la base de los casos de prueba que presentaron defectos.

2.3 SEGUIMIENTO DE LAS PRUEBAS

Se desarrollarán los tipos de documentos completos que se deben distribuir.

Se deberá tener en cuenta que un documento es válido únicamente si el lector puede entenderlo. Los documentos deberán adaptarse a la sofisticación que sea necesaria.

Dependiendo del destinatario del documento podemos distinguir hasta siete posibilidades:

- **Notas personales.**

  Son los documentos más simples. Deben contener información suficiente como para saber qué prueba se hizo, por qué y su resultado.

  Se usan para describir pruebas que deberán ejecutarse de nuevo, recordar qué se ha hecho, qué no se ha hecho y para contestar a preguntas de los programadores.

- **Notas para otro miembro del equipo.**

Son para dirigir a una persona que conoce el producto y puede resolver dudas.

Estos documentos deben contar: cómo ejecutar la prueba, los resultados esperados, el significado de cada dato, cualquier instrucción adicional, qué pruebas deben ser ejecutadas regularmente y qué es lo que se busca.

- **Notas para otro testeador experimentado.**

  Se debe tener en cuenta que no se pueden solucionar dudas sobre estos documentos. Para escribirlo se debe planificar de 3 a 10 veces el tiempo que se utilizó para desarrollar la prueba y ejecutarla una vez.

  Contiene información sobre todo lo que se especificaba a un miembro del grupo, más materiales generales y indicadores de dependencia.

- **Notas para ser usadas en la próxima publicación (un año más tarde)**

  Deben ser similares a los anteriores. No debe asumirse que se despreciarán los casos mal documentados, al contrario, se utilizarán de forma errónea.

  Debe incluir: detalles de cada prueba, un historial de los errores del programa e incluso más profundidad a la hora de especificar los motivos de cada prueba.

- **Notas de prueba para testeadores inexpertos.**

Si una persona no tiene experiencia como testeador o no está familiarizado con el programa deben añadirse notas que guiarán cada prueba.

Esto permitirá reducir el tamaño del equipo de prueba y proporcionará un conjunto estándar de pruebas.

- **Notas para el gestor.**
- **Seguimiento y auditoria legal.**

2.3.1 Tipos de Documentos de Prueba

A continuación se describen algunos de los documentos que se pueden utilizar en el testing de sistemas. Los mismos no se los debe aplicar hasta el detalle, sino más bien, deben servir como una guía que ayuda a comenzar a escribir el plan de pruebas.

2.3.1.1 Plan de Prueba:

Proporciona un resumen del esfuerzo de prueba del producto. Se puede poner todo en este documento, pero es mejor utilizar varios documentos y referenciarlos en las secciones apropiadas.

Secciones definidas:

- **Identificador del plan.** Nombre o número.
- **Introducción.** Incluye referencias a documentos estándar, planes del producto a alto nivel.

- **Elementos de prueba.** Un elemento de prueba es un elemento de software (función, módulo, característica, etc.) que debe probarse. Se deben listar todos o referir a un documento que lo haga. Incluye referencias a especificaciones y manuales.
- **Características que se debe probar.** Referencias cruzadas a las especificaciones de diseño.
- **Características que no se van a probar.** Cuales y por qué.
- **Aproximación.** Describe la aproximación que se va a utilizar: quién la hace, actividades principales, técnicas y herramientas para cada grupo / característica. Cómo se decidirá si un grupo de características se ha probado satisfactoriamente.
- **Criterios para decidir si pasa / falla un elemento.** Cómo decide un testeador cuando el programa falla o pasa una prueba dada.
- **Criterios de suspensión y reanudación.** Lista de cualquier cosa que pueda causar el paro de las pruebas hasta que se solucione. ¿Qué se debe hacer para volver a comenzar?. ¿Qué pruebas se deben hacer en este punto?.
- **Pruebas que se deben entregar.** Lista de todos los documentos que han de escribirse para este producto.
- **Tareas de prueba.** Lista de todas las tareas que se necesita preparar para realizar las pruebas. Mostrar dependencias entre tareas, personas necesarias, quienes realizan cada prueba, cuanto esfuerzo se requerirá y cuando se realizará.

- **Necesidades ambientales.** Describe el hardware necesario, software, herramientas de prueba, laboratorios, etc.
- **Responsabilidades.** Detalle de los grupos o personas responsables del manejo, diseño, preparación, ejecución, testigos, comprobaciones, arreglos, resolución, preparación del equipo (hardware / software), etc.
- **Necesidades de plantilla y entrenamiento.** Cuánta gente se necesita con una habilidad y qué entrenamiento requieren.
- **Planificación.** Lista de todos los hitos con fechas y cuando será necesario cada recurso.
- **Riesgos y contingencias.** Las mayores suposiciones de riesgo del plan de prueba. Qué puede ir lo suficientemente mal como para retrasar la planificación y qué se hará al respecto.
- **Aprobación.** ¿Quién debe aprobar el plan? Espacio para las firmas.

2.3.1.2 Lista de Funciones (Visibles y no Visibles)

- Simple:
  - Nombre de función.
  - Parámetros (con tipo).
  - Campo de verificación de prueba (checklist) .
- Detallada:

- Nombre de función.
- Parámetros (con tipo).
- Breve descripción.
- Módulos que la referencian (hacen uso de ella).
- Módulos que ella referencia (que usa).
- Campo de verificación de prueba (checklist).

2.3.1.3 Criterio Para Aceptar una Prueba

Prueba breve que debe pasar el programa cuando se envía al grupo de pruebas. Si la pasa se produce un ciclo de prueba completo, de otra forma se rechaza como demasiado inestable.

Esta prueba debe ocupar menos de media hora (nunca más de dos horas).

El documento debe describir esta prueba exactamente. Se debe entregar a los programadores preferiblemente antes de la primera prueba.

2.3.1.4 Especificación del Diseño de las Pruebas

Describe cómo se probará una característica o grupo de ellas. Incluye:

- Identificador de la especificación de diseño.
- Característica que se va a probar.

- Aproximaciones de refinamiento.
- Identificación de las pruebas.
- Criterios de aceptación / fallo.

2.3.1.5 Especificación de los Casos de Prueba

Especifica los casos de prueba. Incluye:

- Identificador de la especificación del caso de prueba.
- Elementos de prueba.
- Especificaciones de entrada. Lista de todos los valores por valor, rangos o nombre. Identificar todo lo que sea relevante, como situación de memoria, valores del sistema operativo, programas o bases de datos de soporte, etc. Describir cualquier consideración temporal.
- Especificaciones de salida. Lista de todos los valores de salida o mensajes. Considerar también los tiempos de respuesta.
- Necesidades ambientales. Lista de requisitos especiales (hardware, software, facilidades, etc.).
- Requisitos procedurales especiales. Lista de cualquier cosa especial en la configuración, acciones o análisis de la prueba.
- Dependencias entre casos. Qué pruebas han de ejecutarse después de esta, por qué y que ocurre si falla entonces.

2.3.1.6 Especificación de Procedimientos de Prueba

Describe las etapas para ejecutar un conjunto de casos de prueba y analizar su resultado. Incluye:

- Identificador de la especificación de procedimiento.
- Propósito. Para qué es el procedimiento y referencias cruzadas a todos los casos de prueba que usen estos procedimientos.
- Requisitos especiales. Lista de cualquier prerrequisito, cualidades especiales de los testeadores y necesidades de entorno.
- Pasos del procedimiento: Log (formato), Configuración (preparación), Comienzo (cómo comenzar la ejecución), Proceder (acciones necesaria), Medida (cómo se realizan medidas), Shut down (cómo parar el procedimiento), Restart (dónde y como comenzar), Parada (cómo parar al final), Finalizar (restaurar entorno original), Contingencias (qué hacer cuando todo va mal).

2.3.1.7 Informe de Transmisión de Elementos de Prueba

Acompaña cualquier entrega para prueba. Cuenta qué es lo que están proveyendo. Incluye:

- Identificador.

- Elementos transmitidos. Nombre de los programas o módulos, versión, revisión. Nombre de las personas responsables de la entrega.
- Localización. Dónde está el material entregado (como está etiquetado).
- Estado. Cómo ha cambiado desde la última vez que se probó, qué problemas se resolvieron, si cambia el comportamiento del programa, etc.
- Aprobación. Las personas que deciden que este material está listo para pasar a la fase de pruebas deben firmarlo.

2.3.1.8 Notas de Prueba

Incluye:

- Instrucciones generales.
- Descripción paso por paso del procedimiento de cada prueba.
- Casillas de marcado para cada paso y resultado.
- Espacio abundante para describir el comportamiento que fuese extraño o no entendido.

2.3.1.9 Registros de Prueba

Registro cronológico de las ejecuciones de pruebas y eventos ocurridos. Incluye:

- Identificador del log.
- Descripción. Qué se está probando, incluyendo versión, donde se hace, hardware y cualquier información de configuración.
- Entradas de eventos y actividades. Qué ocurrió, incluyendo:
    - Descripción de la ejecución. Procedimiento, quien lo hizo.
    - Resultados del procedimiento. Qué ocurrió.
    - Información de entorno. Cambios específicos para esta prueba.
    - Eventos inesperados. Qué ocurrió antes y después.
    - Identificadores de informes de errores.

2.3.1.10 Informe Resumen de Pruebas

Resumen de las series de pruebas del tipo deseado. Después de un ciclo completo de prueba, incluyendo una breve descripción de las pruebas y evaluación de resultados.

Incluye:

- Identificador del informe.
- Resumen. Qué se probó, en qué entorno y el resumen de su evaluación.

- Variaciones. Informar sobre cualquier desviación de los procedimientos de prueba.
- Valoración de la exhaustividad. Decir si las pruebas fueron lo exhaustivas que marcaba el plan de prueba.
- Resumen de resultados. Problemas detectados, resueltos, cual fue su solución y cuales continúan.
- Evaluación. Evaluación global de cada elemento probado (probabilidad o riesgo de fallo).
- Resumen de actividades (tiempo de máquina usado).
- Aprobación.

2.3.1.11 Informe de Incidencias

Por cada error o problema detectado o conjunto de ellos, se describe la importancia de como inciden o afectan los mismos en el sistema en desarrollo.

2.3.1.12 Documentación Insertada en Archivos de Datos y Control

Comentarios en ficheros de entrada para una prueba. Comentarios en ficheros de control. Ventajas: no se desactualizan, van junto con los procedimientos o datos de prueba.

Cuidado, este tipo de documentación no está estandarizada.

2.3.1.13 Otros Listados Útiles

- **Lista de todos los mensajes de error que puede producir el programa**: esta debería poder ser provista por el sistema. La misma debió ser probada cuando se puso el programa en situación de error y comprobado si los mensajes eran correctos, apropiados y si el programa se recuperaba de los mismos. Se debe considerar que la etapa de manejo de errores es una buna ocasión para detectar fallas y defectos del programa.
- **Lista de hardware compatible** (procesadores, memoria, scanners, cualquier dispositivo que soporte).
- **Lista de software compatible**: programas que interactúan con el sistema describiendo interfaces y áreas compatibles.
- **Lista de sistemas operativos compatibles** (utilidades, interfaces, drivers, etc.).

NOTA

No generar demasiado papel. Seleccionar los documentos que interesen.

En la Actividad 3 se muestran un conjunto mínimo de plantillas para registrar los casos de prueba diseñados, para registrar los errores y las iteraciones y para poder realizar el seguimiento de errores.

2.3.2 Error Reproducible

Un error es reproducible únicamente si alguien es capaz de repetir lo que se ha hecho para detectarlo y obtener de esa forma una respuesta inadecuada. En definitiva, debe ser posible explicar la forma de llegar a cierto estado conocido y, desde ese estado, especificar una serie de pasos exactos que muestran el problema.

No debería restringirse el tiempo que se emplea en descubrir y describir los pasos que llevan a un error. En caso de que éstos sean complejos pueden dividirse en una serie. En el análisis se deben detallar:

- Consecuencias más serias del problema.
- Condiciones más simples, cortas y generales de disparar el error.
- Caminos alternativos para el mismo error.
- Problemas relacionados.
- Consecuencias más graves del error.

Disparar un error puede llevar muchos pasos. Esto puede hacer pensar que es tan complicado que el usuario no lo va a percibir. Se puede contrarrestar esta impresión mostrando varios caminos al mismo problema, lo que indica que el problema es aún más grave.

Muchas veces, los problemas están relacionados o reproducidos en otras partes del software, por lo tanto se deben buscar otros lugares donde se pueda hacer algo similar a lo que produjo un error.

2.3.3 Hacer Reproducible un Error

Un error irreproducible difícilmente puede ser corregido. Muchas veces éstos son producto de situaciones descontroladas en el ambiente en el que se ejecuta el programa y son totalmente aleatorios. Aún así, la gran mayoría de los errores encontrados pueden ser reproducidos en ambientes controlados donde puede hacerse el seguimiento correspondiente para aislar la causa del error.

Para hacer un error reproducible se debe documentar:

- Como llegar a cierto estado y como disparar el error.
- Escribir todo lo que se recuerde haber hecho la primera vez e intentar reproducir el error.

Si una vez obtenido el error no se puede reproducir, habrá que considerar algunas hipótesis:

- Condiciones de velocidad: generalmente se ejecuta la prueba por segunda vez mucho más lento. El error pudo producirse porque el programa no puede trabajar tan rápido. Volver a realizar la prueba de forma rápida.
- Detalles olvidados: aplicar un plan paso-a-paso.
- Errores de usuario: no se hizo lo que se pensó haber hecho.
- Un efecto del error hace imposible la reproducción.
- Los datos pueden haber sido modificados por la aplicación, utilizar siempre una copia de los datos.

- El error depende del estado de la memoria: ver la cantidad de memoria, la fragmentación, etc. puede ayudar a encontrar la condición.
- Es un error de estado inicial.
- El error se ha basado en datos corruptos.
- El error es un efecto lateral de algún otro problema.
- Dependencia temporal: el error puede depender de la fecha u hora en la que se realiza la prueba.
- Dependencia de los recursos.
- Errores que no tiene un impacto inmediato.
- Casos especiales en el código: la ayuda del programador puede ahorrar mucho tiempo.

2.3.4 Análisis de un Error Reproducible

Al igual que ocurre con otras áreas de trabajo, como la medicina o la mecánica, los problemas no se detectan sino por medio de sus síntomas. Los síntomas menores son a veces la primera manifestación del error.

En el caso de los sistemas, el diagnóstico de las causas de un error deben centrarse en la búsqueda de información a partir de:

- Mensajes de error.
- Retardos en el procesamiento.
- Parpadeos en la pantalla.

- Saltos del cursor.
- Múltiples cursores.
- Texto no alineado.
- Caracteres dobles u omitidos.
- Luz de uso cuando el dispositivo no está en uso.

Cuanto más aspectos del comportamiento del programa puedan hacerse visibles, más cosas se podrán ver y con mayor probabilidad se encontrará el paso crítico, maximizando así la visibilidad de su comportamiento.

Las técnicas de localización de errores incluyen el uso de depuradores, utilizar un terminal lento, llevar el registro de todos los accesos a disco, pantalla, etc.

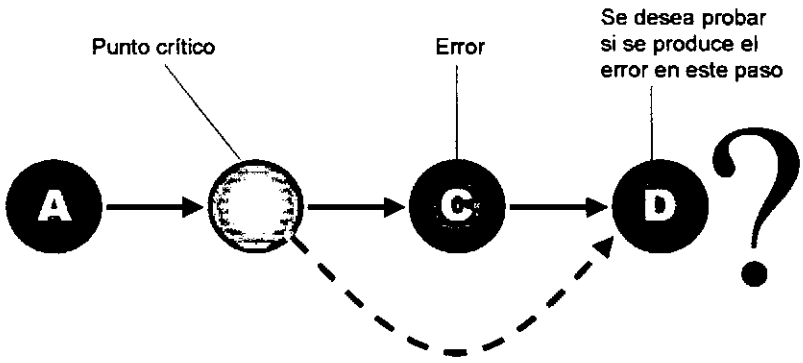


Figura 11 - Búsqueda de Errores Siguientes

Una vez que se ha encontrado el paso crítico, intentar variar el comportamiento para buscar errores siguientes. Por ejemplo, si se sabe que la secuencia A

entonces B entonces C provoca un error en C, y se sabe que el paso crítico está en B, puede intentarse A entonces B entonces D (ver Figura 11).

Incluso, si no se conoce el paso crítico, una vez encontrado el error, seguir ejecutando el programa. Si se detecta la existencia de errores siguientes, se deberá considerar que pueden estar derivados del primero (no serán reproducibles una vez subsanado el primero). Por otra parte, se puede asumir que no son consecuencia directa (se deben probar diferentes caminos).

Los pasos que conducen al problema deberían ser variar o anulados progresivamente. Si el problema es complejo, probar que ocurre al suprimir o cambiar algunos pasos, incluso se pueden omitir algunos pasos.

Un buen indicador temporal del origen del error es verificar versiones anteriores, lo que nos puede indicar que el error ha sido producido por un cambio.

2.3.5 Seguimiento de Problemas

- Generación del informe del problema.
- Informe va al director del proyecto.
    - Una vez que el informe del error está en la base de datos, una copia irá al director del proyecto.
    - El director del proyecto prioriza el problema y realiza una de las siguientes tareas:

- En la mayoría de los casos añade algún comentario, lo prioriza y lo pasa a los programadores.
- Puede intentar reproducir el problema. Si no lo consigue devuelve el informe para su revisión.
- Puede devolver el informe incluso sin intentarlo, porque necesita información de configuración, clarificación, etc. Se debe considerar que:
  - El tiempo de un testeador es generalmente más barato que el de un programador.
  - Las tareas de un programador son con frecuencia puntos críticos al final de un proyecto.
  - En algunos proyectos los testeadores son depuradores más experimentados que los programadores.
  - Nunca es apropiado gastar deliberadamente tiempo de alguien más.

  - El director puede aplazar el informe o marcarlo "Como se diseño". También puede consultar al testeador para que lo considere un tema de documentación.

- El director envía el informe a los programadores.

- Cuando un informe va al programador el director está pidiendo una solución o la investigación y explicación de porque no puede ser solucionado. Generalmente se soluciona.
- El programador puede preguntar por más información, argumentar la imposibilidad de reproducirlo, que es demasiado complicado de solucionar, que no es un error, que es algo que sólo un idiota llegaría a ejecutar, que es un caso de prueba injusto o que no es digno de consideración.

- Cuando un problema es (supuestamente) resuelto.
  - Cuando un programador soluciona un error lo marca como corregido en la base de datos y posiblemente añade algún comentario.
  - Este no es el final del informe. El programador con frecuencia está equivocado. Un ratio del 10% de fallos es muy bueno. En sistemas grandes se ha llegado a tener un 80%.
  - La persona ideal para volver a probar el error es el testeador que lo descubrió. Si es un error que no descubrió un testeador se deberá intentar reproducirlo en la versión en la que se detectó.
  - Si la corrección pasa la prueba inicial se probarán algunas variaciones para comprobar efectos laterales. Si falla la misma prueba marcar el informe como pendiente.

- Problemas irreproducibles

- Si el programador y el director no pueden reproducir el problema lo marcarán como irreproducible y lo reenviaran al testeador.
- Si se consigue recrear el problema (usando preferentemente la versión actual) ponerlo en el comentario. Añadir detalles y hacer una demostración si es necesario.
- Si no se puede recrear en la versión actual y si en una versión anterior se puede marcar como corregido.
- Si no se puede recrear en ninguna versión confirmar que es irreproducible pero mantener el informe abierto.

- Aplazamientos y proceso de apelación.
  - Un aplazamiento admite que existe un problema, pero el director elige no solucionarlo en esta versión del producto.
  - El riesgo de efectos colaterales y un límite en el refinamiento del diseño hacen necesario parar la corrección de errores unas semanas antes de la prueba final.
  - Los motivos del aplazamiento pueden concretarse en el campo de comentarios, siendo útil para la siguiente versión del producto.
  - Algunos directores marcan los problemas como "cómo se diseño" en vez de aplazado. Es una forma de ocultar los errores. Puede haber discrepancias sobre la clasificación entre director y testeador.

- Cada algunas semanas y más frecuentemente al final del proyecto, el director del proyecto y el testeador principal realizan una revisión de errores aplazados. También pueden revisar los marcados "como se diseñó". En esta reunión se toma la última decisión sobre si un error debe ser aplazado.
- A la reunión pueden acudir invitados que pueden objetar sobre el aplazamiento de algún error. Esto proporciona un mecanismo de apelación

- Problemas que no se han tratado.
  - Algunos informes se pierden, otros se separan deliberadamente, otros con baja prioridad son olvidados. Regla importante: todos deben ser resueltos antes de que el producto se publique.
  - Debe ponerse en circulación un informe resumen, cada una o dos semanas, que liste los errores pendientes para evitar que se olviden.
- Informe de estado del proyecto.
  - Informe manejable que indica cuantos errores se han encontrado, cuantos están pendientes, cuantos han sido aplazados comparado con corregidos, etc.
  - Muestra el progreso de cada semana y los totales, ayudando a los directores a evaluar la calidad del esfuerzo de programación, la fiabilidad, la efectividad de las pruebas, etc.

INFORME DEL DESARROLLO DE LA ACTIVIDAD N° 3

1 ENUNCIADO DE LA ACTIVIDAD

Generación de plantillas para la documentación de los resultados obtenidos de la ejecución de procedimientos e implementación de las estrategias desarrolladas en la actividad 2.

2 DESARROLLO

En la Actividad 2 de este proyecto se desarrolló la estrategia para realizar un proceso de testing a un sistema que se va a poner en producción. A medida que se va avanzando en el proceso de testing, es necesario ir plasmando en algún lugar los resultados que se van obteniendo. Al no contar con documentos estándares para el plasmado de resultados, se hace incontrolable el proceso de Testing en la medida que se vaya avanzando en iteraciones del desarrollo y versionado de documentos.

En ésta actividad, se desarrollarán varias plantillas para tener uniformidad, no sólo para un solo sistema, sino en todos los proyectos de desarrollo de software a futuro que se implementen. Las mismas permiten registrar, verificar y aprobar los casos de prueba diseñados, registrar los errores y las iteraciones y realizar el seguimiento de errores.

2.1 PLANTILLA: REGISTRO, REVISIÓN Y APROBACIÓN DE CASOS DE PRUEBA

Esta plantilla sirve para generar un documento uniforme y versionado que permite registrar los casos de prueba definidos para un plan de prueba, realizar todo el proceso de revisión de los mismos hasta llegar a la aprobación.

La versión del documento aprobada es la que finalmente se ejecuta. El documento también sirve para registrar los errores detectados. Para continuar con el registro de errores en forma más detallada, a este documento se le asocia el que se genera con la plantilla 2.2.

Este documento permite también llevar las iteraciones de las distintas ejecuciones del plan de prueba.

PROYECTO	PLAN DE PRUEBAS		
NOMBRE:	ID:	VERSION:	ITERACION:

DATOS DE CREACIÓN		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

DATOS DE REVISIÓN		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

DATOS DE APROBACIÓN		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

DATOS DEL TESTER		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

DATOS DE LA PRUEBA
TIPO DE PRUEBA: ☐ FUNCIONAL ☐ NO FUNCIONAL ☐ OTRA: _____ NIVEL DE TESTING: ☐ UNIDAD ☐ INTEGRACIÓN ☐ SISTEMA ☐ ALFA ☐ BETA ☐ ACEPTACIÓN SISTEMA: SUBSISTEMA: MÓDULO:

OBSERVACIONES

PÁGINA 1

PROYECTO	PLAN DE PRUEBAS		
NOMBRE:	ID:	VERSION:	ITERACION:

CASOS DE PRUEBA

CASO DE PRUEBA	ELEMENTO	COMP.	FUNCIÓN	PANTALLA	CAMPO	BOTÓN	PROPIEDAD(ES)	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
									Val.	Inv.	Val.	Inv.	Val	Inv.	Val.	Inv.

PÁGINA 2

PROYECTO	PLAN DE PRUEBAS		
NOMBRE:	ID:	VERSION:	ITERACION:

PROYECTO
- NOMBRE: Nombre del proyecto de Software

PLAN DE PRUEBAS
- ID: Identificación del documento. Esta identificación de documentos de Plan de Pruebas debe ser única en el proyecto.
- VERSIÓN: Número de Versión del Documento.
- ITERACIÓN: Número de iteración dentro de la fase de la prueba.

DATOS DE CREACIÓN
- APELLIDO Y NOMBRE: Nombre y apellido del /de los autor/es del documento.
- ORGANIZACIÓN: Organización a la que pertenece/n la/s persona/s.
- TELEFONO: Teléfono de la organización.
- E-MAIL: E-mail de la persona que realizó la creación del documento.
- FECHA: Fecha de creación del documento.

DATOS DE REVISIÓN
- APELLIDO Y NOMBRE: Nombre y apellido del /de los revisor/es del documento.
- ORGANIZACIÓN: Organización a la que pertenece/n la/s persona/s.
- TELEFONO: Teléfono de la organización.
- E-MAIL: E-mail de la persona que realizó la revisión del documento.
- FECHA: Fecha de revisión del documento.

DATOS DE APROBACIÓN
- APELLIDO Y NOMBRE: Nombre y apellido del /de los aprobador/es del documento.
- ORGANIZACIÓN: Organización a la que pertenece/n la/s persona/s.
- TELEFONO: Teléfono de la organización.
- E-MAIL: E-mail de la persona que realizó la aprobación del documento.
- FECHA: Fecha de aprobación del documento.

PÁGINA 3

PROYECTO	PLAN DE PRUEBAS		
NOMBRE:	ID:	VERSION:	ITERACION:

DATOS DEL TESTER

- APELLIDO Y NOMBRE: Nombre y apellido del /de los testeador/es del documento.
- ORGANIZACIÓN: Organización a la que pertenece/n la/s persona/s.
- TELEFONO: Teléfono de la organización.
- E-MAIL: E-mail de la persona que realizó el Plan de Pruebas.
- FECHA: Fecha de comienzo de ejecución del Plan de Pruebas.

DATOS DE LA PRUEBA

- TIPO DE PRUEBA: Tipo de prueba que se realiza.
- NIVEL DE TESTING: Nivel o fase en la que se encuentra el testing del sistema.
- SISTEMA: Nombre del sistema que se está probando.
- SUBSISTEMA: Nombre del subsistema que se está probando.
- MÓDULO: Modulo o funcionalidad que se está probando.

OBSERVACIONES

- Comentarios adicionales sobre el documento.

CASOS DE PRUEBA

- CASO DE PRUEBA Númeración del caso de prueba. Esta numeración debe ser única en todo el proyecto.
- ELEMENTO: Elemento del sistema que se está probando.
- COMP: Componente del sistema que se está probando.
- FUNCIÓN: Función dentro del módulo del sistema que se está probando.
- PANTALLA: Pantalla dentro del módulo del sistema que se está probando.
- CAMPO: Campo dentro del módulo del sistema que se está probando.
- BOTÓN: Botón dentro del módulo del sistema que se está probando.
- PROPIEDADES: Propiedades del Botón / Campo / Función / Pantalla.
- SECUENCIA DE PASOS: Pasos que se realizan para la prueba.
- DATOS DE ENTRADA: Datos que se van a cargar para realizar la prueba.
  - Válidos: Datos de Entrada Válidos.
  - Inválidos: Datos de Entrada Inválidos.

PÁGINA 4

PROYECTO	PLAN DE PRUEBAS		
NOMBRE:	ID:	VERSION:	ITERACION:

- RESULTADO ESPERADO: Datos que debería esperarse del sistema.
  - Válidos: Resultado esperado para datos de entrada Válidos.
  - Inválidos: Resultado esperado para datos de entrada Inválidos.
- DATOS DE SALIDA: Datos entregados por el sistema.
  - Válidos: Datos de Salida para datos de entrada Válidos.
  - Inválidos: Datos de Salida para datos de entrada Inválidos.
- ERROR: Afirmación o negación si hubo o no error.
  - Válidos: Afirmación o negación si hubo o no error para datos válidos.
  - Inválidos: Afirmación o negación si hubo o no error para los datos inválidos.

PÁGINA 5

2.2 PLANTILLA: CAPTURAS DE PANTALLAS

El documento que permite generar esta plantilla sirve para registrar en forma gráfica los errores que se detectan producto de la ejecución de un caso de prueba de un plan de prueba específico

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE:	

DATOS DEL TESTER		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

DATOS DE LA PANTALLA	
SISTEMA: MÓDULO:	SUBSISTEMA: FUNCIÓN:

PLAN DE PRUEBAS ASOCIADO		
ID:	VERSION:	ITERACION:

OBSERVACIONES

PÁGINA 1

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE:	

CASO DE PRUEBA ASOCIADO

IMAGEN

PÁGINA 2

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE:	

PROYECTO
- NOMBRE: Nombre del proyecto de Software.

DATOS DEL TESTER
- APELLIDO Y NOMBRE: Nombre y apellido del /de los tester/s del documento.
- ORGANIZACIÓN: Organización a la que pertenece/n la/s persona/s.
- TELEFONO: Teléfono de la organización.
- E-MAIL: E-mail de la persona que realizó la Captura de pantallas.
- FECHA: Fecha de Captura de pantallas.

DATOS DE LA PANTALLA
- SISTEMA: Nombre del sistema que se está probando a la cual pertenecen las pantallas.
- SUBSISTEMA: Nombre del subsistema que se está probando a la cual pertenecen las pantallas.
- MÓDULO: Modulo o funcionalidad que se está probando a la cual pertenecen las pantallas.
- FUNCIÓN: Función a la cual pertenece la pantalla a la cual pertenecen las pantallas.

PLAN DE PRUEBAS ASOCIADO
- ID: Identificación del documento de Plan de Pruebas asociado con las pantallas.
- VERSIÓN: Número de versión del documento de Plan de Pruebas.
- ITERACIÓN: Número de iteración dentro de la fase de la prueba.

OBSERVACIONES
- Comentarios acerca de las pantallas capturadas.

CASO DE PRUEBA ASOCIADO
- Número del caso de prueba a la cual están asociadas las pantallas.

IMAGEN
- Imagenes de pantallas capturadas.

PÁGINA 3

2.3 PLANTILLA: SEGUIMIENTO DE ERRORES

Esta plantilla permite generar un documento de seguimiento de errores, que se confecciona a partir de los dos anteriores.

PROYECTO	SEGUIMIENTO DE ERRORES
NOMBRE:	

DATOS DEL VALIDADOR		
APELLIDO Y NOMBRE: ORGANIZACIÓN:	TELÉFONO: E-MAIL:	FECHA: / /

OBSERVACIONES

DETALLES DE ERRORES

CASO DE PRUEBA	TIPO ERROR	DETALLE	ITEMS AFECTADO	SECUENCIA DE PASOS	OCURR.	REPROD.	PRIOR.	IMPACTO	DATOS EN QUE SE PRODUCE ERROR	FECHA RESOLUCIÓN

PÁGINA 1

PROYECTO	SEGUIMIENTO DE ERRORES
NOMBRE:	

PROYECTO
- NOMBRE: Nombre del proyecto de Software

DATOS DEL VALIDADOR
- APELLIDO Y NOMBRE: Nombre y apellido del /de los validador/es del documento
- ORGANIZACIÓN: Organización a la que pertenece/n las personas
- TELEFONO: Teléfono de la organización
- E-MAIL: E-mail de la persona que realiza la validación
- FECHA: Fecha de creación del documento

OBSERVACIONES
- Comentarios adicionales acerca del documento

CASOS DE PRUEBA
- CASO DE PRUEBA: Número del Caso de Prueba donde se presentó el error.
- TIPO ERROR: Tipo del error detectado (mejora, error, nuevo requerimiento)
- DETALLE: Detalle del error encontrado
- ITEMS AFECTADO: Items que afecta del proyecto
- SECUENCIA DE PASOS: secuencia de pasos utilizada
- OCURR.: Tipo de ocurrencia del error (nunca, siempre, frecuente)
- REPROD.: ¿Es el error reproducible? (si, no)
- PRIOR.: Prioridad en la corrección del error (alta, media, baja)
- IMPACTO: impacto que causa el error (alto, medio, bajo)
- DATOS EN QUE SE PRODUCE ERROR: Datos de entrada que provocaron el error
- FECHA RESOLUCIÓN: Fecha en que el desarrollador corregirá el error

PÁGINA 2

INFORME DEL DESARROLLO DE LA ACTIVIDAD N° 4

1 ENUNCIADO DE LA ACTIVIDAD

Evaluación del trabajo realizado mediante la aplicación en un caso práctico de las técnicas y estrategias descriptas. Se trabajará con un aplicativo que la S.E.T.I. considere conveniente.

2 DESARROLLO

La aplicación de las técnicas recomendadas en el presente trabajo fueron realizadas, a pedido de la S.E.T.I., sobre el aplicativo llamado "Sistema de Capacitación".

De este aplicativo se procedió a realizar testing de los subsistemas "Administración de Cursos", "Consultas y Listados" y "Parámetros".

Sobre estos subsistemas se realizaron pruebas funcionales para un nivel alfa de testing con los objetivos que se listan a continuación:

- Probar que las funcionalidades, bases de datos, opciones de búsqueda, menús, íconos y links cumplan las especificaciones de los requerimientos de usuario.

- Asegurar que el funcionamiento de los botones definidos por omisión sea correcto y que los botones de comando no aparezcan (o estén deshabilitados) cuando no deban utilizarse.
- Verificar que los campos de los formularios tengan controles de contenido de datos.
- Probar en la base de datos: resultados de búsqueda, integridad de datos, etc.
- Evaluar que la creación, modificación y eliminación de datos en tablas se realice tal como se lo especifica en las funcionalidades.
- Chequear que los valores por omisión sean almacenados en la base de datos si el usuario no los especifica.
- Chequear que puedan abrirse múltiples instancias de la aplicación al mismo tiempo.
- Probar cada barra de herramientas, ítem de menú, navegación de las ventanas para comandos utilizando mouse y teclado.
- Ver que los colores, fuentes y tamaño de fuente sean estándares y coherentes a lo largo de la aplicación.
- Verificar que todos los campos alfanuméricos tengan alineación a la izquierda y que los campos numéricos estén alineados a la derecha.
- Son inaceptables los errores ortográficos a lo largo de la aplicación (texto mostrado en las ventanas: título, opciones en la barra de estado, campos de texto, textos emergentes y mensajes del sistema).

El "Criterio de finalización" del testing alfa del sistema de Capacitación se define como:

Se ejecutarán, con sus correspondientes iteraciones, todos los casos de prueba hasta que los mismos ya no detecten errores de acuerdo a los objetivos planteados para el mismo en este nivel.

Si se encuentran más de 3 errores de tipo no aceptable como: errores funcionales, de integridad de datos, de control de campos, de especificaciones de requerimientos, etc. , el director de testing puede determinar que la prueba no ha sido superada o la misma se encuentra inconclusa, fundamentando que el tipo de errores descubiertos debieron ser superados en niveles de testing inferiores.

Para llevar a cabo el testing del sistema de capacitación. se utilizó una copia de una base de datos previamente cargada y se ejecutó el mismo en un ambiente de prueba, que la S.E.T.I. dispone para tales fines.

En el Anexo 1 puede consultarse el documento de registro de casos de prueba y en el Anexo 2 se encuentra un registro de las pantallas con error halladas a lo largo de las pruebas. La información presente en ambos anexos, debería servir como ejemplo de uso de las plantillas a aquellas personas que deseen realizar testing de sistemas basándose en la metodología propuesta.

En general, a partir de las pruebas realizadas, se pudo determinar la existencia de errores que debieron ser superados en niveles de testing anteriores, como por ejemplo:

- No hay restricciones de integridad en la base de datos.

- Faltan controles de contenidos de datos en muchos campos.
- Poca claridad en los mensajes de error, existencia de errores ortográficos y de mensajes de capas inferiores de la aplicación sin capturar.
- El sistema se encuentra limitado al uso de empleados del Gobierno de la Provincia de San Luis, lo que quiere decir que ningún tercero puede dictar o asistir a los cursos.
- No se encontró unicidad en los códigos de cursos, exámenes, capítulos, etc., o sea, se observaron códigos repetidos.

Todo esto, sumado a que la cantidad de errores encontrados superaron ampliamente el umbral establecido para la aceptación de la versión alfa, fue el determinante para la finalización del testing nivel alfa. Los errores reportados, deberán ser corregidos y se iniciará un nuevo ciclo de testing como el que se realizó, con su correspondientes iteraciones, para recién aceptar la versión alfa y pasar a otros niveles de testing superiores.

Cabe destacar que la versión beta del sistema de capacitación ya había sido testeado anteriormente, en forma artesanal, por los usuarios finales, y la misma había sido aprobada para la puesta en producción. Con este proyecto se quiere resaltar la importancia de llevar a cabo un testing de forma metódica, el resultado del mismo está a la vista.

ANEXO 1: BIBLIOGRAFÍA

- Roger S. Pressman, "Ingeniería del Software", 3ra ed, McGraw-Hill, 1996.
- DPTOIA-IT-2001-001, "Verificación con XML, Informe Técnico", Departamento de Informática y Automática - Universidad de Salamanca, Noviembre de 2001.
- Fernández, J. M., "Prueba de Software Basado en Componentes. Estado Actual", Informe Técnico N° 28 Serie Verde, CIC, IPN. 1999.
- Myers, G., "The Art of Software Test", Wiley, 1979.
- Velasco Elizondo, Perla Inés, "Prueba de Componentes de Software Basadas en el Modelo de Javabeans", Departamento de Ingeniería Y Tecnología - Unidad de Estudios de Posgrado - Universidad Autónoma De Tlaxcala, Tesis, Abril de 2001.
- "El Proceso Unificado de Desarrollo de Software" – Autores: Jacobson, Booch, Rumbaugh – Editorial: Addison-Wesley Iberoa
- "Ingeniería del Software III", Material de apoyo publicado en la página de Internet: http://www.ldc.usb.ve
- "Pruebas del Software", Material de apoyo publicado en la página de Internet: www.dccia.ua.es/dccia/inf/asignaturas/ IS2/teoria/tema8S.pdf
- "A Software Engineering View of Data Quality", Trabajo publicado en el sitio de Internet: http://www.dc.uba.ar/people/proyinv/arte/trabajos.htm

- "Ingeniería del Software II", Material de apoyo publicado en la página de Internet: http://www.info-ab.uclm.es
- "Practical UML", Material publicado en la dirección de Internet: http://www.togethersoft.com/services/practical_guides/umlonlinecourse/
- http://www.terra.es/personal/joanmmas/
- http://www.will-harris.com
- http://www.sitecritique.net/
- www.intrusion.com
- www.rsa.com
- www.digicash.com
- Transacciones electrónicas seguras:
    - www.verisign.com,
    - www.cylink.com,
    - www.terisa.com,
    - www.cybercash.com,
    - www.checkfree.com

Lecturas Recomendadas

- Bashir, I. y Goel A. "Testing Object-Oriented Software. Life Cycle Solutions".Springer-Verlag New York, Inc.1999.

- Beizer, B., "Software System Testing and Quality Assurance", Van Nostrand Reinhold, 1984.
- Beizer,B., "Sodtware Testing Techniques", 2da ed., Van Nostrand Reinhold, 1990.

ANEXO 2: ÍNDICE DE FIGURAS

ANEXO 3: ÍNDICE DE TABLAS

ANEXO 4: CASOS DE PRUEBA – SISTEMA DE CAPACITACIÓN

DATOS DE CREACIÓN		
APELLIDO Y NOMBRE: **Fabricio Borda**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **fborda@sanluis.gov.ar**	FECHA: **01 / 11 / 02**

DATOS DE REVISIÓN		
APELLIDO Y NOMBRE: **Cristian Mendoza**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **cmendoza@sanluis.gov.ar**	FECHA: **11 / 11 / 02**

DATOS DE APROBACIÓN		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **14 / 11 / 02**

DATOS DEL TESTER		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **16 / 11 / 02**

DATOS DE LA PRUEBA

TIPO DE PRUEBA: ☒ **FUNCIONAL** ☐ NO FUNCIONAL ☐ OTRA: _____
NIVEL DE TESTING: ☐ UNIDAD ☐ INTEGRACIÓN ☐ SISTEMA ☒ **ALFA** ☐ BETA ☐ ACEPTACIÓN
SISTEMA: **CAPACITACIÓN**
SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS**
MÓDULO: **ABM CURSOS – HORAS – DICTADO – CAPÍTULOS – EXÁMEN – EXÁMEN PRESENCIAL**

OBSERVACIONES

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val.	Inv
CP-EG-SC-TA-001		Buscar Curso	Código		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos	Código completo, vacío, parte del código		Listado con datos que se ajustan al Código ingresado		Listado con datos que se ajustan al Código ingresado			
			Título		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos	Título completo, vacío, parte del título		Listado con datos que se ajustan al Título ingresado		Listado con datos que se ajustan al Título ingresado			
			Área		Alfanumérico, valida contra listado de áreas previamente cargadas en otro sistema	Capacitación/Administración de cursos/AMB_Cursos	Título completo, vacío, parte del título		Listado con datos que se ajustan al Área ingresada		Listado con datos que se ajustan al Área ingresada			
			Tema		Alfanumérico, lista desplegable de temas cargado previamente	Capacitación/Administración de cursos/AMB_Cursos	Tema completo, vacío, parte del título		Listado con datos que se ajustan al Tema ingresado		Listado con datos que se ajustan al Tema ingresado			
				Buscar		Capacitación/Administración de cursos/AMB_Cursos			Busca según los parámetros ingresados		Busca según los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/AMB_Cursos			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Nuevo		Capacitación/Administración de cursos/AMB_Cursos			pantalla de nuevo Curso		pantalla de nuevo Curso			
				Último		Capacitación/Administración de cursos/AMB_Cursos/Buscar			Avanza al último registro		Avanza al último registro			
				Siguiente		Capacitación/Administración de cursos/AMB_Cursos/Buscar			Avanza al siguiente registro		Avanza al siguiente registro			
				Refrescar		Capacitación/Administración de cursos/AMB_Cursos/Buscar			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Nuevo		Capacitación/Administración de cursos/AMB_Cursos/Buscar			pantalla de nuevo Curso		pantalla de nuevo Curso			
				Código (links de Listado de Cursos)		Capacitación/Administración de cursos/AMB_Cursos/Buscar			Accede al registro que indica el código		Accede al registro que indica el código			
CP-EG-SC-TA-002		Nuevo curso	Código		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Nuevo		Cargar el mismo código de curso para varios cursos		Cartel de aviso que el código ya existe (error)		códigos de cursos repetidos		error
			Título		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Nuevo								
			Área		Listado de áreas previamente cargadas en otro sistema	Capacitación/Administración de cursos/AMB_Cursos/Nuevo								
			Tema		Lista desplegable de temas cargado previamente	Capacitación/Administración de cursos/AMB_Cursos/Nuevo								
			Tipo de curso		Lista desplegable de tipos cargados previamente	Capacitación/Administración de cursos/AMB_Cursos/Nuevo								
			Puntaje		Real	Capacitación/Administración de cursos/AMB_Cursos/Nuevo	3,5	Valores menores a 0	3,5	Cartel de valores incorrectos	Cartel de valores incorrectos	Cartel de valores incorrectos	error	
			Duracion (cantidad)		Real	Capacitación/Administración de cursos/AMB_Cursos/Nuevo		Valores menores a 0, controlar que la diferencia fecha hasta - fecha desde sea coherente (mayor) que lo especificado en los canpos de Duración		Cartel de valores incorrectos, Cartel de aviso		Cartel de valores incorrectos, no hay aviso		error
			Duración (tipo)		Lista desplegable de tipo de duración cargados previamente	Capacitación/Administracion de cursos/AMB_Cursos/Nuevo								

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv.	Val.	Inv.	Val.	Inv.	Val.	Inv.
	ABM Cursos		Fecha Desde		Fecha	Capacitación/Administración de cursos/AMB_Cursos/Nuevo		fecha desde > fecha hasta, vacio		Control de contenido de dato para el campo		Nuevo curso sin Fecha Dasta		error
			Fecha Hasta		Fecha	Capacitación/Administración de cursos/AMB_Cursos/Nuevo		fecha hasta < fecha desde vacio		Control de contenido de dato para el campo		Nuevo curso sin Fecha Hasta		error
			Descripción		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Nuevo								
				Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Lista de valores (Para el campo Área)		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Ventana de selección		Ventana de selección			
				Lista de valores (Para el campo Tipo de Curso)		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Ventana de selección		Ventana de selección			
				Continuar		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Nuevo Curso cargado		Nuevo Curso cargado			
				Deshacer		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/AMB_Cursos/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-003		Ingresar Autor	Autores		Listado de personas previamente cargadas en otro sistema	Capacitación/Administración de cursos/AMB_Cursos/Nuevo/Nuevo		Ingresar para un curso dos o más veces el mismo autor		Cartel de aviso que el autor ya existe (error)		Autores repetidos		error
				Nuevo		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/Nuevo			Pantalla de nuevo Autor		Pantalla de nuevo Autor			
				Lista de valores (Para el campo Autores)		Capacitación/Administracion de cursos/AMB_Cursos/Nuevo/Nuevo			Ventana de selección		Ventana de selección			
				Continuar		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/Nuevo			Dá de alta el autor con los datos cargados		Dá de alta el autor con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-004		Consultar o Borrar Autor		Lista de valores (para el campo curso)		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/código de autor			Ventana de selección		Ventana de selección			
				Borrar		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/código de autor			Elimina el autor		Elimina el autor			
				Volver		Capacitación/Administración de cursos/AMB_Cursos/Nuevo/código de autor			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
			Código		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso		Borrar un curso con luminos cursando		Mensaje de error		Permitido		error
			Título		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso								
			Área		Listado de areas previamente cargadas en otro sistema	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso								
			Tema		Lista desplegable de temas cargado previamente	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso								
			Tipo de curso		Lista desplegable de tipos cargados previamente	Capacitación/Administracion de cursos/AMB_Cursos/Buscar/código de curso								

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val.	Inv
CP-EG-SC-TA-005		Actualizar o Borrar Curso	Puntaje		Real	Capacitación/Administración de cursos/AMB_Cursos/Buscar/codigo de curso	3,5	Valores menores a 0	3,5	Cartel de valores incorrectos	Cartel de valores incorrectos	Cartel de valores incorrectos	error	
			Duración (cantidad)		Real	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso		Valores menores a 0		Cartel de valores incorrectos		Cartel de valores incorrectos		
			Duración (tipo)		Lista desplegable de tipo de duración cargados previamente	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso								
			Fecha Desde		Fecha	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso		fecha desde > fecha hasta, vacio		Control de contenido de dato para el campo		Actualización de curso sin Fecha Dasta		error
			Fecha Hasta		Fecha	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso		fecha hasta < fecha desde, vacio		Control de contenido de dato para el campo		Actualización de curso sin Fecha Hasta		error
			Descripción		Alfanumérico	Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso								
				Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Ventana de selección de calendario		Ventana de selección de calendario			
				Lista de valores (Para el campo Área)		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Ventana de selección		Ventana de selección			
				Lista de valores (Para el campo Tipo de Curso)		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Ventana de selección		Ventana de selección			
				Actualizar		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Carga los datos originales del curso		Carga los datos originales del curso			
				Deshacer		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Borrar		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Borra el curso		Borra el curso			
				Volver		Capacitación/Administración de cursos/AMB_Cursos/Buscar/código de curso			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-006	Horas	Consultar Hora		Nuevo		Capacitación/Administración de cursos/Horas			Pantalla de nueva Hora		Pantalla de nueva Hora			
				Tipo de Hora (link sobre el campo)		Capacitación/Administración de cursos/Horas			Pantalla de la hora con los datos del registro seleccionado	Controlar que la suma de la cantidad de horas de los registros no supere el campo duración del curso	Pantalla de la hora con los datos del registro seleccionado	La suma es superada ampliamente		error
CP-EG-SC-TA-007		Nueva Hora	Tipo de Horas		Lista desplegable de tipos de horas cargados previamente	Capacitación/Administración de cursos/Horas/Nuevo								
			Cantidad		Real	Capacitación/Administración de cursos/Horas/Nuevo	85,5	Valores menores a 0	85,5	Cartel de valores incorrectos	Cartel de valores incorrectos	Cartel de valores incorrectos	error	
			Observación		Alfanumérico	Capacitación/Administración de cursos/Horas/Nuevo								
				Continuar		Capacitación/Administración de cursos/Horas/Nuevo			Dá de alta la hora con los datos cargados		Dá de alta la hora con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Horas/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/Horas/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv	Val	Inv	Val	Inv	Val.	Inv
				Código de curso		Capacitación/Administración de cursos/Horas/Nuevo			pantalla del curso seleccionado		pantalla del curso seleccionado			
CP-EG-SC-TA-008		Actualizar o Borrar Hora	Tipo de Horas		Alfanumérico, lista desplegable de tipos de horas cargados previamente	Capacitación/Administración de cursos/Horas/Tipo de hora								
			Cantidad		Real	Capacitación/Administración de cursos/Horas/Tipo de hora		Valores menores a 0		Cartel de valores incorrectos				
			Observación		Alfanumérico	Capacitación/Administración de cursos/Horas/Tipo de hora								
				Actualizar		Capacitación/Administración de cursos/Horas/Tipo de hora			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Borrar		Capacitación/Administración de cursos/Horas/Tipo de hora			Borra la hora		Borra la hora			
				Volver		Capacitación/Administración de cursos/Horas/Tipo de hora			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Código de curso		Capacitación/Administración de cursos/Horas/Tipo de hora			Pantalla del curso seleccionado		Pantalla del curso seleccionado			
CP-EG-SC-TA-009	Dictado	Consultar Fechas de Dictado de Cursos		Código de Fecha		Capacitación/Administración de cursos/Dictado			Fechas de inicio distintas; Fechas iguales a las del curso		Fechas de inicio iguales; Fechas distintas a las del curso		error	
				Refrescar		Capacitación/Administración de cursos/Dictado			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Dictado			Pantalla de nuevo dictado		Pantalla de nuevo dictado			
CP-EG-SC-TA-010		Nuevas Fechas de Dictado de Cursos	Fecha de Inicio		Fecha	Capacitación/Administración de cursos/Dictado/Nuevo		fecha de inicio > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Fecha de Finalización		Fecha	Capacitación/Administración de cursos/Dictado/Nuevo		fecha de finalización < fecha de inicio		Control de contenido de dato para el campo		Permitido		error
			Fecha de Inicio de Inscripción		Fecha	Capacitación/Administración de cursos/Dictado/Nuevo		fecha de inscripción > fecha de finalización de inscripción fecha de inscripción > fecha de inicio fecha de inscripción > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Fecha de Finalización de Inscripción		Fecha	Capacitación/Administración de cursos/Dictado/Nuevo		fecha de finalización de inscripción < fecha de inscripción fecha de finalización de inscripción > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Vacantes		Natural	Capacitación/Administración de cursos/Dictado/Nuevo		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
			Lugar		Alfanumérico	Capacitación/Administración de cursos/Dictado/Nuevo								

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv.	Val	Inv.
			Cantidad de Tutores		Natural	Capacitación/Administración de cursos/Dictado/Nuevo		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
			Horarios		Alfanumérico	Capacitación/Administración de cursos/Dictado/Nuevo								
				Código de Curso		Capacitación/Administración de cursos/Dictado/Nuevo			Pantalla del curso seleccionado		Pantalla del curso seleccionado			
				Calendario para selección de Fecha de Inicio		Capacitación/Administración de cursos/Dictado/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Finalización		Capacitación/Administración de cursos/Dictado/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Inicio de Inscripción		Capacitación/Administración de cursos/Dictado/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Finalización de Inscripción		Capacitación/Administración de cursos/Dictado/Nuevo			Ventana de selección de calendario		Ventana de selección de calendario			
				Continuar		Capacitación/Administración de cursos/Dictado/Nuevo			Dá de alta la fecha de dictado con los datos cargados		Dá de alta la fecha de dictado con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Dictado/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/Dictado/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
			Fecha de Inicio		Fecha	Capacitación/Administración de cursos/Dictado/código de Fecha		fecha de inicio > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Fecha de Finalización		Fecha	Capacitación/Administración de cursos/Dictado/código de Fecha		fecha de finalización < fecha de inicio		Control de contenido de dato para el campo		Permitido		error
			Fecha de Inicio de Inscripción		Fecha	Capacitación/Administración de cursos/Dictado/código de Fecha		fecha de inscripción > fecha de finalización de inscripción fecha de inscripción > fecha de inicio fecha de inscripción > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Fecha de Finalización de Inscripción		Fecha	Capacitación/Administración de cursos/Dictado/código de Fecha		fecha de finalización de inscripción < fecha de inscripción fecha de finalización de inscripción > fecha de finalización		Control de contenido de dato para el campo		Permitido		error
			Vacantes		Nautral	Capacitación/Administración de cursos/Dictado/código de Fecha		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
			Lugar		Alfanumérico	Capacitación/Administración de cursos/Dictado/código de Fecha								

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv.	Val	Inv	Val	Inv	Val.	Inv.
CP-EG-SC-TA-011		Actualizar o Borrar Fechas de Dictado de Cursos	Cantidad de Tutores		Natural	Capacitación/Administración de cursos/Dictado/codigo de Fecha		Valores menores o iguales a 0; Valor del campo menor que la cantidad real		Cartel de valores incorrectos		Fila actualizada con valor de tutores menores que la cantidad real		error
			Horarios		Alfanumérico	Capacitación/Administración de cursos/Dictado/código de Fecha								
				Código de Curso		Capacitación/Administración de cursos/Dictado/código de Fecha			Pantalla del curso seleccionado		Pantalla del curso seleccionado			
				Calendario para selección de Fecha de Inicio		Capacitación/Administración de cursos/Dictado/código de Fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Finalización		Capacitación/Administración de cursos/Dictado/código de Fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Inicio de Inscripción		Capacitación/Administración de cursos/Dictado/código de Fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Calendario para selección de Fecha de Finalización de Inscripción		Capacitación/Administración de cursos/Dictado/código de Fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Actualizar		Capacitación/Administración de cursos/Dictado/código de Fecha			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/Dictado/código de Fecha			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Dictado/código de Fecha			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Dictado/código de Fecha		Borrar con alumnos cursando	Borra la fecha	Rechazo al borrado	Borra la fecha	Fecha eliminada		error
				Nuevo		Capacitación/Administración de cursos/Dictado/código de Fecha			Pantalla de nueva fecha		Pantalla de nueva fecha			
CP-EG-SC-TA-012		Consultar Tutor		Código de Tutor	Fecha	Capacitación/Administración de cursos/Dictado/código de Fecha			Pantalla del Tutor seleccionado		Pantalla del Tutor seleccionado			
CP-EG-SC-TA-013		Ingresar Tutor		Nuevo		Capacitación/Administración de cursos/Dictado/código de Fecha			Pantalla de nuevo tutor		Pantalla de nuevo tutor			
			Apellido		Alfanumérico; valida contralistado de personas previamente cargadas en otro sistema	Capacitación/Administración de cursos/Dictado/código de Fecha/nuevo		Ingresar un mismo tutor varias veces		Mensaje de error		Permitido		error
				Lista de valores para el campo Apellido		Capacitación/Administración de cursos/Dictado/código de Fecha/nuevo			Ventana de selección		Ventana de selección			
				Continuar		Capacitación/Administración de cursos/Dictado/código de Fecha/nuevo			Dá de alta el tutor con los datos cargados		Dá de alta el tutor con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Dictado/código de Fecha/nuevo			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Dictado/código de Fecha/nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-014		Borrar Tutor		Borrar		Capacitación/Administración de cursos/Dictado/código de Fecha/código de tutor			Borra el tutor		Borra el tutor			
				Volver		Capacitación/Administración de cursos/Dictado/código de Fecha/código de tutor			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-015	Capitulos	Capitulos de un Curso		Código (Lista de links de capitulos)		Capacitación/Administración de cursos/Capitulos		Múltiple capítulos con el mismo nombre	Pantalla del Capitulo seleccionado	Mensaje de error	Pantalla del Capitulo seleccionado	Múltiple capítulos con el mismo nombre		error
				Refrescar		Capacitación/Administración de cursos/Capitulos			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Capitulos			Pantalla de nuevo Capitulo		Pantalla de nuevo Capitulo			

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv.	Val.	Inv.	Val.	Inv.	Val.	Inv.
CP-EG-SC-TA-016		Consultar Capitulo		Código de capitulo		Capacitación/Administración de cursos/Capitulos			Pantalla del Capítulo seleccionado		Pantalla del Capítulo seleccionado			
CP-EG-SC-TA-017		Nuevo Capitulo	Código		Alfanumérico	Capacitación/Administración de cursos/Capitulos/nuevo								
			Título		Alfanumérico	Capacitación/Administración de cursos/Capitulos/nuevo								
			Orden		Natural	Capacitación/Administración de cursos/Capitulos/nuevo		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
			Capitulo Asociado		Alfanumérico, valida contra listado de capitulos previamente cargados	Capacitación/Administración de cursos/Capitulos/nuevo								
			Archivo Adjunto		Alfanumérico	Capacitación/Administración de cursos/Capitulos/nuevo								
				Código de capitulo		Capacitación/Administración de cursos/Capitulos/nuevo			Pantalla del Capítulo seleccionado		Pantalla del Capítulo seleccionado			
				Lista de valores (para el campo Capitulo Asociado)		Capacitación/Administración de cursos/Capitulos/nuevo			Ventana de selección		Ventana de selección			
				Examinar (ruta de Upload de archivos)		Capacitación/Administración de cursos/Capitulos/nuevo			Ventana de selección		Ventana de selección			
				Continuar		Capacitación/Administración de cursos/Capitulos/nuevo			Dá de alta el Capitulo con los datos cargados		Dá de alta el Capitulo con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Capitulos/nuevo			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Capitulos/nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-018		Actualizar o Borrar Capitulo	Código		Alfanumérico	Capacitación/Administración de cursos/Capitulos/código de capitulo								
			Título		Alfanumérico	Capacitación/Administración de cursos/Capitulos/código de capitulo								
			Orden		Natural	Capacitación/Administración de cursos/Capitulos/código de capitulo		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
			Capitulo Asociado		Alfanumérico, valida contra listado de capitulos previamente cargados	Capacitación/Administración de cursos/Capitulos/código de capitulo								
			Archivo Adjunto		Alfanumérico	Capacitación/Administración de cursos/Capitulos/código de capitulo								
				Código de capitulo		Capacitación/Administración de cursos/Capitulos/código de capitulo			Pantalla del Capítulo seleccionado		Pantalla del Capítulo seleccionado			
				Lista de valores (para el campo Capitulo Asociado)		Capacitación/Administración de cursos/Capitulos/código de capitulo			Ventana de selección		Ventana de selección			
				Examinar (ruta de Upload de archivos)		Capacitación/Administración de cursos/Capitulos/código de capitulo			Ventana de selección		Ventana de selección			
				Actualizar		Capacitación/Administración de cursos/Capitulos/código de capitulo			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/Capitulos/código de capitulo			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Capitulos/código de capitulo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Capitulos/código de capitulo			Borra la Capitulo		Borra la Capitulo			
				Archivo Adjunto		Capacitación/Administración de cursos/Capitulos/código de capitulo			Ventana de selección		Ventana de selección			
	Examen		Capitulo		Lista desplegable de capitulos cargados previamente	Capacitación/Administración de cursos/Examen								
			Codigo		Alfanumérico	Capacitación/Administración de cursos/Examen	Código completo vacío, parte del código	Código duplicado	Listado con datos que se ajustan al código ingresado	Mensaje de error	Listado con datos que se ajustan al código ingresado	Consulta con registros con código duplicado		error

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv.	Val	Inv.	Val	Inv.	Val.	Inv.
CP-EG-SC-TA-019		Buscar Exámen	Título		Alfanumérico	Capacitación/Administración de cursos/Exámen	Título completo, vacío, parte del título		Listado con datos que se ajustan al Título ingresado		Listado con datos que se ajustan al Título ingresado			
			Autocorregible		Lista desplegable (si /no)	Capacitación/Administración de cursos/Exámen								
				Buscar		Capacitación/Administración de cursos/Exámen			Busca según los parámetros ingresados		Busca segun los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/Exámen			Anula los últimos cambios		Anula los últimos cambios			
				Nuevo		Capacitación/Administración de cursos/Exámen			Pantalla de nuevo Exámen		Pantalla de nuevo Exámen			
CP-EG-SC-TA-020		Consultar Exámen		Código de exámen		Capacitación/Administración de cursos/Exámen/Buscar			Pantalla del exámen seleccionado		Pantalla del exámen seleccionado			
CP-EG-SC-TA-021		Nuevo Exámen	Capítulo		Lista desplegable de capítulos cargados previamente	Capacitación/Administración de cursos/Exámen/Nuevo								
			Código		Alfanumérico	Capacitación/Administración de cursos/Exámen/Nuevo								
			Título		Alfanumérico	Capacitación/Administración de cursos/Exámen/Nuevo								
			Puntaje mínimo de aprobación		Real	Capacitación/Administración de cursos/Exámen/Nuevo		Valores menores o iguales a 0		Control de contenido de dato para el campo		Permitido		error
			Autocorregible		Lista desplegable (si /no)	Capacitación/Administración de cursos/Exámen/Nuevo			Ventana de selección		Ventana de selección			
				Continuar		Capacitación/Administración de cursos/Exámen/Nuevo			Dá de alta el exámen con los datos cargados		Dá de alta el exámen con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Exámen/Nuevo			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Exámen/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-022		Actualizar o Borrar Exámen	Capítulo		Lista desplegable de capítulos cargados previamente	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen		Borrar exámenes rendidos		Mensaje de error		Permitido		error
			Código		Alfanumérico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen								
			Título		Alfanumérico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen								
			Puntaje mínimo de aprobación		Real	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen		Valores menores o iguales a 0		Control de contenido de dato para el campo		Permitido		error
			Autocorregible		Lista desplegable (si /no)	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen								
				Actualizar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Borra el exámen		Borra el exámen			
				Código de exámen		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Pantalla del exámen seleccionado		Pantalla del exámen seleccionado			

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val	Inv
CP-EG-SC-TA-023		Consultar Preguntas		código de descripción de la pregunta		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Pantalla de la pregunta seleccionada		Pantalla de la pregunta seleccionada			
CP-EG-SC-TA-024		Nueva Pregunta	Descripción		Alfanumérico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen								
			Orden		Natural	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen		Valores menores o iguales a 0, repetir varias preguntas con el mismo orden		Control de contenido de dato para el campo		Control de contenido de dato para el campo; permite repetir orden de preguntas		error
			Valor		Real	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen	2.5	Valores menores o iguales a 0	2.5	Control de contenido de dato para el campo	Mensaje de error en numero	Control de contenido de dato para el campo	error	
			Tipo de Pregunta		Lista desplegable de tipo de preguntas cargadas previamente	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen		Seleccionar la opción descriptiva en un exámen autocorregible	Ventana de selección	Mensaje de error o no visualización de la opción	Ventana de selección	Permitido		error
			Cantidad de respuestas correctas		Natural	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen		Valores menores o iguales a 0		Control de contenido de dato para el campo		Control de contenido de dato para el campo		
				Continuar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Dá de alta la pregunta con los datos cargados		Dá de alta la pregunta con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-025		Actualizar o Borrar Preguntas	Descripción		Alfanumerico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta								
			Orden		Natural	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta		Valores menores o iguales a 0, repetir varias preguntas con el mismo orden		Control de contenido de dato para el campo		Control de contenido de dato para el campo; permite repetir orden de preguntas		error
			Valor		Real	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta	2.5	Valores menores o iguales a 0	2.5	Control de contenido de dato para el campo	Mensaje de error en número		error	error
			Tipo de Pregunta		Lista desplegable de tipo de preguntas cargadas previamente	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta								
			Cantidad de respuestas correctas		Natural	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta		Valores menores o iguales a 0, cargar un numero menor al de respuestas correctas cargadas		Mensaje de error		Permite cargar un numero menor al de respuestas correctas cargadas		error
				Actualizar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Anula los últimos cambios		Anula los últimos cambios			

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv.	Val	Inv	Val	Inv.
				Volver		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Borra la pregunta		Borra la pregunta			
CP-EG-SC-TA-026		Consultar Respuestas		código de descripción de la respuesta		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Pantalla de la pregunta seleccionada		Pantalla de la pregunta seleccionada			
				Nuevo		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			pantalla de nueva respuesta		pantalla de nueva respuesta			
			Descripción		Alfanumerico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta		Pargar la misma respuesta con el campo correcta en si y luego en no		Mensaje de error		Permitido		error
			Orden		Natural	Capacitacion/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta		Valores menores o iguales a 0; valores duplicados		Mensaje de error		Respuesta con orden erróneo		error
CP-EG-SC-TA-027		Nueva Respuesta	Correcta		Lista desplegable (si /no)	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta								
				Continuar		Capacitacion/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Dá de alta la respuesta con los datos cargados		Dá de alta la respuesta con los datos cargados			
				Deshacer		Capacitacion/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitacion/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
			Descripción		Alfanumérico	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta/nuevo		modificar una respuesta de forma que sea la misma a otra cuyo campo correcta sea contrario a la misma		Mensaje de error		Permitido		error
			Orden		Natural	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta/nuevo		Valores menores o iguales a 0		Mensaje de error		Mensaje de error		
CP-EG-SC-TA-028		Actualizar o Borrar Respuestas	Correcta		Lista desplegable (si /no)	Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/codigo de descipcion de la pregunta/nuevo		Cantidad menor que las cargadas		Mensaje de error		Respuesta actualizada		error
				Actualizar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/codigo de descipcion de la pregunta/nuevo			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta/nuevo			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta/nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv.	Val.	Inv
				Borrar		Capacitación/Administración de cursos/Exámen/Buscar/código de exámen/código de descipcion de la pregunta/nuevo			Borra la respuesta		Borra la respuesta			
CP-EG-SC-TA-029	Exámen Presencial	Buscar Exámen Presencial		Exámen presencial		Capacitación/Administración de cursos			Pantalla del exámen presencial seleccionado		Pantalla del exámen presencial seleccionado			
CP-EG-SC-TA-030		Consultar Exámen Presencial		Código de fecha		Capacitación/Administración de cursosExámen presencial			Pantalla de la fecha seleccionada		Pantalla de la fecha seleccionada			
				Código de curso		Capacitación/Administración de cursosExámen presencial			Pantalla del curso seleccionado		Pantalla del curso seleccionado			
CP-EG-SC-TA-031		Nuevo Exámen Presencial	Fecha		Fecha	Capacitación/Administración de cursosExámen presencial/código de fecha		fecha <fecha de inicio		Control de contenido de dato para el campo		Exámen presencial creado con fecha errónea		error
			Lugar		Alfanumérico	Capacitación/Administración de cursosExámen presencial/código de fecha		Vacio		Mensaje de error		Permitido		error
			Observación		Alfanumérico	Capacitación/Administración de cursosExámen presencial/código de fecha								
				Calendario para selección de Fecha		Capacitación/Administración de cursosExámen presencial/código de fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Continuar		Capacitación/Administración de cursosExámen presencial/código de fecha			Dá de alta el exámen presencial con los datos cargados		Dá de alta el exámen presencial con los datos cargados			
				Deshacer		Capacitación/Administración de cursosExámen presencial/código de fecha			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursosExámen presencial/código de fecha			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-032		Actualizar o Borrar Exámen Presencial	Fecha		Fecha	Capacitación/Administración de cursosExámen presencial/código de fecha/codigo de fecha		fecha < fecha de inicio. fecha < fecha actual		Control de contenido de dato para el campo		Exámen presencial actualizado con fecha errónea		error
			Lugar		Alfanumérico	Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha								
			Observación		Alfanumérico	Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha								
				Calendario para selección de Fecha		Capacitación/Administración de cursosExámen presencial/código de fecha/codigo de fecha			Ventana de selección de calendario		Ventana de selección de calendario			
				Actualizar		Capacitación/Administración de cursosExámen presencial/código de fecha/codigo de fecha			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha			Anula los ultimos cambios		Anula los ultimos cambios			
				Volver		Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Nuevo		Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha			Link a la pantalla de nuevo Exámen presencial		Link a la pantalla de nuevo Exámen presencial			
				Borrar		Capacitación/Administración de cursosExámen presencial/código de fecha/código de fecha			Borra el exámen presencial		Borra el exámen presencial			
CP-EG-SC-TA-033	Notas	Consultar Nota		Usuario (lista de links de alumnos)		Capacitacion/Administración de cursos			pantalla del usuario seleccionado		pantalla del usuario seleccionado			
			Capitulo		Lista desplegable de capitulos previamente cargados	Capacitacion/Administracion de cursos/Notas								

CASO DE PRUEBA	MODULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv.	Val	Inv
CP-EG-SC-TA-034		Buscar Nota		Buscar		Capacitación/Administración de cursos/Notas			Busca según los parámetros ingresados		Busca según los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/Notas			Anula los últimos cambios		Anula los últimos cambios			
				Código de usuriao		Capacitación/Administración de cursos/Notas			pantalla del usuario seleccionado		pantalla del usuario seleccionado			
				Usuario (lista de links alumnos que rindieron el exámen)		Capacitacion/Administración de cursos/Notas			pantalla del usuario seleccionado		pantalla del usuario seleccionado			
CP-EG-SC-TA-035		Cargar o Actualizar Nota	Calificacion		Real	Capacitación/Administración de cursos/NotasBuscar/código de examen/código de usuario		Valores menores o iguales a 0		Mensaje de error		Permitido		error
			Aprobación		Lista desplegable (si /no)	Capacitación/Administración de cursos/NotasBuscar/código de examen/código de usuario		NO con calificación >= puntaje minimo de aprobación		Mensaje de error		Permitido		error
				Actualizar		Capacitación/Administración de cursos/NotasBuscar/código de examen/código de usuario			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/NotasBuscar/código de examen/código de usuario			Anula los últimos cambios		Anula los últimos cambios			
				Volver		Capacitación/Administración de cursos/NotasBuscar/código de examen/código de usuario			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-036	Aprobación Curso	Buscar Aprobación de Curso		Código de curso		Capacitación/Administración de cursos			Ventana de selección		Ventana de selección			
CP-EG-SC-TA-037		Consultar Aprobación de Curso		Código de alumno		Capacitación/Administración de cursos			Pantalla del alumno seleccionado		Pantalla del alumno seleccionado			
				Nombre (lista de links de alumnos que se han inscripto para rendir el curso)		Capacitacion/Administración de cursos			Ventana de selección		Ventana de selección			
CP-EG-SC-TA-038		Cargar o Actualizar Aprobación de Curso	Aprobación		Lista desplegable (si /no)	Capacitación/Administración de cursos/código de alumno								
			Calificación		Real	Capacitación/Administración de cursos/código de alumno		Valores menores o iguales a 0		Control de contenido de dato para el campo				
				Código de curso		Capacitación/Administración de cursos/código de alumno			Pantalla del curso seleccionado		Pantalla del curso seleccionado			
				Actualizar		Capacitación/Administración de cursos/código de alumno			Actualiza el registro con los cambios realizados		Actualiza el registro con los cambios realizados			
				Deshacer		Capacitación/Administración de cursos/código de alumno			Anula los últimos cambios		Anula los ultimos cambios			
				Volver		Capacitación/Administración de cursos/código de alumno			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			

DATOS DE CREACIÓN		
APELLIDO Y NOMBRE: **Fabricio Borda**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **fborda@sanluis.gov.ar**	FECHA: **01 / 11 / 02**

DATOS DE REVISIÓN		
APELLIDO Y NOMBRE: **Cristian Mendoza**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **cmendoza@sanluis.gov.ar**	FECHA: **11 / 11 / 02**

DATOS DE APROBACIÓN		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **14 / 11 / 02**

DATOS DEL TESTER		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **15 / 11 / 02**

DATOS DE LA PRUEBA

TIPO DE PRUEBA: ☒ **FUNCIONAL** ☐ NO FUNCIONAL ☐ OTRA: _____
NIVEL DE TESTING: ☐ UNIDAD ☐ INTEGRACIÓN ☐ SISTEMA ☒ **ALFA** ☐ BETA ☐ ACEPTACIÓN
SISTEMA: **CAPACITACIÓN**
SUBSISTEMA: **PARÁMETROS**
MÓDULO: **TEMAS – AUTORES – TUTORES – UNIDADES DE TIEMPO**

OBSERVACIONES

CASO DE PRUEBA	MÓDULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val.	Inv.
CP-EG-SC-TA-039	Temas	Buscar Tema	Tema		Alfanumérico	Capacitacion/Administración de cursos/Temas	tema completo, vacío; %parte del tema%		Listado con datos que se ajustan al Código ingresado		Listado con datos que se ajustan al Código ingresado			
				Buscar		Capacitación/Administración de cursos/Temas			Busca según los parámetros ingresados		Busca según los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/Temas			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Temas			pantalla de nuevo Tema		pantalla de nuevo Tema			
				Ultimo		Capacitación/Administración de cursos/Temas/Buscar			Avanza al último registro		Avanza al último registro			
				Siguiente		Capacitación/Administración de cursos/Temas/Buscar			Avanza al siguiente registro		Avanza al siguiente registro			
				Refrescar		Capacitación/Administración de cursos/Temas/Buscar			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Temas/Buscar			Pantalla de nuevo Tema		Pantalla de nuevo Tema			
				Volver		Capacitación/Administración de cursos/Temas/Buscar			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-040		Consultar Tema		Código de Tema		Capacitación/Administración de cursos/Temas/Buscar			registro con los datos del tema selecionado		registro con los datos del tema selecionado			
CP-EG-SC-TA-041		Nuevo Tema	Tema		Alfanumérico	Capacitación/Administración de cursos/Temas		cargar el mismo tema varias veces		Control de contenido de dato para el campo		Permitido		error
				Continuar		Capacitación/Administración de cursos/Temas			Dá de alta el tema con los datos cargados		Dá de alta el tema con los datos cargados			
				Deshacer		Capacitación/Administración de cursos/Temas			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/Temas			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-042		Actualizar o Borrar Tema	Tema		Alfanumérico	Capacitación/Administración de cursos/Temas/Nuevo								
			Fecha de Activación		Fecha	Capacitación/Administración de cursos/Temas/Buscar/Código de Tema		fecha de activación > fecha de desactivación		Control de contenido de dato para el campo		Permitido		error
			Fecha de Desactivación		Fecha	Capacitación/Administración de cursos/Temas/Buscar/Código de Tema		fecha de desactivación < fecha de activación		Control de contenido de dato para el campo		Permitido		error
				Calendario para selección de Fecha de Activación		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			ventana de selección de calendario		icono omitido			error
				Calendario para selección de Fecha de Desactivación		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			ventana de selección de calendario		icono omitido			error
				Actualizar		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			incorpora las modificaciones al sistema		incorpora las modificaciones al sistema			
				Deshacer		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			reestablece datos antes de ser modificados		reestablece datos antes de ser modificados			
				Volver		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Temas/Buscar/Código de Tema			coloca la fecha actual en el campo fecha de desactivación		coloca la fecha actual en el campo fecha de desactivación			
			Legajo		Entero	Capacitación/Administración de cursos/Autores								
			Tipo Documento		Lista desplegable de tipos de documentos cargados previamente	Capacitación/Administración de cursos/Autores								
			Numero		Entero	Capacitación/Administración de cursos/Autores								
			Apellido		Alfabetico	Capacitación/Administración de cursos/Autores								
			Nombre		Alfabetico	Capacitación/Administración de cursos/Autores								
				Lista de Valores (para el campo Legajo)		Capacitación/Administración de cursos/Autores			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			

CASO DE PRUEBA	MÓDULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val.	Inv.
CP-EG-SC-TA-843	Autores	Buscar		Lista de Valores (para el campo Numero)		Capacitación/Administración de cursos/Autores			ventana de búsqueda de autor por nro. doc. / listado de autores		ventana de búsqueda de autor por nro doc. / listado de autores			
				Lista de Valores (para el campo Apellido)		Capacitación/Administración de cursos/Autores			ventana de búsqueda de autor por apellido / listado de autores		ventana de búsqueda de autor por apellido / listado de autores			
				Buscar		Capacitación/Administración de cursos/Autores			Busca según los parámetros ingresados		Busca según los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/Autores			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Autores			pantalla de nuevo Autor		pantalla de nuevo Autor			
				Ultimo		Capacitación/Administración de cursos/Autores/Buscar			Avanza al último registro		Avanza al último registro			
				Siguiente		Capacitación/Administración de cursos/Autores/Buscar			Avanza al siguiente registro		Avanza al siguiente registro			
				Refrescar		Capacitación/Administración de cursos/Autores/Buscar			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Volver		Capacitación/Administración de cursos/Autores/Buscar			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Nuevo		Capacitación/Administración de cursos/Autores/Buscar			pantalla de nuevo Autor		pantalla de nuevo Autor			
CP-EG-SC-TA-844		Consultar Autor		Código de Legajo		Capacitación/Administración de cursos/Autores/Buscar			datos del autor		datos del autor			
CP-EG-SC-TA-845		Consultar Curriculum del Autor		curriculum (link al curriculum del Autor consultado)		Capacitación/Administración de cursos/Autores/Buscar			muestra el archivo del curriculum		muestra el archivo del curriculum			
CP-EG-SC-TA-846		Nuevo Autor	Legajo		Entero	Capacitación/Administración de cursos/Autores/Nuevo		cargar el mismo autor varias veces		mensaje de error		Permitido		error
			Tipo Documento		Lista desplegable de tipos de documentos cargados previamente	Capacitación/Administración de cursos/Autores/Nuevo		cargar el mismo autor varias veces		mensaje de error		Permitido		error
			Numero		Entero	Capacitación/Administración de cursos/Autores/Nuevo		cargar el mismo autor varias veces		mensaje de error		Permitido		error
			Apellido		Alfabético	Capacitación/Administración de cursos/Autores/Nuevo		cargar el mismo autor varias veces		mensaje de error		Permitido		error
			Nombre		Alfabético	Capacitación/Administración de cursos/Autores/Nuevo		cargar el mismo autor varias veces		mensaje de error		Permitido		error
			Curriculum		Alfanumérico	Capacitación/Administración de cursos/Autores/Nuevo	cargar archivos de distintos directorios con nombres largos		permitido		no se pudo cargara archivos con pasos largos y / o nombres extensos		error	
				Lista de Valores (para el campo Legajo)		Capacitación/Administración de cursos/Autores/Nuevo			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Numero)		Capacitación/Administración de cursos/Autores/Nuevo			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Apellido)		Capacitación/Administración de cursos/Autores/Nuevo			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			
				Examinar		Capacitación/Administración de cursos/Autores/Nuevo			pantalla para selección de archivo		pantalla para selección de archivo			
				Continuar		Capacitación/Administración de cursos/Autores/Nuevo			Dá de alta el curriculum a un autor		Dá de alta el curriculum a un autor			
				Deshacer		Capacitación/Administración de cursos/Autores/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			

CASO DE PRUEBA	MÓDULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val	Inv
				Volver		Capacitación/Administración de cursos/Autores/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-047		Actualizar o Borrar Autores (sólo curriculum)	Curriculum		Alfanumérico	Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo	cambiar archivos de distintos directorios con nombres largos		permitido		no se pudo cargara archivos con pasos largos y / o nombres extensos		error	
				Actualizar		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			incorpora las modificaciones al sistema		incorpora las modificaciones al sistema			
				Deshacer		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			reestablece datos antes de ser modificados		reestablece datos antes de ser modificados			
				Volver		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			elimina el archivo de curriculum del autor		elimina el archivo de curriculum del autor			
				Examinar		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			pantalla para selección de archivo		pantalla para selección de archivo			
				Curriculum (link al curriculum del Autor consultado)		Capacitación/Administración de cursos/Autores/Buscar/Código de Legajo			muestra el archivo del curriculum		muestra el archivo del curriculum			
CP-EG-SC-TA-048		Buscar Tutor	Legajo		Entero	Capacitación/Administración de cursos/Tutores								
			Tipo Documento		Lista desplegable de tipos de documentos cargados previamente	Capacitación/Administración de cursos/Tutores								
			Número		Entero	Capacitación/Administración de cursos/Tutores								
			Apellido		Alfabético	Capacitación/Administración de cursos/Tutores								
			Nombre		Alfabético	Capacitación/Administración de cursos/Tutores								
				Buscar		Capacitación/Administración de cursos/Tutores			Busca según los parámetros ingresados		Busca según los parámetros ingresados			
				Deshacer		Capacitación/Administración de cursos/Tutores			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Nuevo		Capacitación/Administración de cursos/Tutores			Pantalla de nuevo Tutor		Pantalla de nuevo Tutor			
				Lista de Valores (para el campo Legajo)		Capacitación/Administración de cursos/Tutores			Ventana de búsqueda de autor por legajo / listado de autores		Ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Número)		Capacitación/Administración de cursos/Tutores			Ventana de búsqueda de autor por legajo / listado de autores		Ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Apellido)		Capacitación/Administración de cursos/Tutores			Ventana de búsqueda de autor por legajo / listado de autores		Ventana de búsqueda de autor por legajo / listado de autores			
				Último		Capacitación/Administración de cursos/Tutores/Buscar			Avanza al último registro		Avanza al último registro			
				Siguiente		Capacitación/Administración de cursos/Tutores/Buscar			Avanza al siguiente registro		Avanza al siguiente registro			
				Refrescar		Capacitación/Administración de cursos/Tutores/Buscar			Actualiza los registros de la pantalla		Actualiza los registros de la pantalla			
				Volver		Capacitación/Administración de cursos/Tutores/Buscar			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Nuevo		Capacitación/Administración de cursos/Tutores/Buscar			Pantalla de nuevo Tutor		Pantalla de nuevo Tutor			
CP-EG-SC-TA-049		Consultar Tutor		Legajo (lista de links a tutores a consultar)		Capacitación/Administración de cursos/Tutores/Buscar			Datos del Tutor		Datos del Tutor			
CP-EG-SC-TA-050		Consultar Curriculum del Tutor		curriculum (link al curriculum del tutor consultado)		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Muestra el archivo del curriculum		Muestra el archivo del curriculum			
			Legajo		Entero	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar el mismo tutor varias veces		Mensaje de error		Permitido		error	
			Tipo Documento		Lista desplegable de tipos de documentos cargados previamente	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar el mismo tutor varias veces		Mensaje de error		Permitido		error	

CASO DE PRUEBA	MÓDULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val.	Inv.	Val.	Inv.	Val.	Inv.	Val.	Inv.
CP-EG-SC-TA-051	Tutores	Nuevo Tutor	Numero		Entero	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar el mismo tutor varias veces		Mensaje de error		Permitido		error	
			Apellido		Alfabético	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar el mismo tutor varias veces		Mensaje de error		Permitido		error	
			Nombre		Alfabético	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar el mismo tutor varias veces		Mensaje de error		Permitido		error	
			Curriculum		Alfanumérico	Capacitación/Administración de cursos/Tutores/Nuevo	Cargar archivos de distintos directorios con nombres largos		Permitido		No se pudo cargara archivos con pasos largos y / o nombres extensos		error	
				Lista de Valores (para el campo Legajo)		Capacitación/Administración de cursos/Tutores/Nuevo			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Número)		Capacitación/Administración de cursos/Tutores/Nuevo			ventana de búsqueda de autor por legajo / listado de autores		ventana de búsqueda de autor por legajo / listado de autores			
				Lista de Valores (para el campo Apellido)		Capacitación/Administración de cursos/Tutores/Nuevo			Ventana de búsqueda de autor por legajo / listado de autores		Ventana de búsqueda de autor por legajo / listado de autores			
				Continuar		Capacitación/Administración de cursos/Tutores/Nuevo			Dá de alta el curriculum a un autor		Dá de alta el curriculum a un autor			
				Deshacer		Capacitación/Administración de cursos/Tutores/Nuevo			Pantalla para selección de archivo		Pantalla para selección de archivo			
				Volver		Capacitación/Administración de cursos/Tutores/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Examinar		Capacitación/Administración de cursos/Tutores/Nuevo			Limpia todos los campos de la pantalla		No se encuentra el botón		error	
CP-EG-SC-TA-052		Actualizar o Borrar Tutores	Curriculum		Alfanumérico	Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo	Cambiar archivos de distintos directorios con nombres largos		Permitido		No se pudo cargara archivos con pasos largos y / o nombres extensos		error	
				Actualizar		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Incorpora las modificaciones al sistema		Incorpora las modificaciones al sistema			
				Deshacer		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
				Borrar		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Elimina el archivo de curriculum del autor		Elimina el archivo de curriculum del autor			
				Volver					Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Examinar		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Pantalla para selección de archivo		Pantalla para selección de archivo			
				Curriculum (link al curriculum del tutor consultado)		Capacitación/Administración de cursos/Tutores/Buscar/Código de Legajo			Muestra el archivo del curriculum		Muestra el archivo del curriculum			
CP-EG-SC-TA-053		Consultar Unidad de Tiempo		Nuevo		Capacitación/Administración de cursos/Unidades de Tiempo			Pantalla de nueva Unidad de Tiempo		Pantalla de nueva Unidad de Tiempo			
				Volver		Capacitación/Administración de cursos/Unidades de Tiempo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-054				Código de Tipo de Unidad de Tiempo (lista de links a unidades de tiempo)		Capacitación/Administración de cursos/Unidades de Tiempo			Muestra los datos de la unidad de tiempo seleccionada		Muestra los datos de la unidad de tiempo seleccionada			
			Descripción		Alfanumérico	Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo		Cargar la misma unidad de tiempo varias veces		Control de contenido de dato para el campo		Permitido		error

CASO DE PRUEBA	MÓDULO	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
							Val	Inv	Val	Inv	Val	Inv	Val	Inv.
CP-EG-SC-TA-055	Unidades de Tiempo	Nueva Unidad de Tiempo	órden		Natural	Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo		Repetir órden; 0; 5; 2;5		Control de contenido de dato para el campo		Permite el repetir el órden		error
			Clave Externa		Alfanumérico	Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo								
				Continuar		Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo			Dá de alta el curriculum a un autor		Dá de alta el curriculum a un autor			
				Deshacer		Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo			Limpia todos los campos de la pantalla		Limpia todos los campos de la pantalla			
				Volver		Capacitación/Administración de cursos/Unidades de Tiempo/Nuevo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
CP-EG-SC-TA-056		Actualizar o Borrar Unidad de Tiempo	Descripción		Alfanumérico	Capacitación/Administración de cursos/Unidades de Tiempo/Código de Unidad de Tiempo								
			órden		Natural	Capacitación/Administración de cursos/Unidades de Tiempo/Código de Unidad de Tiempo		Repetir órden; 0; 5; 2;5		Control de contenido de dato para el campo		Permite el repetir el órden		error
				Actualizar		Capacitación/Administración de cursos/Unidades de Tiempo/Código de Unidad de Tiempo			Incorpora las modificaciones al sistema		Incorpora las modificaciones al sistema			
				Deshacer					Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
				Volver		Capacitación/Administración de cursos/Unidades de Tiempo/Código de Unidad de Tiempo			Vuelve a la pantalla anterior		Vuelve a la pantalla anterior			
				Borrar		Capacitación/Administración de cursos/Unidades de Tiempo/Código de Unidad de Tiempo			Coloca la fecha actual en el campo fecha de desactivación		Coloca la fecha actual en el campo fecha de desactivación			

DATOS DE CREACIÓN		
APELLIDO Y NOMBRE: **Fabricio Borda**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **fborda@sanluis.gov.ar**	FECHA: **01 / 11 / 02**

DATOS DE REVISIÓN		
APELLIDO Y NOMBRE: **Cristian Mendoza**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **cmendoza@sanluis.gov.ar**	FECHA: **11 / 11 / 02**

DATOS DE APROBACIÓN		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **14 / 11 / 02**

DATOS DEL TESTER		
APELLIDO Y NOMBRE: **Matias Mauro Fredes**		TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**	FECHA: **15 / 11 / 02**

DATOS DE LA PRUEBA

TIPO DE PRUEBA: ☒ **FUNCIONAL** ☐ NO FUNCIONAL ☐ OTRA: _____
NIVEL DE TESTING: ☐ UNIDAD ☐ INTEGRACIÓN ☐ SISTEMA ☒ **ALFA** ☐ BETA ☐ ACEPTACIÓN
SISTEMA: **CAPACITACIÓN**
SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:

OBSERVACIONES

CASO DE PRUEBA	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
						Val	Inv.	Val	Inv	Val	Inv	Val	Inv
CP-EG-SC-TA-058	Cursos por Alumnos	Fecha Desde		Fecha	Capacitación/Administración de cursos/Cursos por alumnos	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		permitido		error
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Cursos por alumnos	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		permitido		error
			Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Cursos por alumnos			Ventana de selección de calendario		Icono omitido		error	
			Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/Cursos por alumnos			Ventana de selección de calendario		Icono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Cursos por alumnos			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Cursos por alumnos			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			
CP-EG-SC-TA-059	Cursos	Área		Alfanumérico	Capacitación/Administración de cursos/Cursos	NA	NA	NA	NA				
		Curso		Alfanumérico	Capacitación/Administración de cursos/Cursos	NA	NA	NA	NA				
			Lista de valores (para el campo Área)		Capacitación/Administración de cursos/Cursos			Ventana de búsqueda de área / listado de áreas		Ventana de búsqueda de área / listado de áreas			
			Lista de valores (para el campo Curso)		Capacitación/Administración de cursos/Cursos			Ventana de búsqueda de curso / listado de cursos		Ventana de búsqueda de curso / listado de cursos			
			Deshacer		Capacitación/Administración de cursos/Cursos			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Cursos			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			
CP-EG-SC-TA-060	Cursos que realiza un Alumno	Fecha Desde		Fecha	Capacitación/Administración de cursos/Cursos que realiza un alumno	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		Permitido		error
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Cursos que realiza un alumno	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		Permitido		error
		Nro. Doc		Entero	Capacitación/Administración de cursos/Cursos que realiza un alumno	Número completo; %parte del número conocida%	NA	Único registro, listado de registros	NA				
		Apellido y nombre		Alfanumérico	Capacitación/Administración de cursos/Cursos que realiza un alumno	Apellido y nombre completo; %parte del apellido y /o nombre%.	NA	Único registro, listado de registros	NA				
			Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Cursos que realiza un alumno			Ventana de selección de calendario		Icono omitido		error	
			Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/Cursos que realiza un alumno			Ventana de selección de calendario		Icono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Cursos que realiza un alumno			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Cursos que realiza un alumno			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			

CASO DE PRUEBA	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
						Val	Inv	Val.	Inv	Val	Inv	Val	Inv
CP-EG-SC-TA-061	Consulta Completa de Cursos	curso		Alfanumérico	Capacitación/Administración de cursos/Consulta completa de cursos	NA	NA	NA	NA				
			Lista de valores (para el campo Curso)		Capacitación/Administración de cursos/Consulta completa de cursos			Ventana de búsqueda de curso / listado de cursos		Ventana de búsqueda de curso / listado de cursos			
			Deshacer		Capacitación/Administración de cursos/Consulta completa de cursos			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Consulta completa de cursos			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			
CP-EG-SC-TA-062	Listado de Usuarios	Fecha Desde		Fecha	Capacitación/Administración de cursos/Listado de usuarios	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		Permitido		error
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Listado de usuarios	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		Permitido		error
		Área		Alfanumérico	Capacitación/Administración de cursos/Listado de usuarios	NA	NA	NA	NA				
		Curso		Alfanumérico	Capacitación/Administración de cursos/Listado de usuarios	NA	NA	NA	NA				
			Lista de valores (para el campo Área)		Capacitación/Administración de cursos/Listado de usuarios			Ventana de búsqueda de área / listado de áreas		Ventana de búsqueda de área / listado de áreas			
			Lista de valores (para el campo Curso)		Capacitación/Administración de cursos/Listado de usuarios			Ventana de búsqueda de curso / listado de cursos		Ventana de búsqueda de curso / listado de cursos			
			Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Listado de usuarios			Ventana de selección de calendario		Ícono omitido		error	
			Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/Listado de usuarios			Ventana de selección de calendario		Ícono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Listado de usuarios			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Listado de usuarios			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			
CP-EG-SC-TA-063	Cursos por Tutor	Fecha Desde		Fecha	Capacitación/Administración de cursos/Cursos por tutor	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		Permitido		
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Cursos por tutor	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		Permitido		
			Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Cursos por tutor			Ventana de selección de calendario		Ícono omitido		error	
			Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/Cursos por tutor			Ventana de selección de calendario		Ícono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Cursos por tutor			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Cursos por tutor			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			

CASO DE PRUEBA	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
						Val	Inv	Val	Inv	Val	Inv	Val.	Inv
		Fecha Desde		Fecha	Capacitación/Administración de cursos/Alumnos por curso	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		Permitido		error
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Alumnos por curso	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		Permitido		error
		Area		Alfanumérico	Capacitación/Administración de cursos/Alumnos por curso	NA	NA	NA	NA				
		Curso		Alfanumérico	Capacitación/Administración de cursos/Alumnos por curso	NA	NA	NA	NA	se observa que no se ha decrementado la cantidad de vacantes cuando se han inscripto alumnos en un curso		error	
CP-EG-SC-TA-064	Alumnos por Curso		Lista de valores (para el campo Área)		Capacitación/Administración de cursos/Alumnos por curso			Ventana de búsqueda de área / listado de áreas		Ventana de búsqueda de área / listado de áreas			
			Lista de valores (para el campo Curso)		Capacitación/Administración de cursos/Alumnos por curso			Ventana de búsqueda de curso / listado de cursos		Ventana de búsqueda de curso / listado de cursos			
			Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Alumnos por curso			Ventana de selección de calendario		Ícono omitido		error	
			Calendario para selección de Fecha Hasta		Capacitación/Administración de cursos/Alumnos por curso			Ventana de selección de calendario		Ícono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Alumnos por curso			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Alumnos por curso			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Registros incorrectos en consulta		error	
		Curso		Alfanumérico	Capacitación/Administración de cursos/Exámenes	NA	NA	NA	NA				
			Lista de valores (para el campo Curso)		Capacitación/Administración de cursos/Exámenes			Ventana de búsqueda de curso / listado de cursos		Ventana de búsqueda de curso / listado de cursos			
CP-EG-SC-TA-065	Examenes		Deshacer		Capacitación/Administración de cursos/Exámenes			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administración de cursos/Exámenes			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			
		Fecha Desde		Fecha	Capacitación/Administración de cursos/Estadisticas	NA	fecha de inicio > fecha de finalización	NA	Control de contenido de dato para el campo		Permitido		error
		Fecha Hasta		Fecha	Capacitación/Administración de cursos/Estadisticas	NA	fecha de finalización < fecha de inicio	NA	Control de contenido de dato para el campo		Permitido		error
		Área		Alfanumérico	Capacitación/Administración de cursos/Estadisticas	NA	NA	NA	NA				
			Lista de valores (para el campo Área)		Capacitación/Administración de cursos/Estadisticas			Ventana de busqueda de área / listado de áreas		Ventana de busqueda de área / listado de áreas			
CP-EG-SC-TA-066	Estadisticas		Calendario para selección de Fecha Desde		Capacitación/Administración de cursos/Estadisticas			Ventana de selección de calendario		Ícono omitido		error	

CASO DE PRUEBA	FUNCIÓN	CAMPO	BOTÓN / ICONO / LINK	PROPIEDADES	SECUENCIA DE PASOS	DATOS DE ENTRADA		RESULT. ESPERADO		DATOS DE SALIDA		ERROR	
						Val	Inv	Val.	Inv.	Val	Inv	Val.	Inv
			Calendario para seleccion de Fecha Hasta		Capacitacion/Administración de cursos/Estadisticas			Ventana de selección de calendario		Ícono omitido		error	
			Deshacer		Capacitación/Administración de cursos/Estadisticas			Reestablece datos antes de ser modificados		Reestablece datos antes de ser modificados			
			Consultar		Capacitación/Administracion de cursos/Estadisticas			Resultado de la consulta (lista de acuerdo a los parámetros ingresados)		Resultado de la consulta (lista de acuerdo a los parámetros ingresados)			

NA = No Aplica

ANEXO 5: REGISTRO DE ERRORES – SISTEMA DE CAPACITACIÓN

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

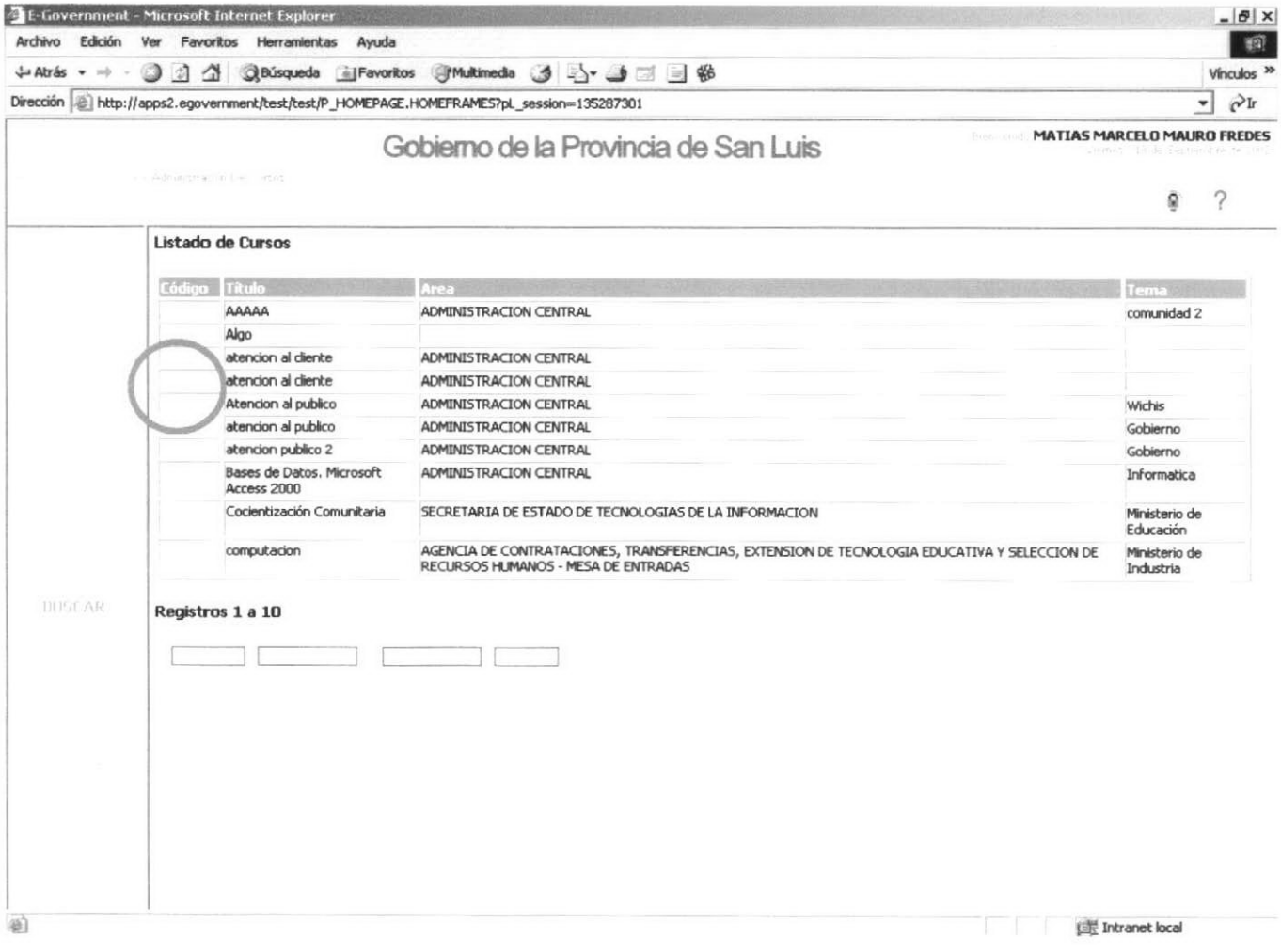
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
DUPLICACIÓN DE REGISTROS - EL CÓDIGO DE UN CURSO DEBE SER ÚNICO.
IMAGEN



Código	Título	Area	Tema
	AAAAA	ADMINISTRACION CENTRAL	comunidad 2
	Algo		
	atencion al cliente	ADMINISTRACION CENTRAL	
	atencion al cliente	ADMINISTRACION CENTRAL	
	Atencion al publico	ADMINISTRACION CENTRAL	Wichis
	atencion al publico	ADMINISTRACION CENTRAL	Gobierno
	atencion publico 2	ADMINISTRACION CENTRAL	Gobierno
	Bases de Datos. Microsoft Access 2000	ADMINISTRACION CENTRAL	Informatica
	Cocientización Comunitaria	SECRETARIA DE ESTADO DE TECNOLOGIAS DE LA INFORMACION	Ministerio de Educación
	computacion	AGENCIA DE CONTRATACIONES, TRANSFERENCIAS, EXTENSION DE TECNOLOGIA EDUCATIVA Y SELECCION DE RECURSOS HUMANOS - MESA DE ENTRADAS	Ministerio de Industria

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Ingresar Autor**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-003

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

ADMITE MÚLTIPLES CARGAS DE UN MISMO AUTOR.
EL CAMPO DURACIÓN NO CONTROLA O NO VERIFICA LA DIFERENCIA ENTRE FECHA HASTA Y FECHA DESDE.

IMAGEN

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

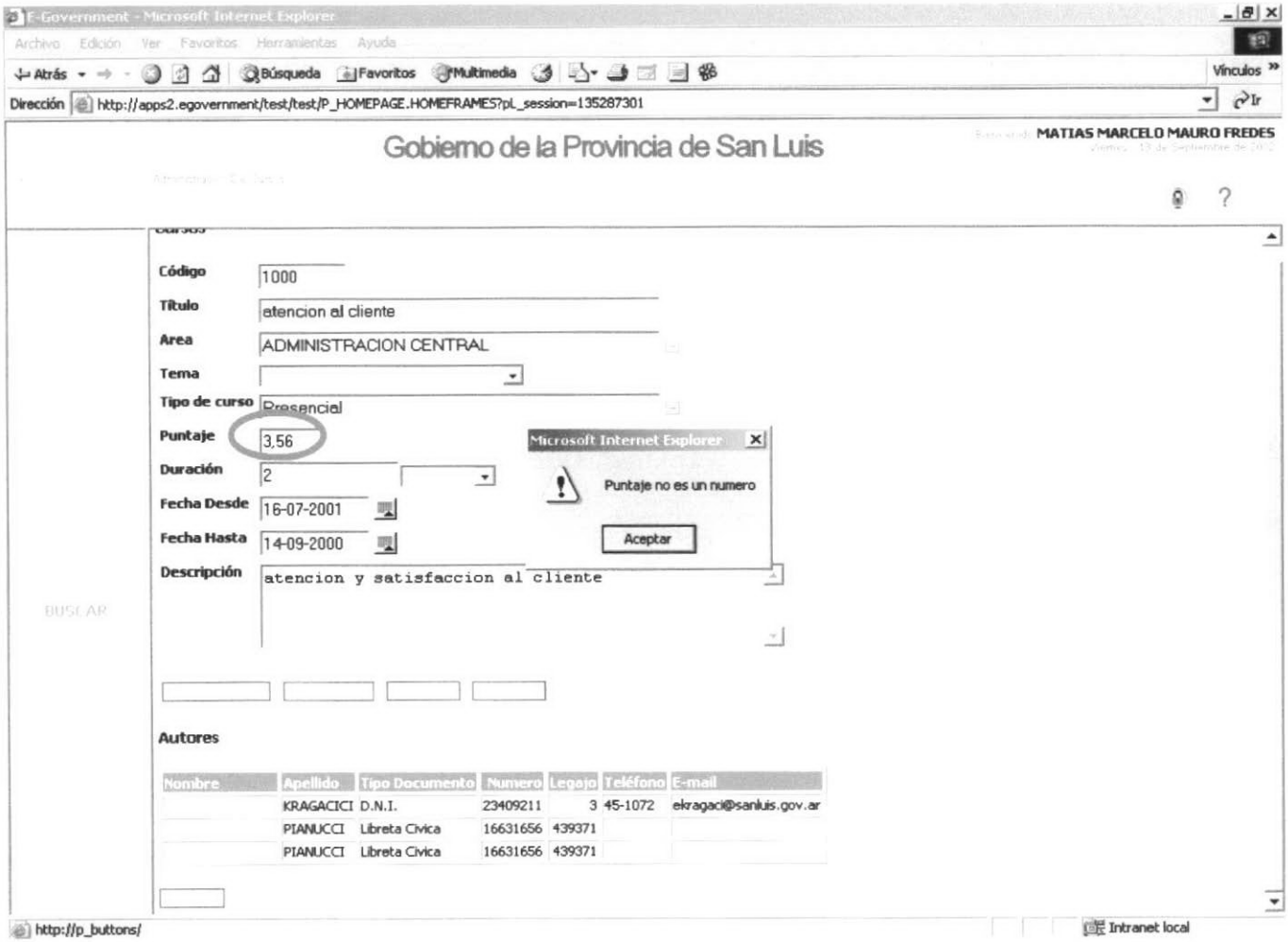
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EL CAMPO PUNTAJE NO CUMPLE CON PERMITIR NÚMEROS REALES POSITIVOS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Consultar Hora**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

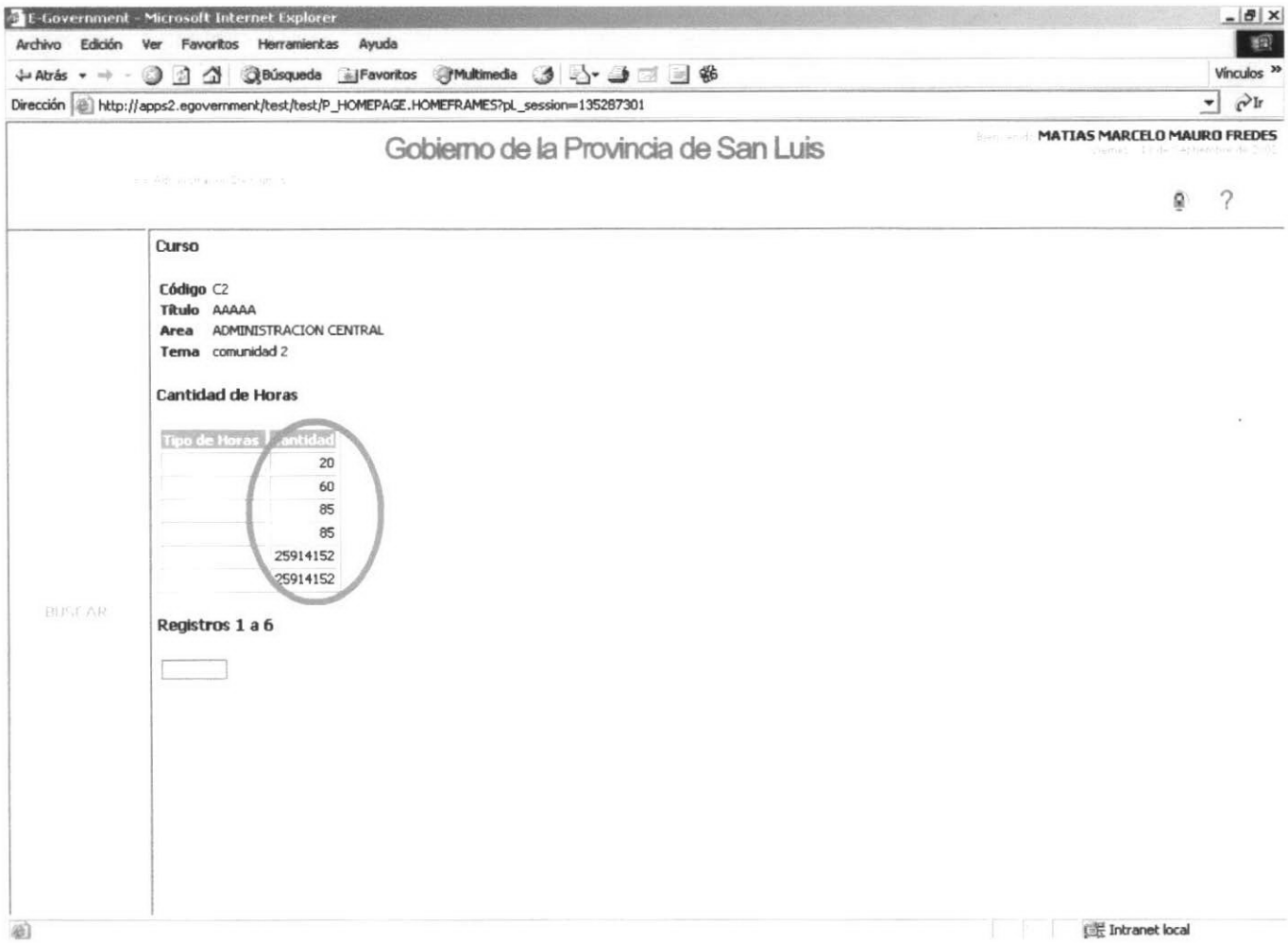
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-006

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

PARA UN CURSO ADMITE MÚLTIPLES REGISTROS DE UN TIPO DE HORA.
LA SUMA DE CANTIDAD DE HORAS NO ES CONTROLADA RESPECTO DEL TIEMPO ASIGNADO PARA EL CURSO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

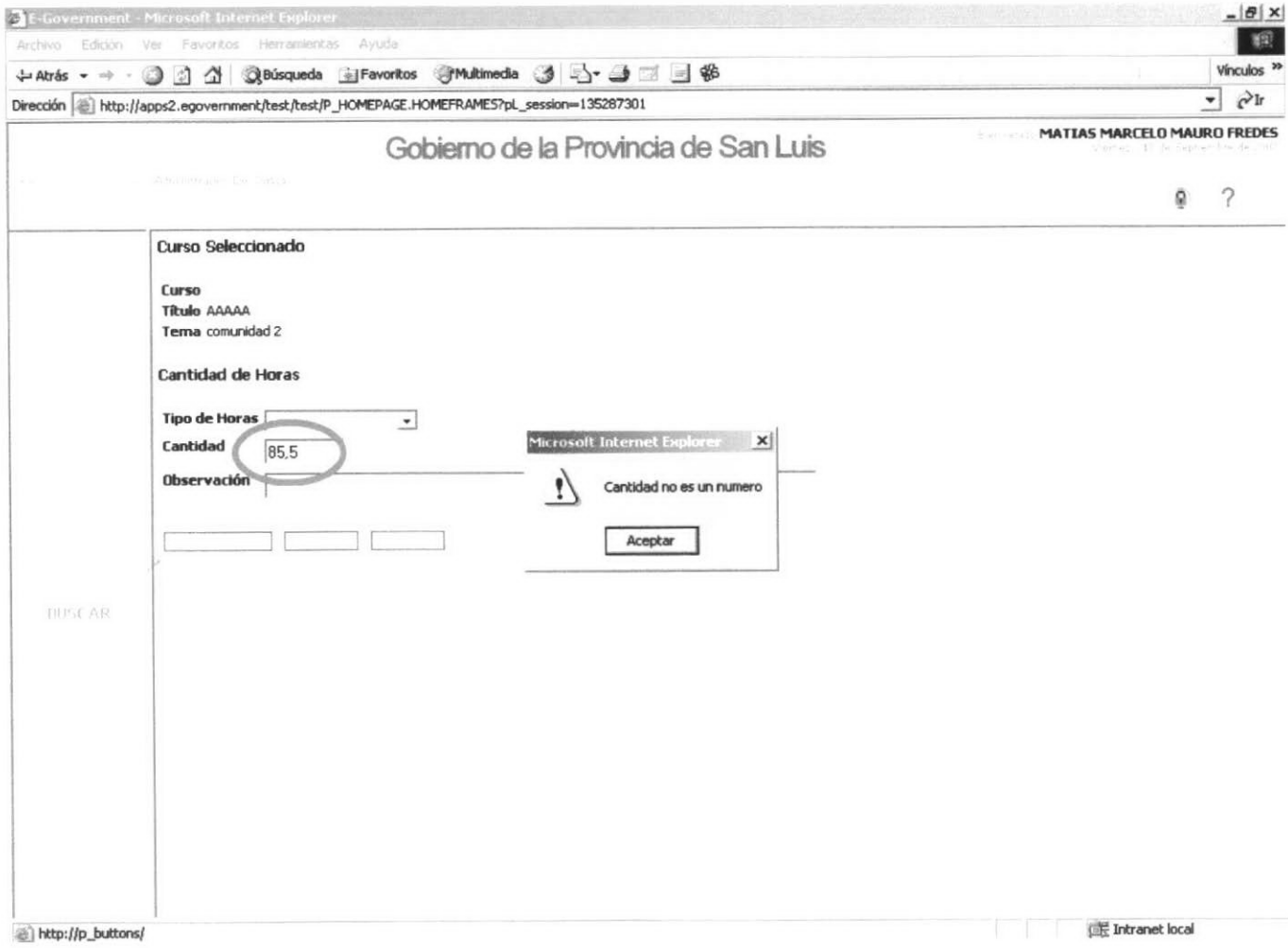
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Hora**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-007

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EL CAMPO CANTIDAD NO CUMPLE CON PERMITIR NÚMEROS REALES POSITIVOS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

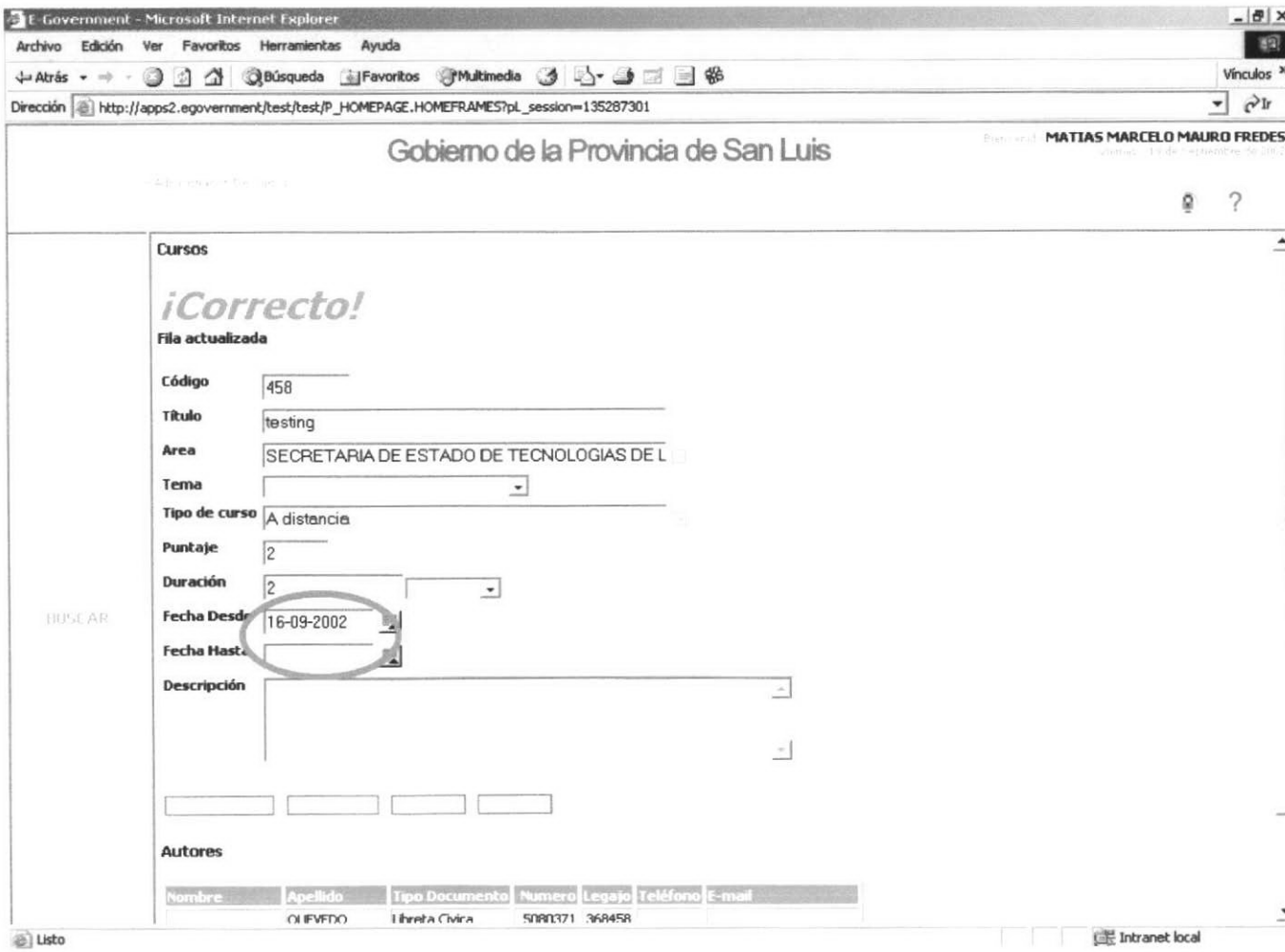
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso- Actualizar Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

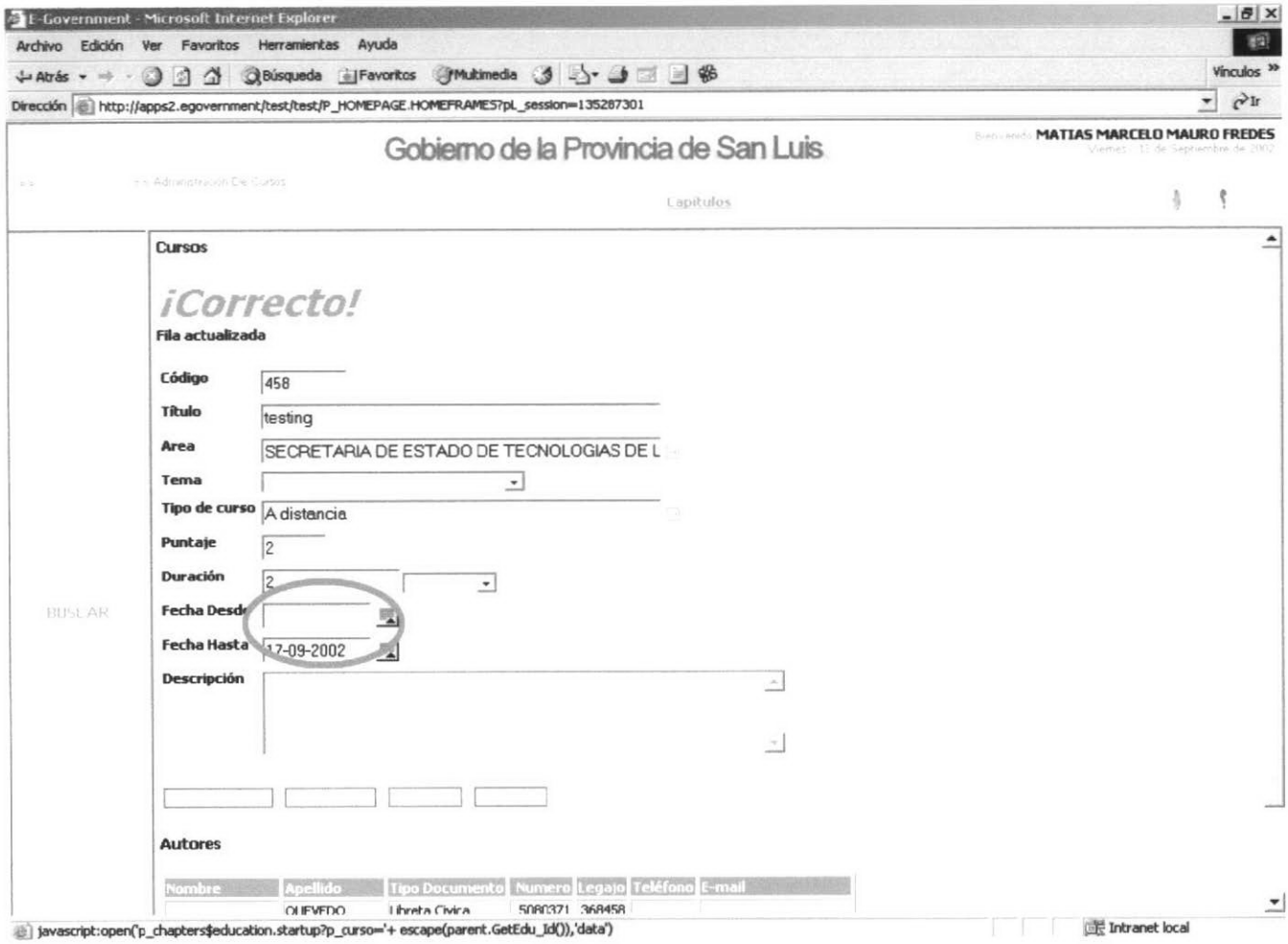
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002 **CP-EG-SC-TA-005**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
ADMITE EN LA CARGA Y ACTUALIZACIÓN TENER FECHA DESDE SIN TENER FECHA HASTA Y VICEVERSA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

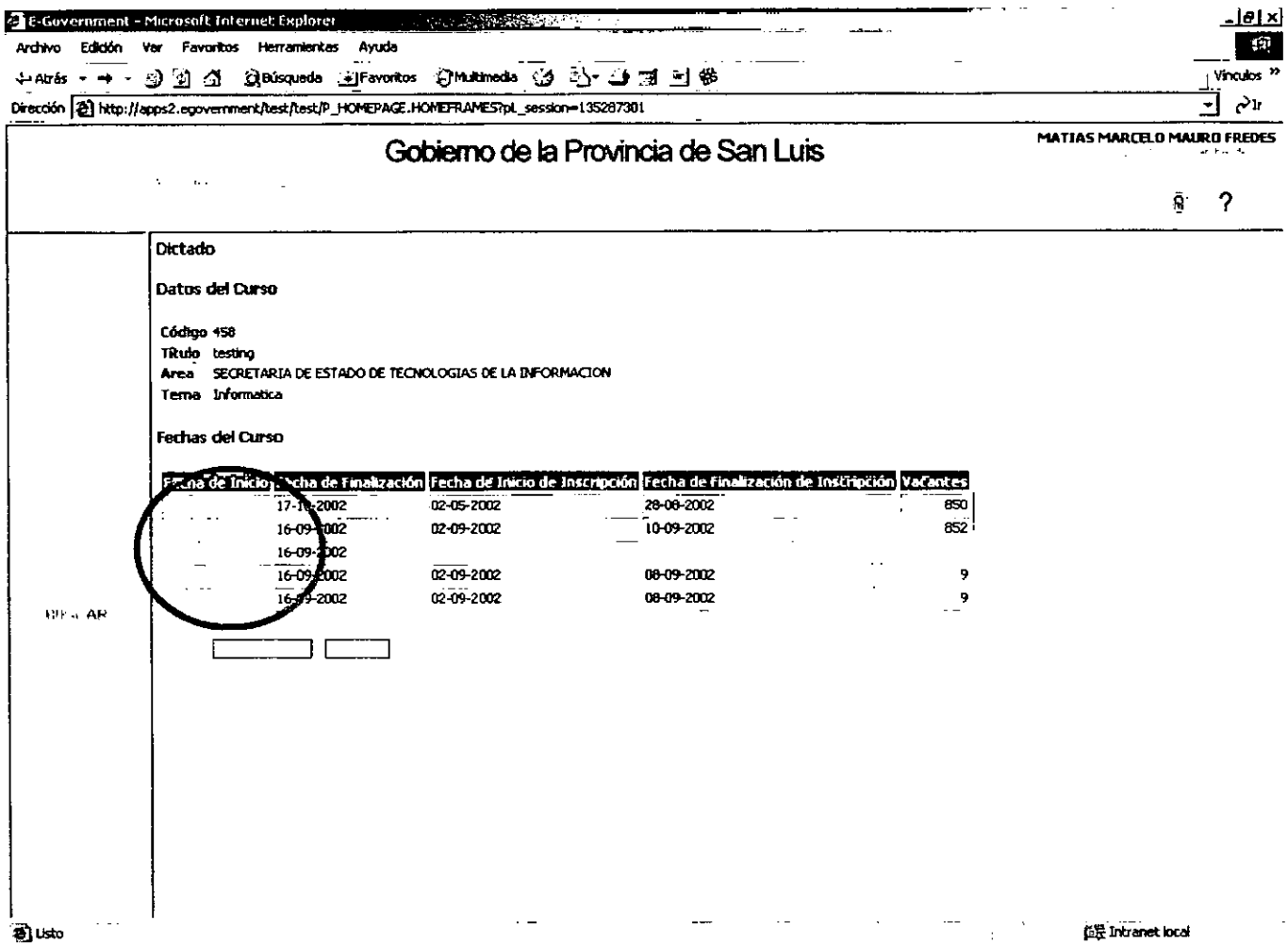
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Dictado**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Consultar Fechas de Dictado de Cursos**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-009

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
ADMITE MÚLTIPLES REGISTROS IGUALES DE FECHAS DEL CURSO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

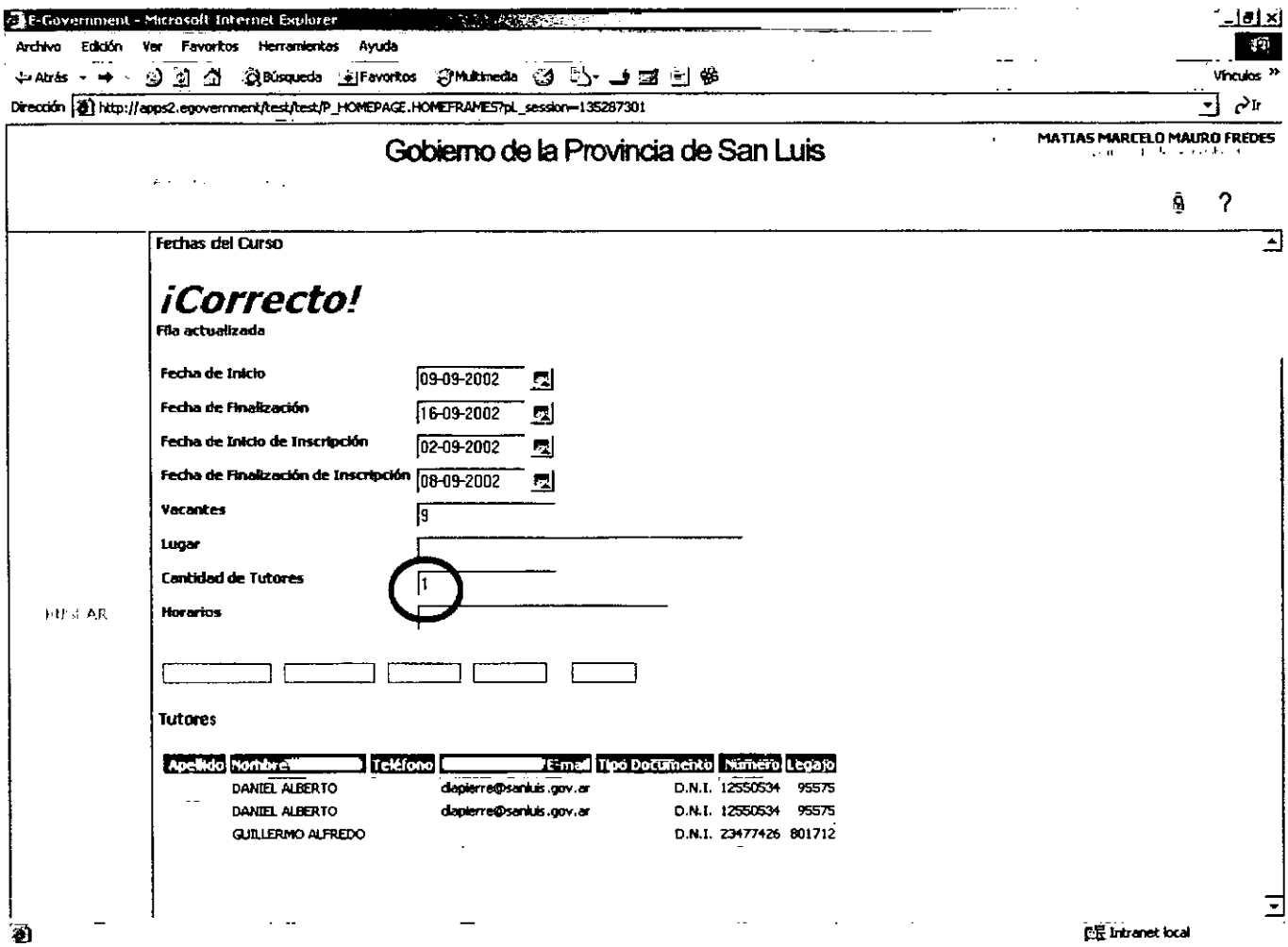
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Dictado**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar o Borrar Fechas de Dictado de Cursos**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-011

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
ADMITE ACTUALIZAR LA CANTIDAD DE TUTORES A UN NÚMERO MENOR AL CARGADO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

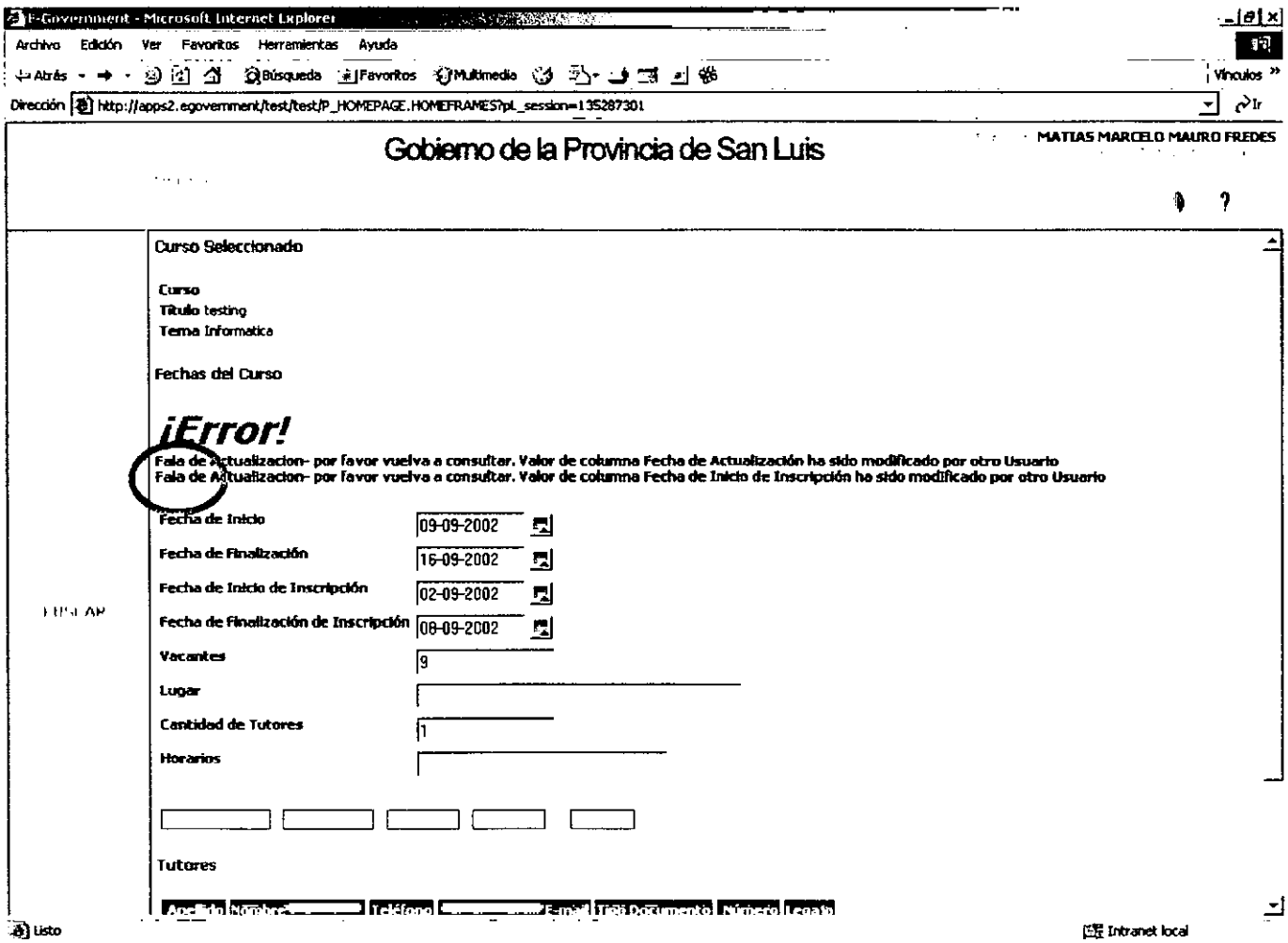
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar o Borrar Fechas de Dictado de Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-011

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
ERROR DE ORTOGRAFÍA EN MENSAJE.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

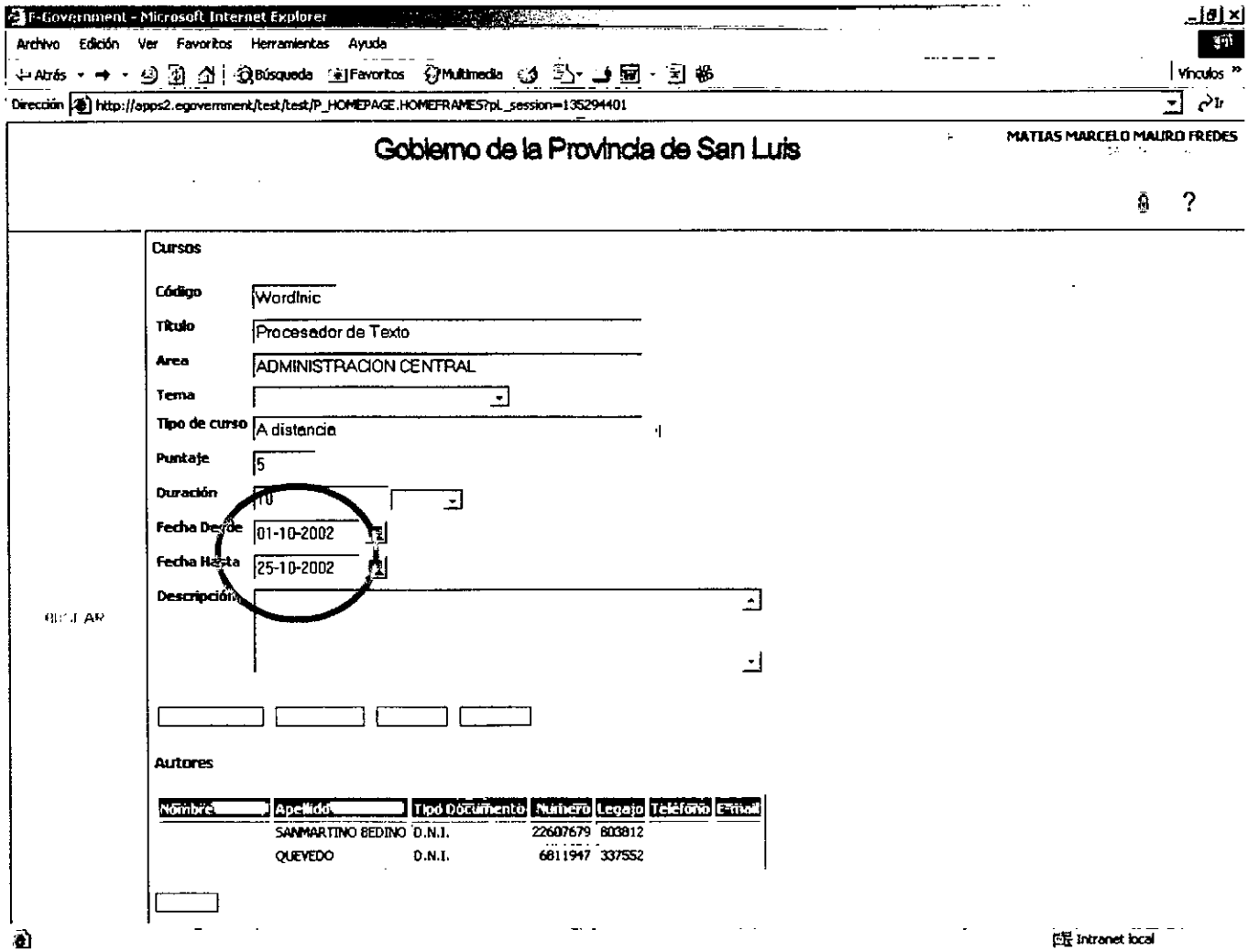
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Dictado**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Consultar Fechas de Dictado de Cursos**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

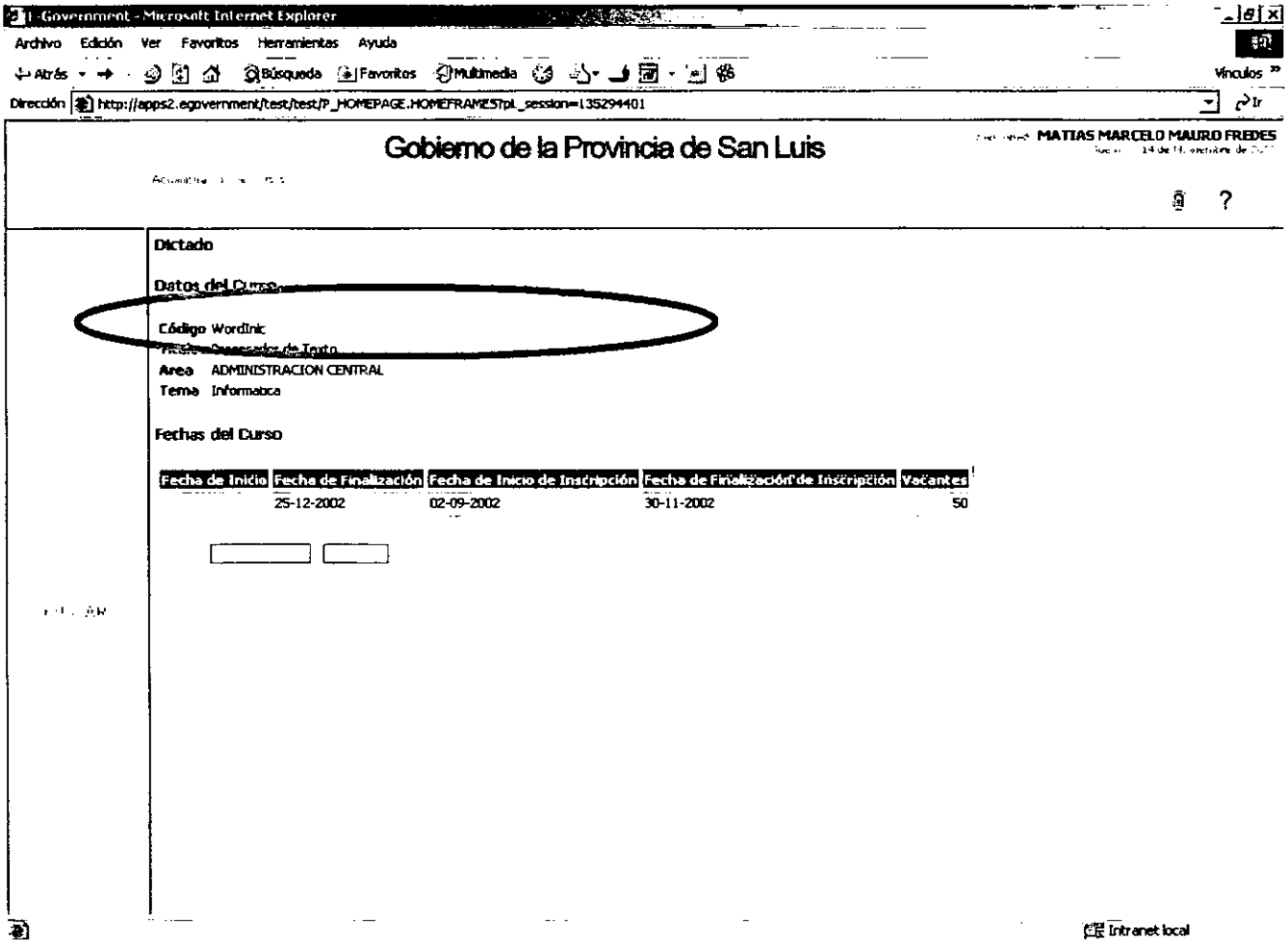
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-009

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
FECHA DESDE Y FECHA HASTA DE CURSOS NO SE ENCUENTRA RELACIONADO CON LAS FECHAS DE DICTADO DEL CURSO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

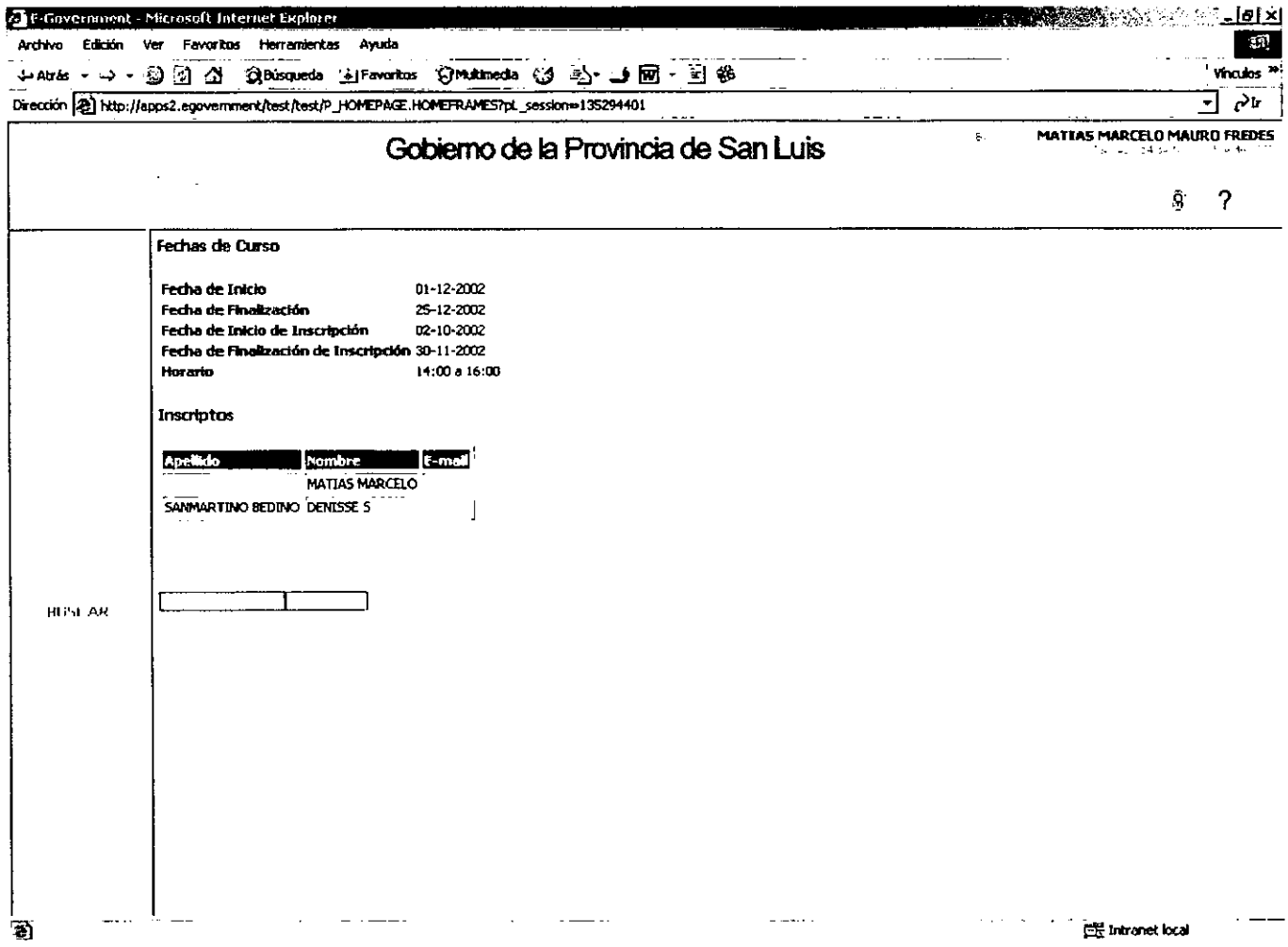
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Dictado**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar o Borrar Fechas de Dictado de Cursos**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-011

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
PERMITE BORRAR EL REGISTRO DE DICTADO DE UN CURSO CON ALUMNOS INSCRIPTOS EN EL MISMO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAMES?pL_session=135294401

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Curso
Título Procesador de Texto
Tema Informatica

Fechas del Curso

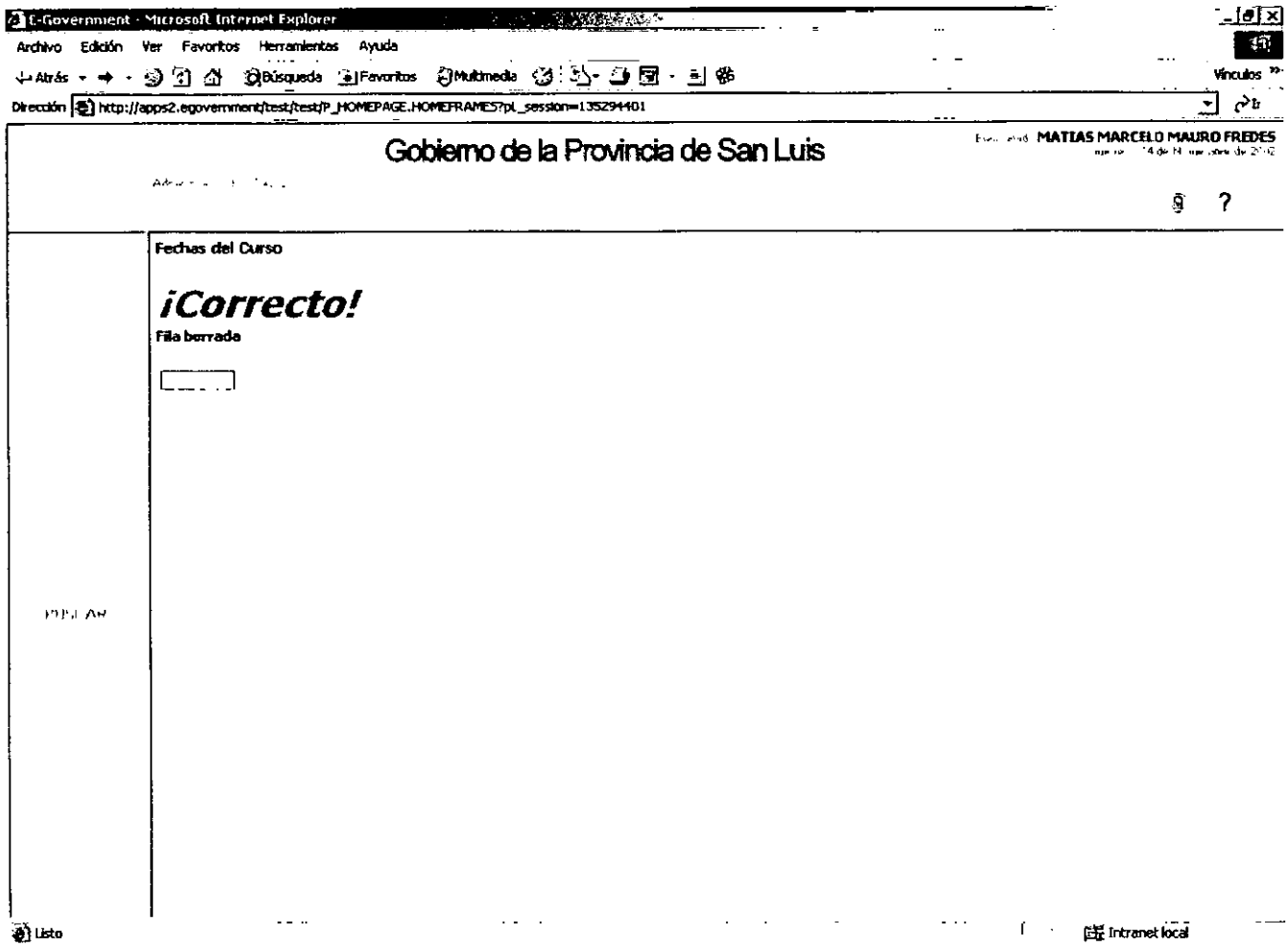
Fecha de Inicio	01-12-2002
Fecha de Finalización	25-12-2002
Fecha de Inicio de Inscripción	02-09-2002
Fecha de Finalización de Inscripción	30-11-2002
Vacantes	50
Lugar	
Cantidad de Tutores	3
Horarios	14:00 a 16:00

Tutores

Apellido	Nombre	Teléfono	E-mail	Tipo Documento	Número	Legajo
	HECTOR REGALADO			D.N.I.	6805896	357014
	OSCAR HUGO			Libreta de Enrolamiento	6811117	959544

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

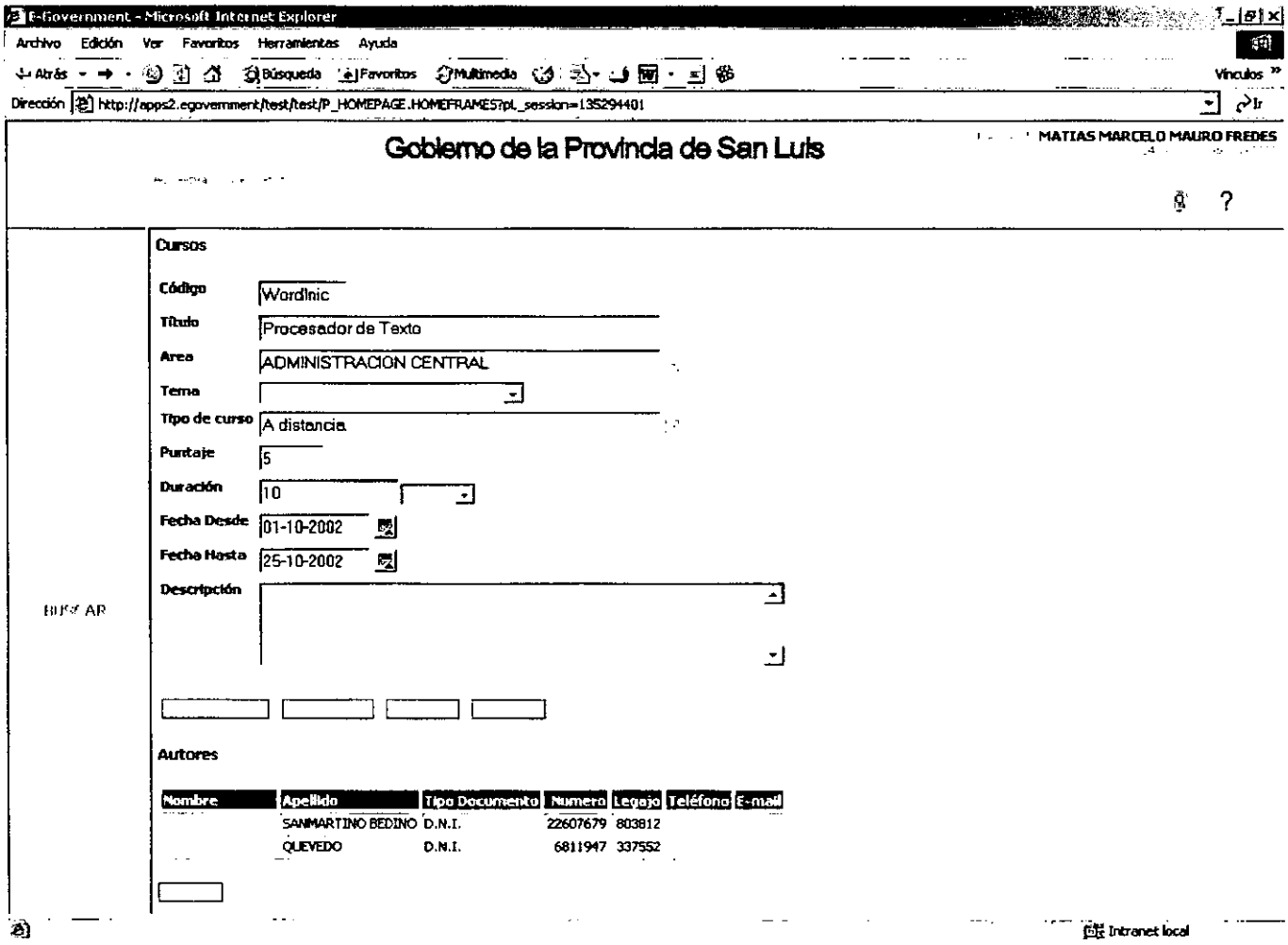
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar o Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-005

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
PERMITE EN LA ACTUALIZACIÓN BORRAR LA FECHA DE INICIO O FIN DE INSCRIPCIÓN INDISTINTAMENTE.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAME5?pL_session=135287301

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Dictado

Datos del Curso

Código 458
Título testing
Area SECRETARIA DE ESTADO DE TECNOLOGIAS DE LA INFORMACION
Tema Informatica

Fechas del Curso

Fecha de Inicio	Fecha de Finalización	Fecha de Inicio de Inscripción	Fecha de Finalización de Inscripción	Vacantes
	17-10-2002	02-05-2002	28-06-2002	850
	16-09-2002	02-09-2002	10-09-2002	852
	16-09-2002			
	16-09-2002	02-09-2002	08-09-2002	9
	16-09-2002		08-09-2002	9

Listo

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAME5?pL_session=135287301

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

?

Dictado

Datos del Curso

Código 458
Título testing
Area SECRETARIA DE ESTADO DE TECNOLOGIAS DE LA INFORMACION
Tema Informatica

Fechas del Curso

Fecha de Inicio	Fecha de Finalización	Fecha de Inicio de Inscripción	Fecha de Finalización de Inscripción	Vacantes
	17-10-2002	02-05-2002	28-08-2002	850
	16-09-2002	02-09-2002	10-09-2002	852
	16-09-2002			
	16-09-2002	02-09-2002		9
	16-09-2002		08-09-2002	9

Listo Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

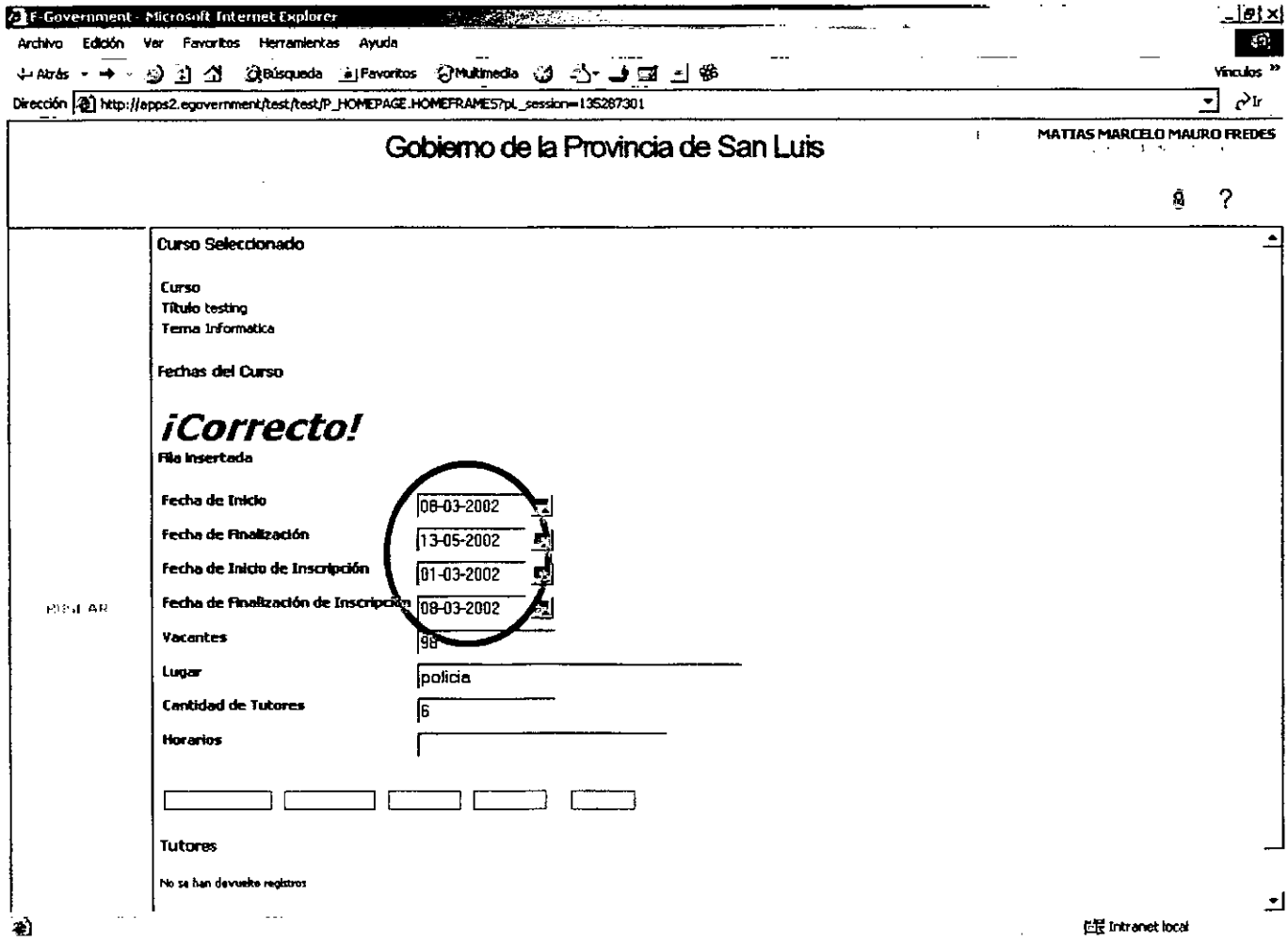
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen Presencial**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Exámen Presencial – Actualizar o Borrar Exámen Presencial**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-031 **CP-EG-SC-TA-032**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
ADMITE CARGAR Y ACTUALIZAR CURSOS QUE SON EN LÍNEA CON FECHAS ANTERIORES A LA ACTUAL, ES DECIR QUE NADIE PODRÍA INSCRIBIRSE EN LOS MISMOS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso – Actualizar o Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

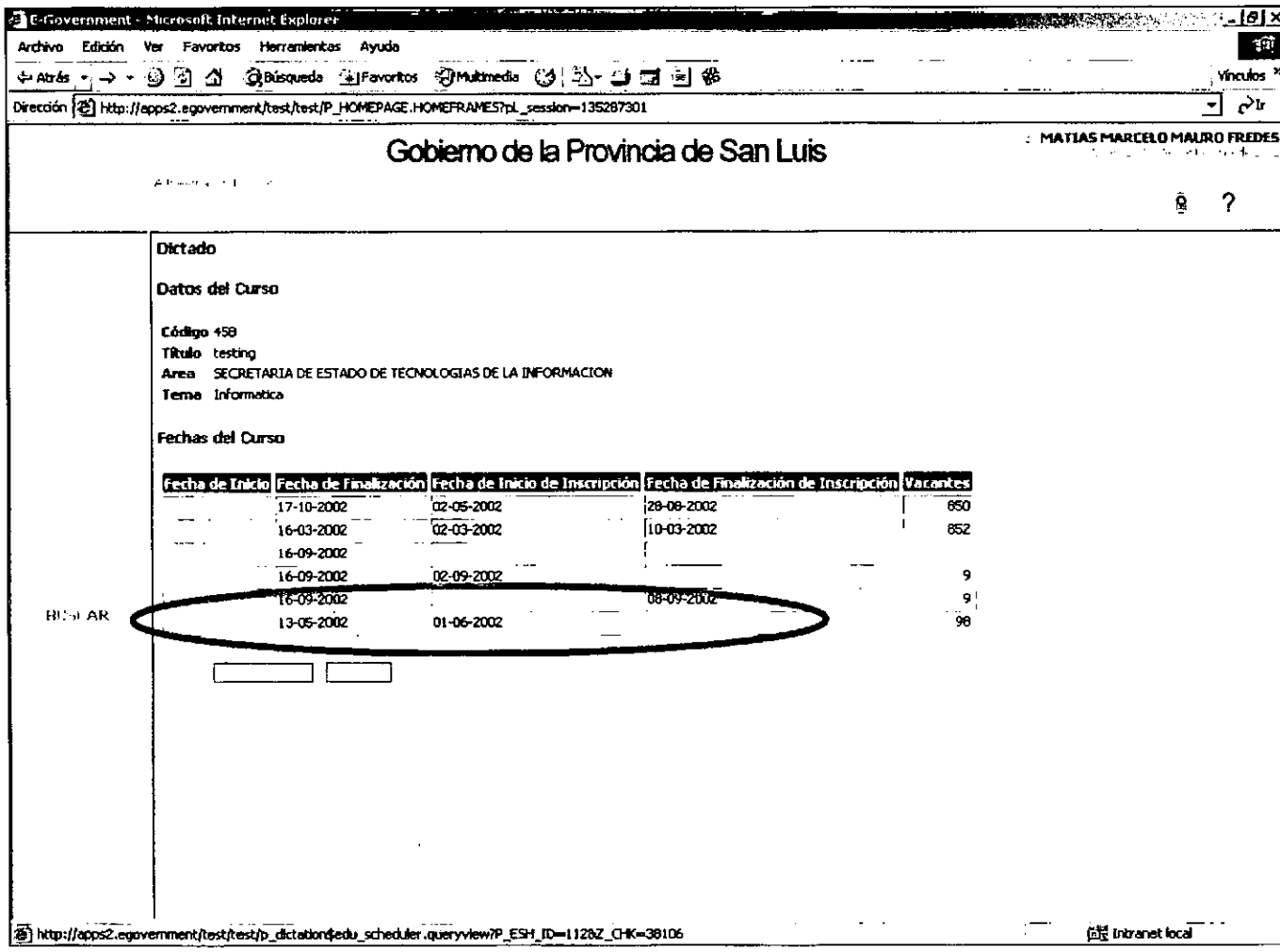
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002 **CP-EG-SC-TA-005**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

SI NO SE CARGA LA FECHA DE FINALIZACIÓN DE INSCRIPCIÓN, ADMITE FECHA DE INICIO DE INSCRIPCIÓN POSTERIOR A LA FECHA DE FINALIZACIÓN EL CURSO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAMES?pL_session=135287301

Gobierno de la Provincia de San Luis

Bienvenid MATIAS MARCELO MAURO FREDES

Curso Seleccionado

Curso
Título testing
Tema Informática

Fechas del Curso

¡Correcto!
Fila actualizada

Fecha de Inicio	08-09-2002
Fecha de Finalización	17-10-2002
Fecha de Inicio de Inscripción	02-05-2003
Fecha de Finalización de Inscripción	
Vacantes	850
Lugar	seti
Cantidad de Tutores	20
Horarios	8:30 10:30

Tutores

No se han devuelto registros

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

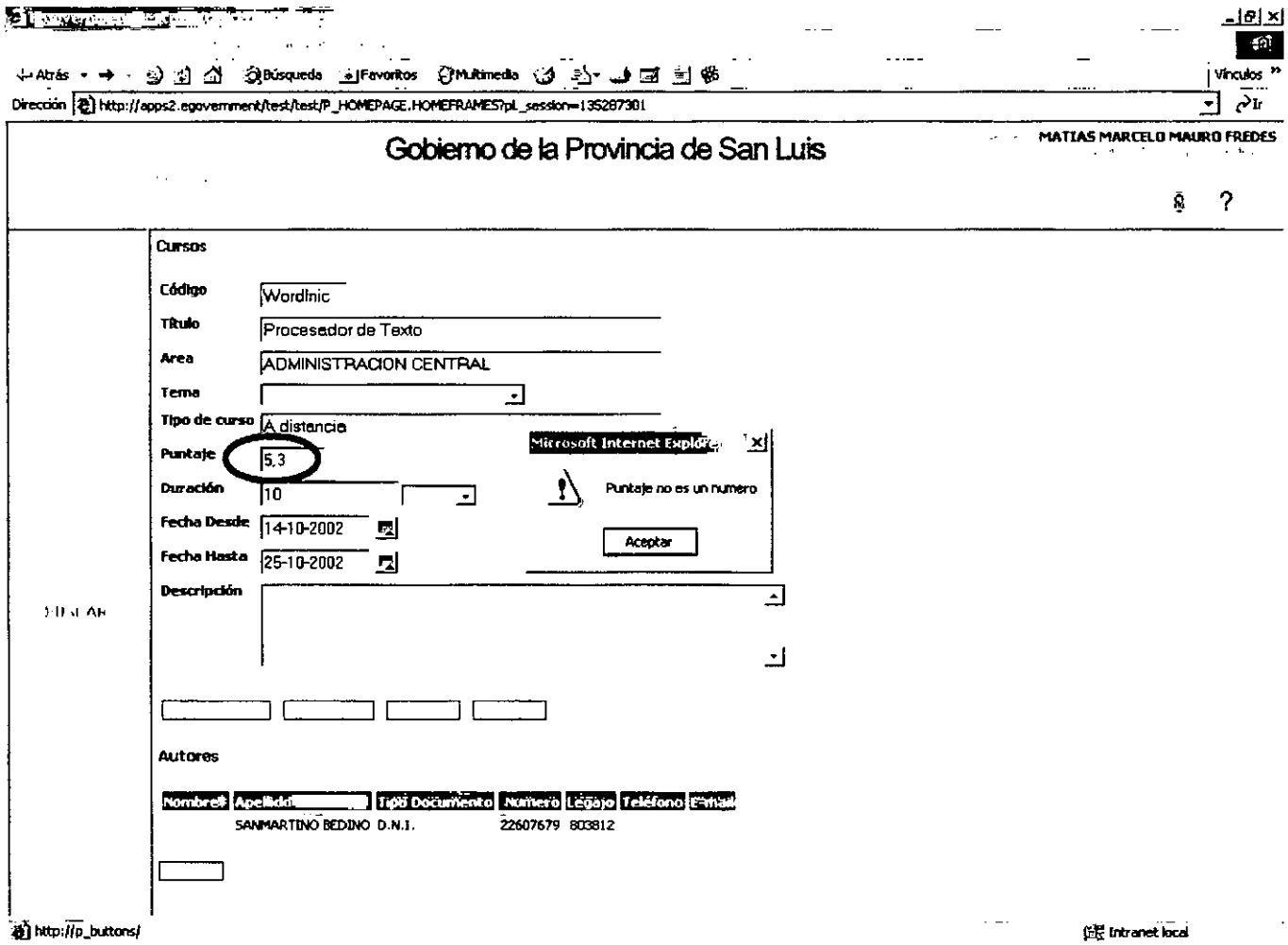
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso - Actualizar o Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002 **CP-EG-SC-TA-005**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
NO ADMITE NÚMERO DE PUNTAJE CON DECIMALES.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Tema**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Actualizar o Borrar Tema**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

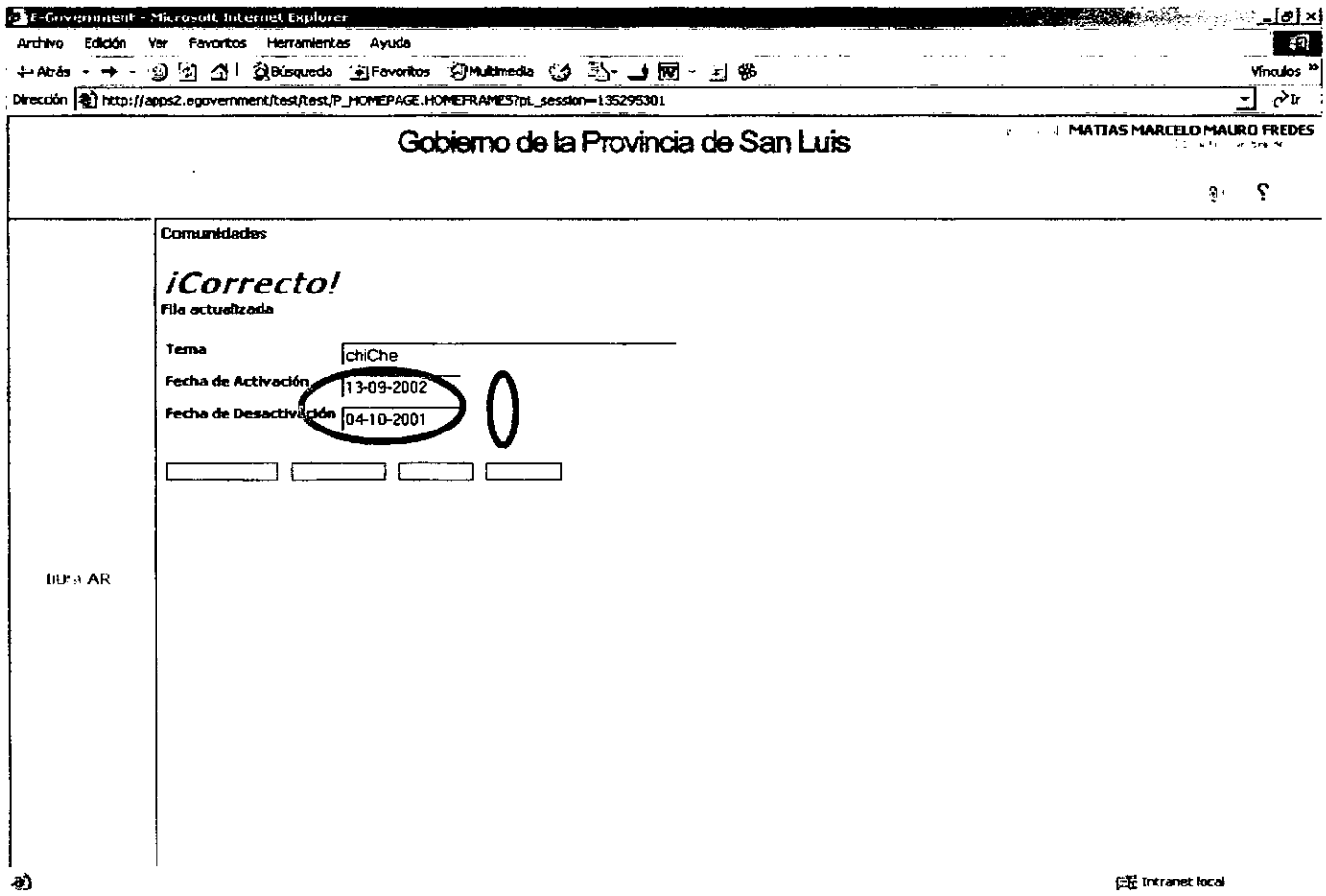
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-042

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

EN LA ACTUALIZACIÓN ADMITE FECHA DE ACTIVACIÓN MAYOR QUE LA FECHA DE DESACTIVACIÓN.
FALTAN ICONOS DE CALENDARIOS PARA LAS FECHAS.
ADMITE FECHA DE ACTIVACIÓN ANTERIORES A LA FECHA ACTUAL.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

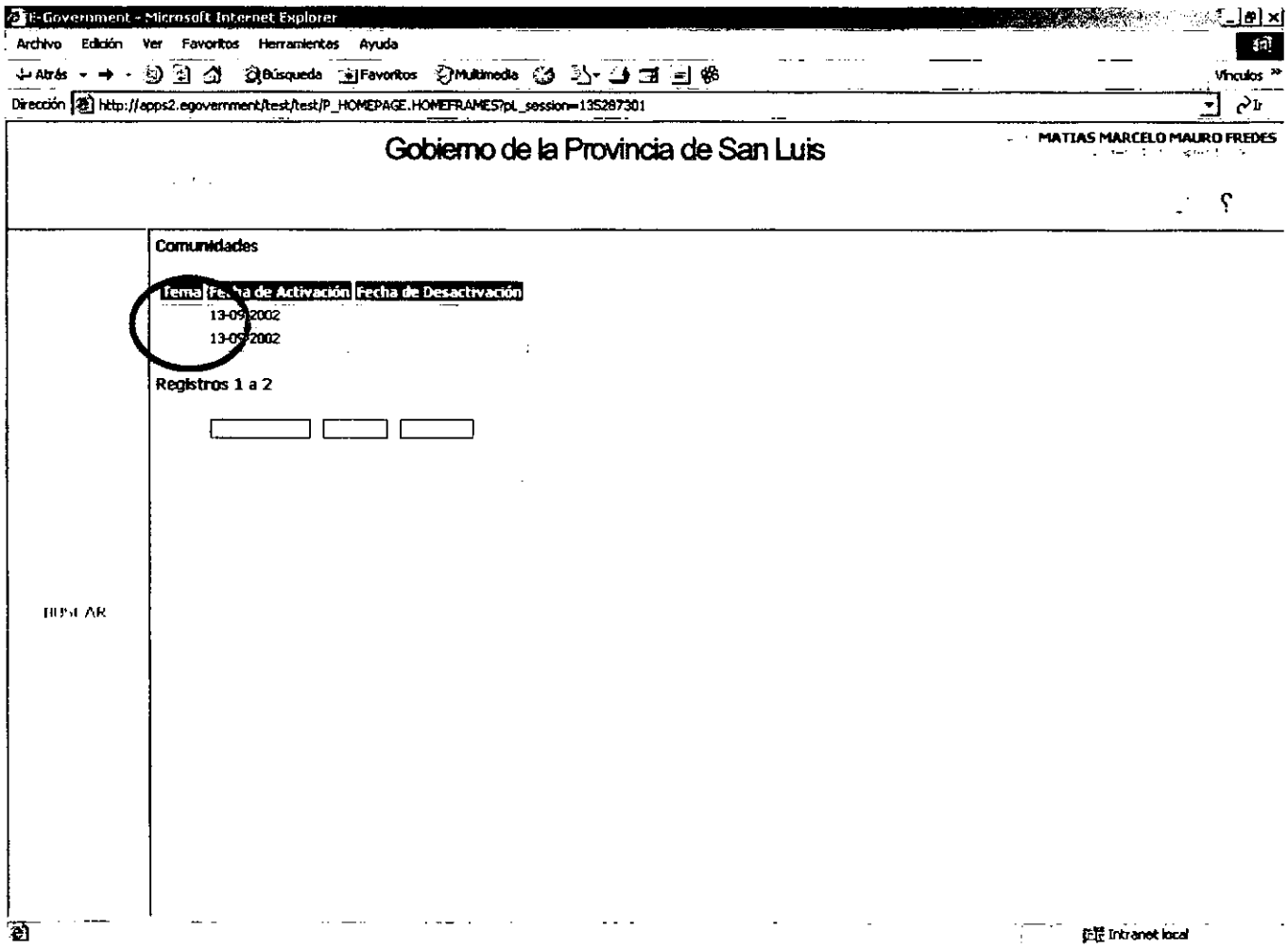
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Tema**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nuevo Tema**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-041

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
ADMITE MULTIPLICIDAD DE TEMAS, SÓLO VERIFICA POR STRINGS IDÉNTICOS EN (LETRAS MINÚSCULAS/ MAYÚSCULAS, ESPACIOS, ETC). ES CASE SENSITIVE CUANDO NO DEBERÍA SERLO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

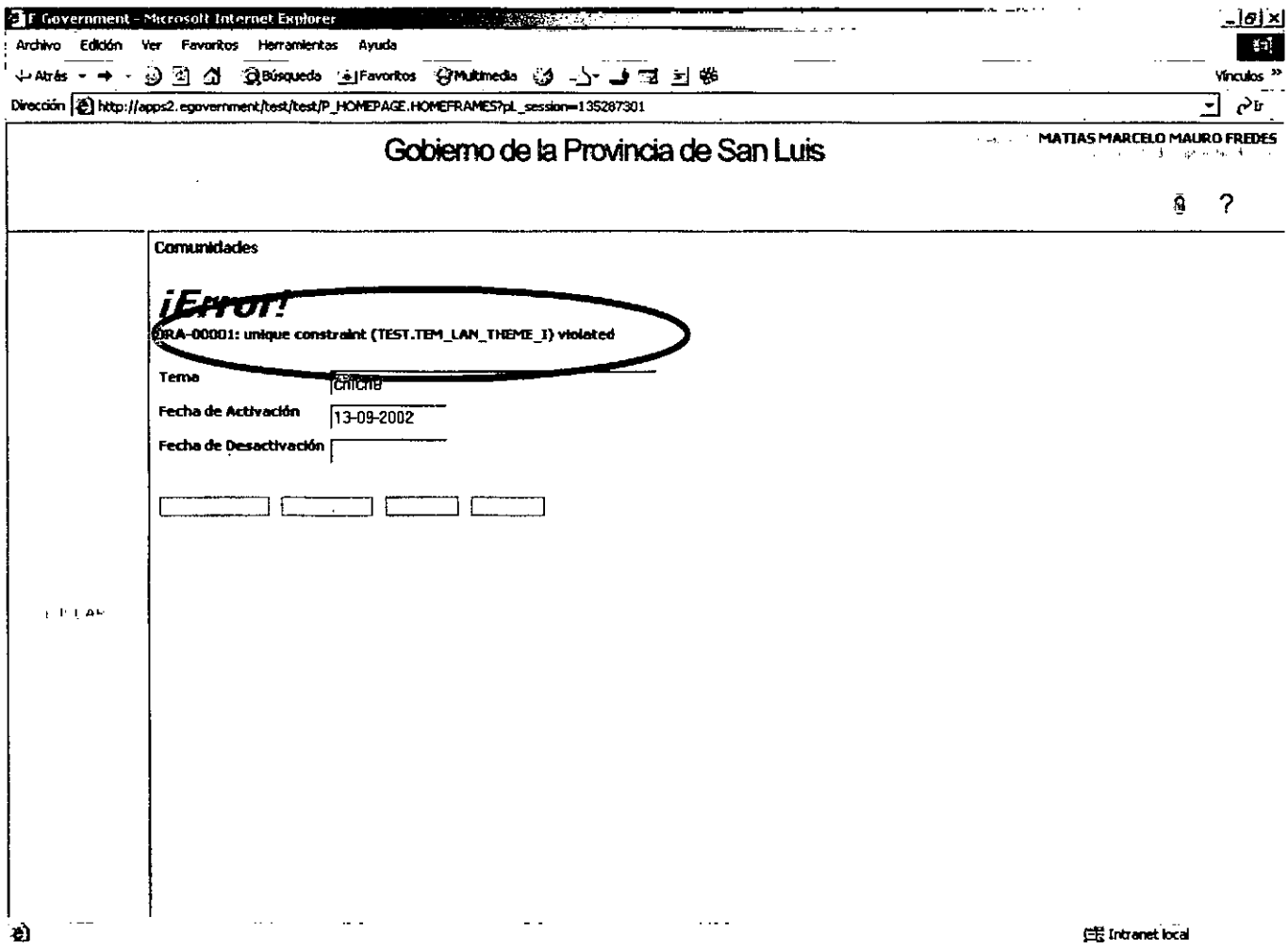
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **PARÁMETROS**
MÓDULO: **Tema**	FUNCIÓN: **Actualizar o Borrar Tema**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-042

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
MENSAJE DE ERROR EN INGLÉS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

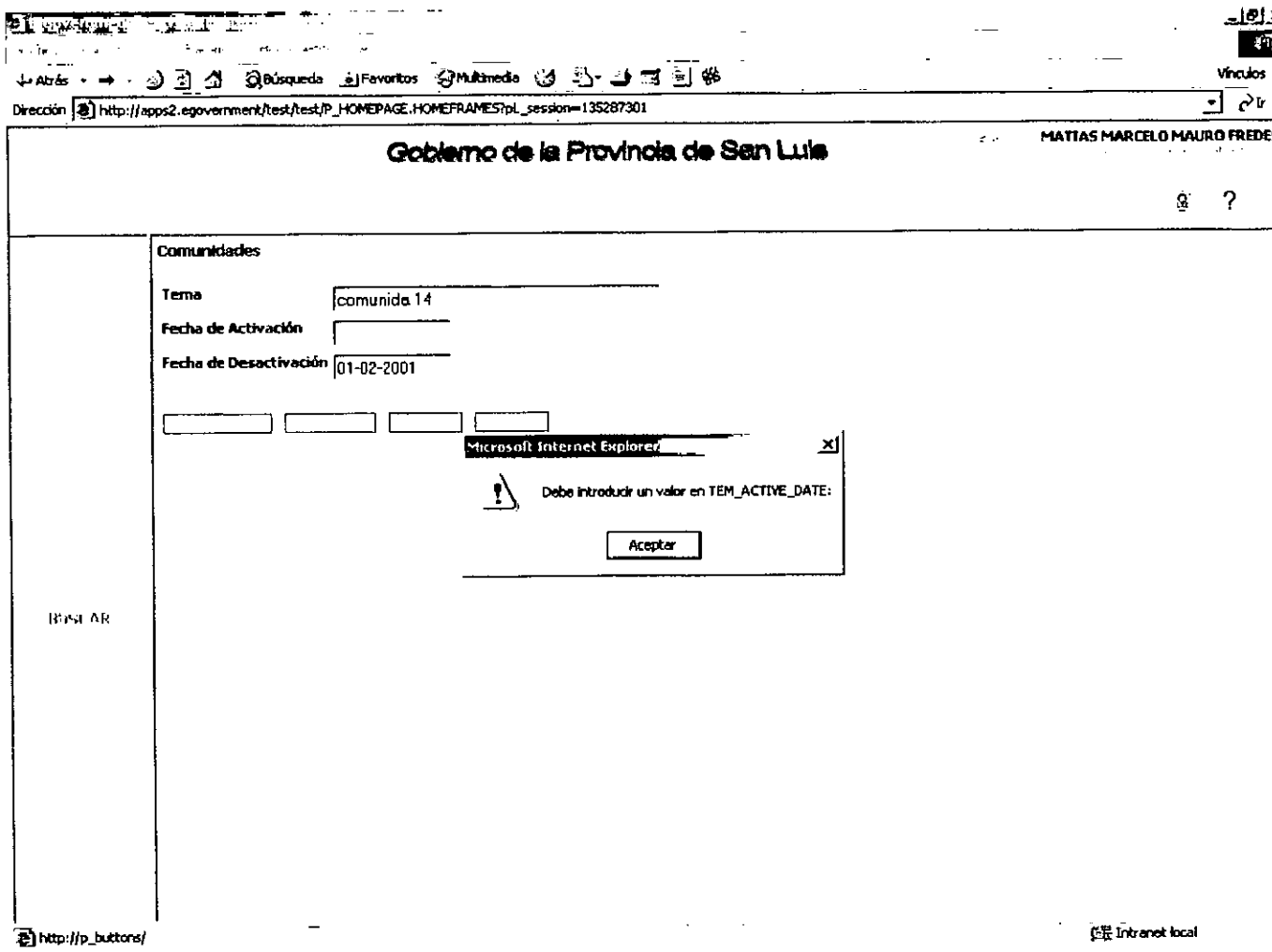
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Tema**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Actualizar o Borrar Tema**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-042

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
MENSAJE NO CLARO PARA EL USUARIO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Autores**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nuevo Autor**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

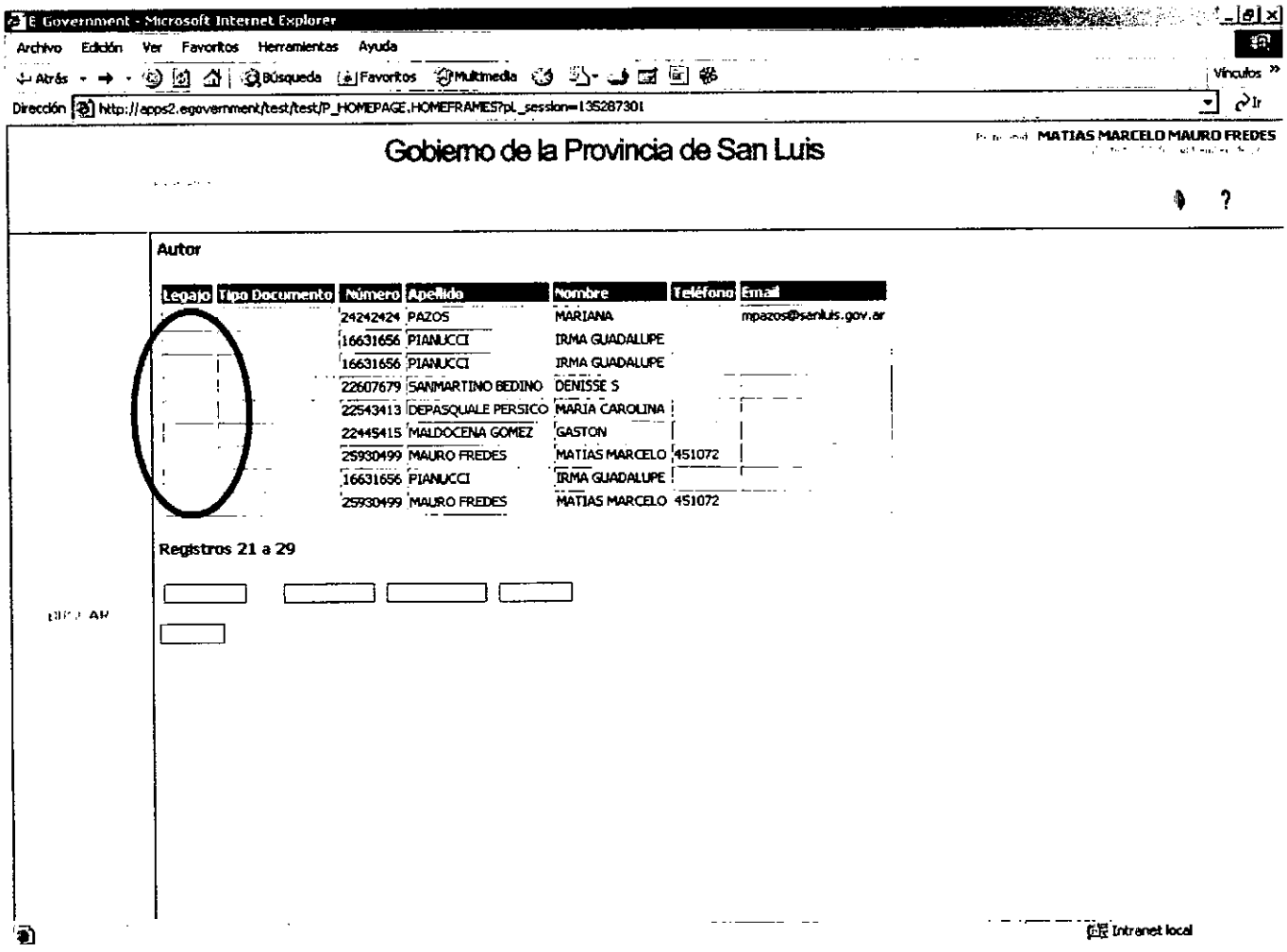
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-046

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

PERMITE CARGAR EN LA TABLA DE AUTORES MÚLTIPLES VECES EL MISMO AUTOR.
¿POR QUÉ EL CAMPO TIPO DE DOCUMENTOS SON LINKS A LOS REGISTROS DE LOS DATOS DE AUTORES?

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Autores**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nuevo Autor – Actualizar o Borrar Autores**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

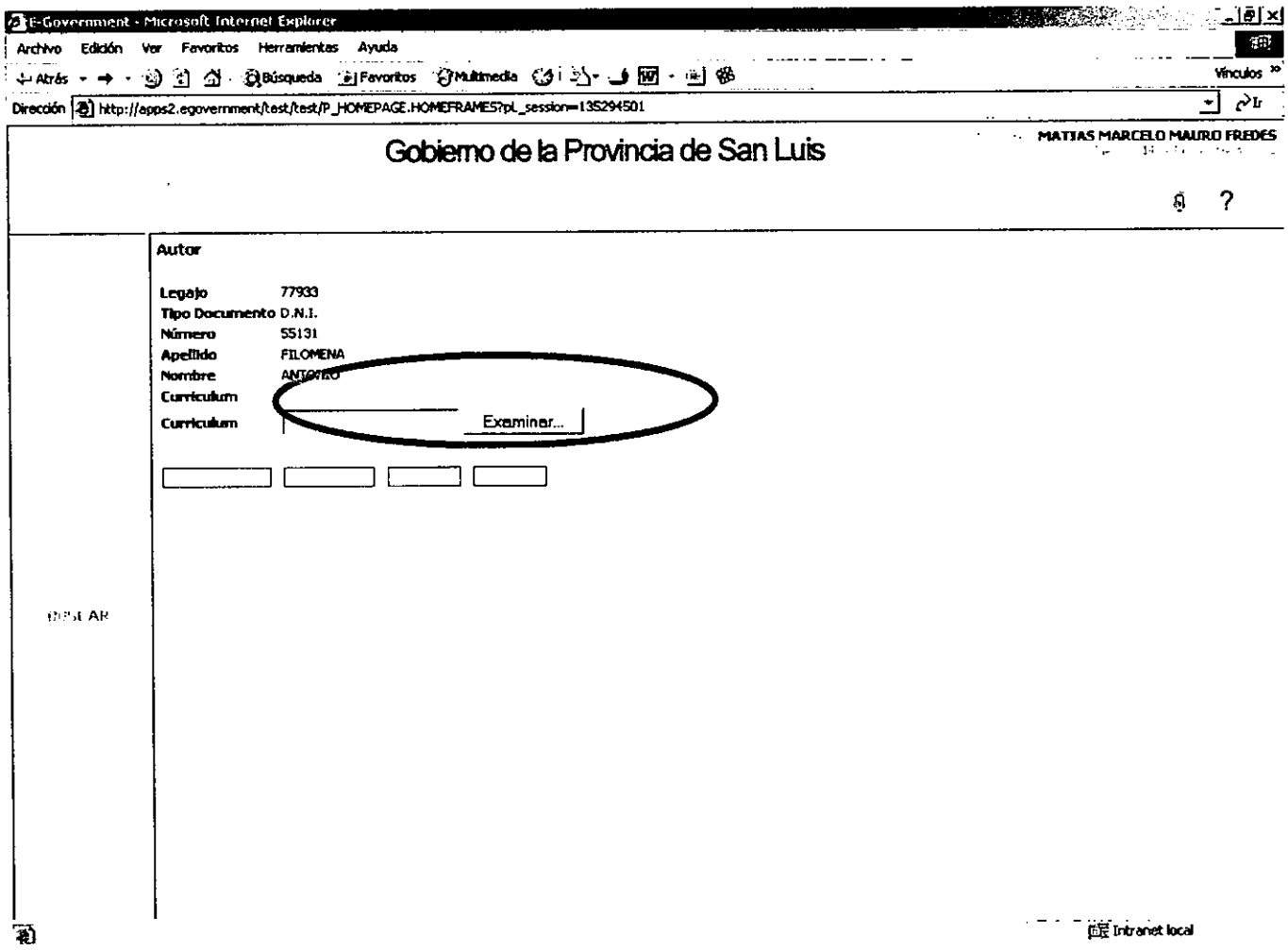
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-046 **CP-EG-SC-TA-047**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

DA ERROR EN LA CARGA / ACTUALIZACIÓN DE CURRÍCULUM DE AUTOR CON PASOS DE VARIOS NIVELES Y / O NOMBRES EXTENSOS.
EN EL NOMBRE DEL LINK ASOCIADO AL CURRÍCULUM FIGURA EL PASO DE DONDE SE CARGÓ, LIMITANDO LA LONGITUD DEL MISMO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAMES?pl_session=135294501

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

We were unable to process your request at this time.

```
Error occured while accessing
     "/test/test/p_author$author.actionview"
at Thu Nov 14 15:03:50 2002

OWS-02007: db execute


  PL/SQL Cartridge SERVICE: test/test
  PROCEDURE: p_author$author.actionview


  PARAMETERS:
  ===========
  P_AUT_ID:
   17
  P_AUT_LAN_ID:
   1
  FILE:
   C:\Documents and Settings\Administrador\Mis documentos\Libros\tesis - doc vers
  O_AUT_CREATION_USR_ID:
   6
  O_AUT_USR_ID:
```

Listo

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Tutores**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nuevo Tutor**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

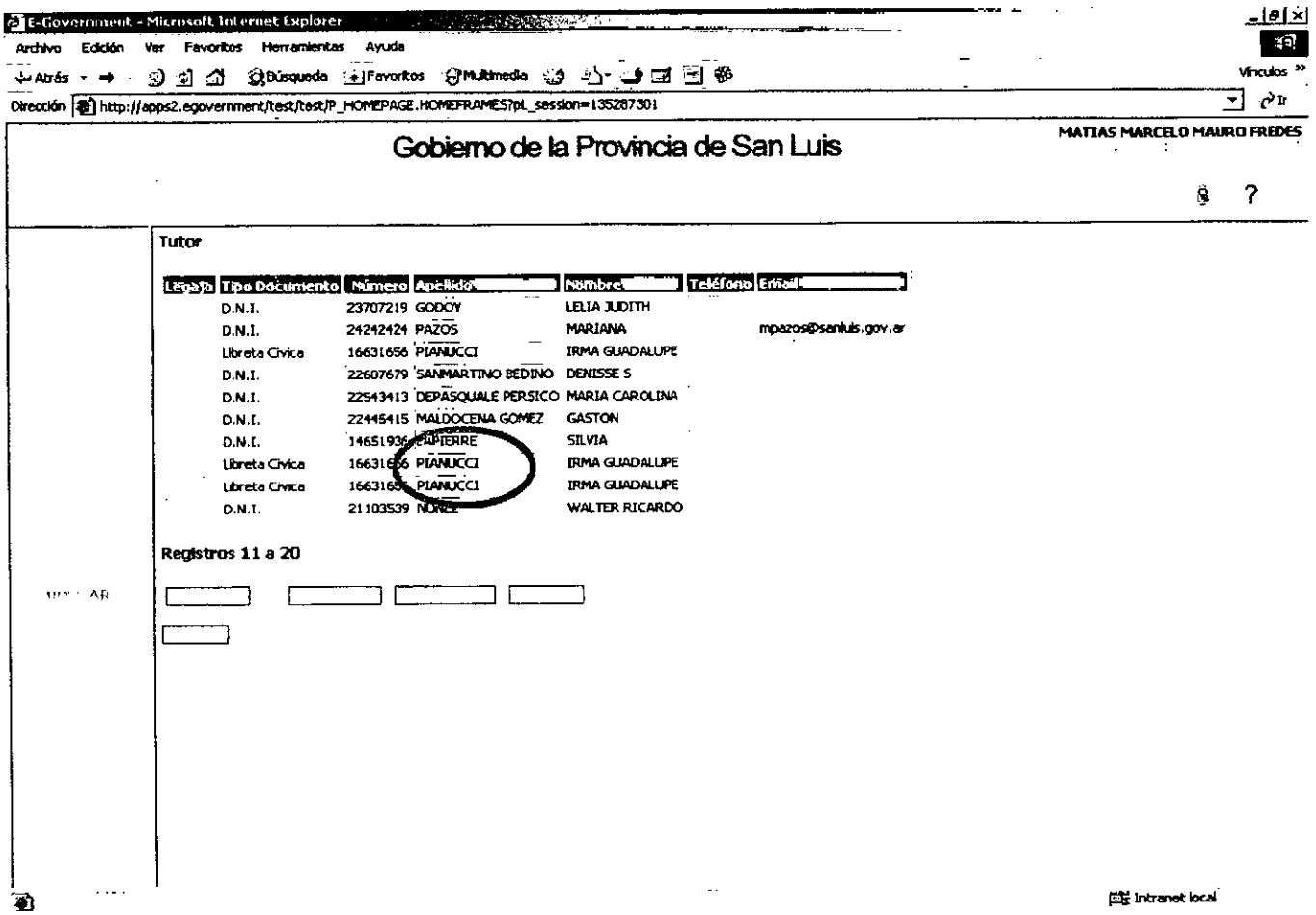
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-051

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

ADMITE MÚLTIPLES REGISTROS DE TUTORES IGUALES.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Tutores**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nuevo Tutor – Actualizar o Borrar Tutores**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

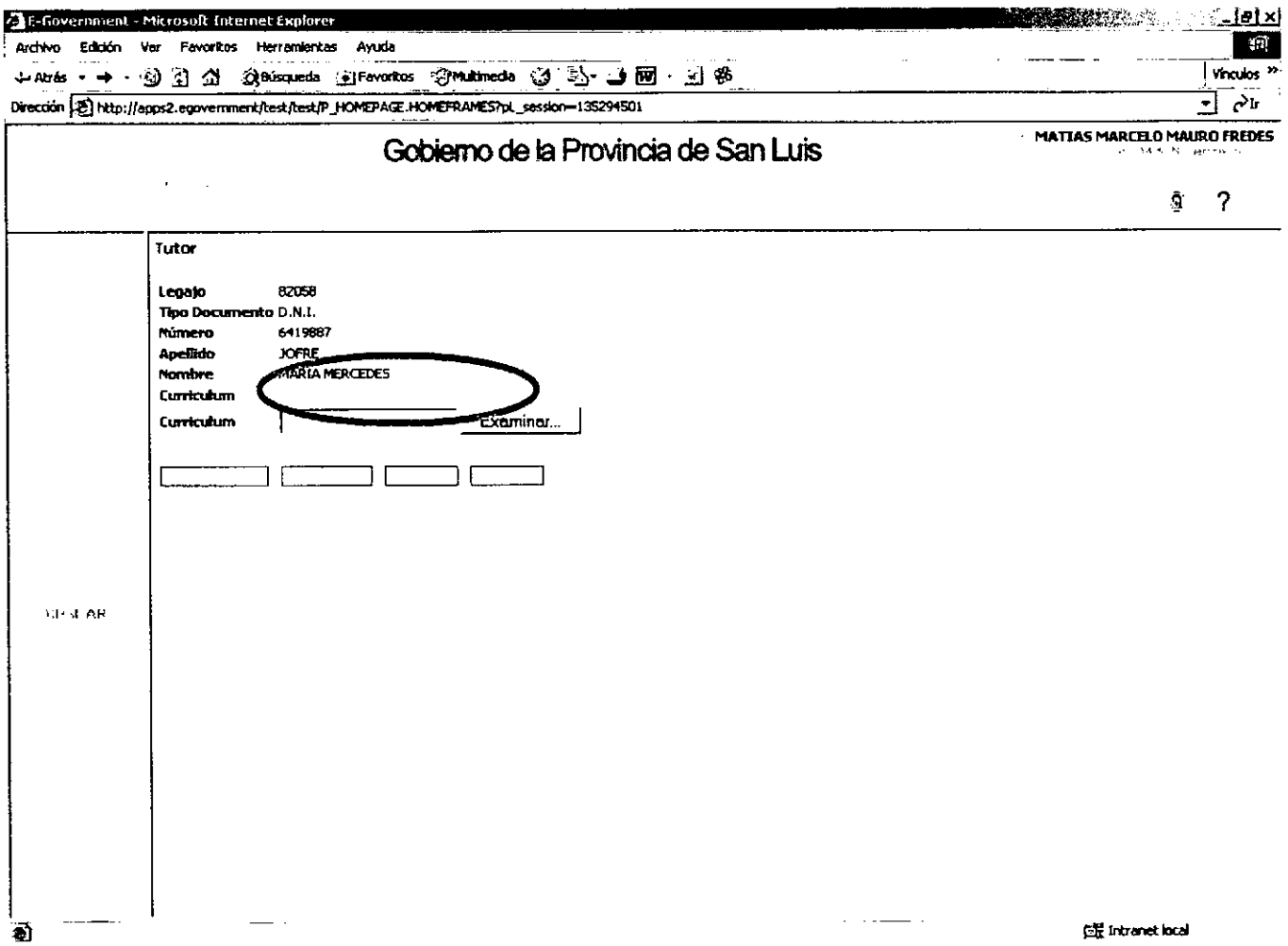
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-051 **CP-EG-SC-TA-052**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

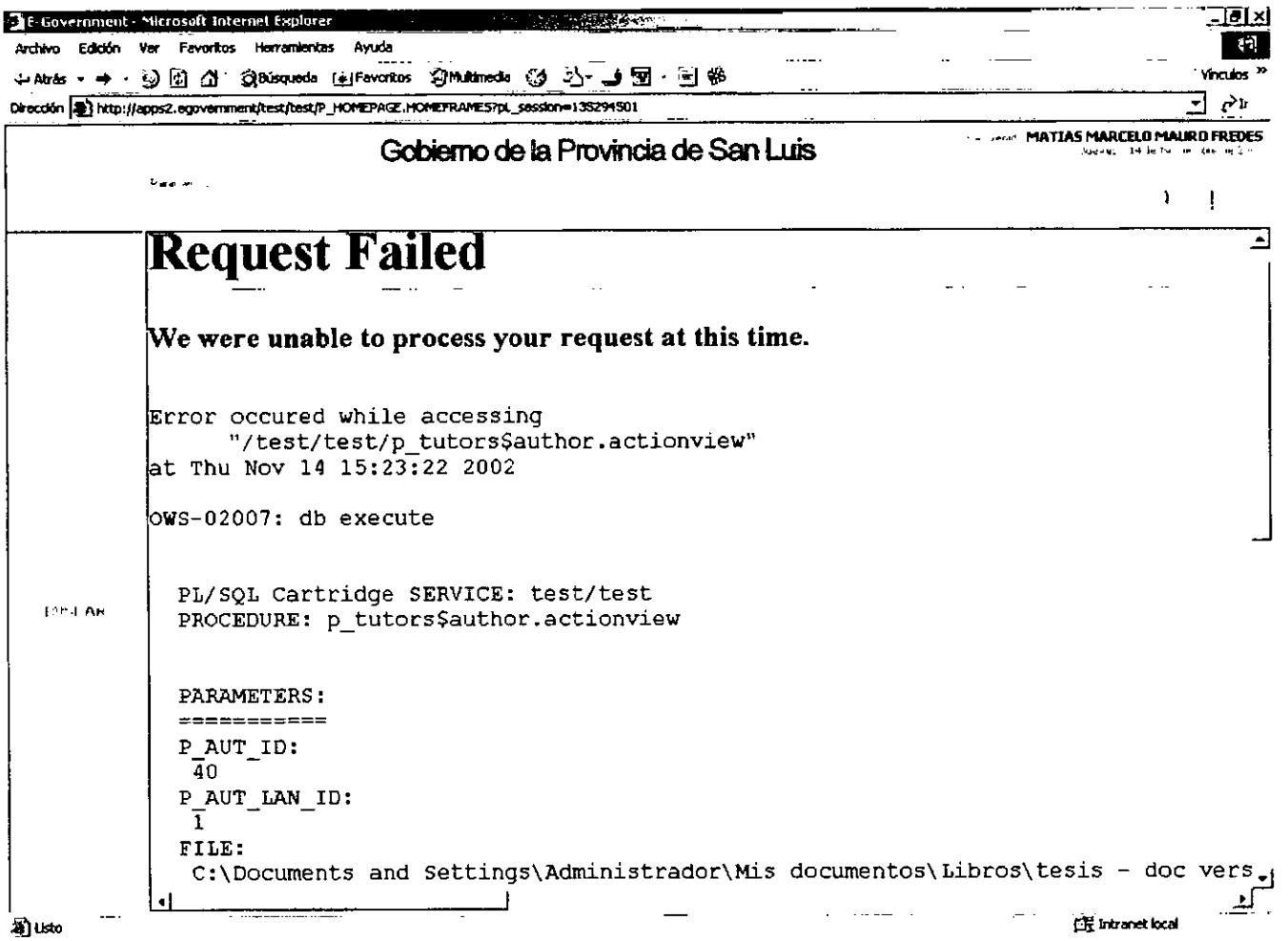
OBSERVACIONES

DA ERROR EN LA CARGA / ACTUALIZACIÓN DE CURRÍCULUM CON PASOS DE VARIOS NIVELES. EN EL NOMBRE DEL LINK ASOCIADO AL CURRÍCULUM FIGURA EL PASO DE DONDE SE CARGÓ, LIMITANDO LA LONGITUD DEL MISMO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Unidades de Tiempo**	SUBSISTEMA: **PARÁMETROS** FUNCIÓN: **Nueva Unidad de Tiempo – Actualizar o Borrar Unidad de Tiempo**

PLAN DE PRUEBAS ASOCIADO	
ID: **002**	VERSIÓN: **01**

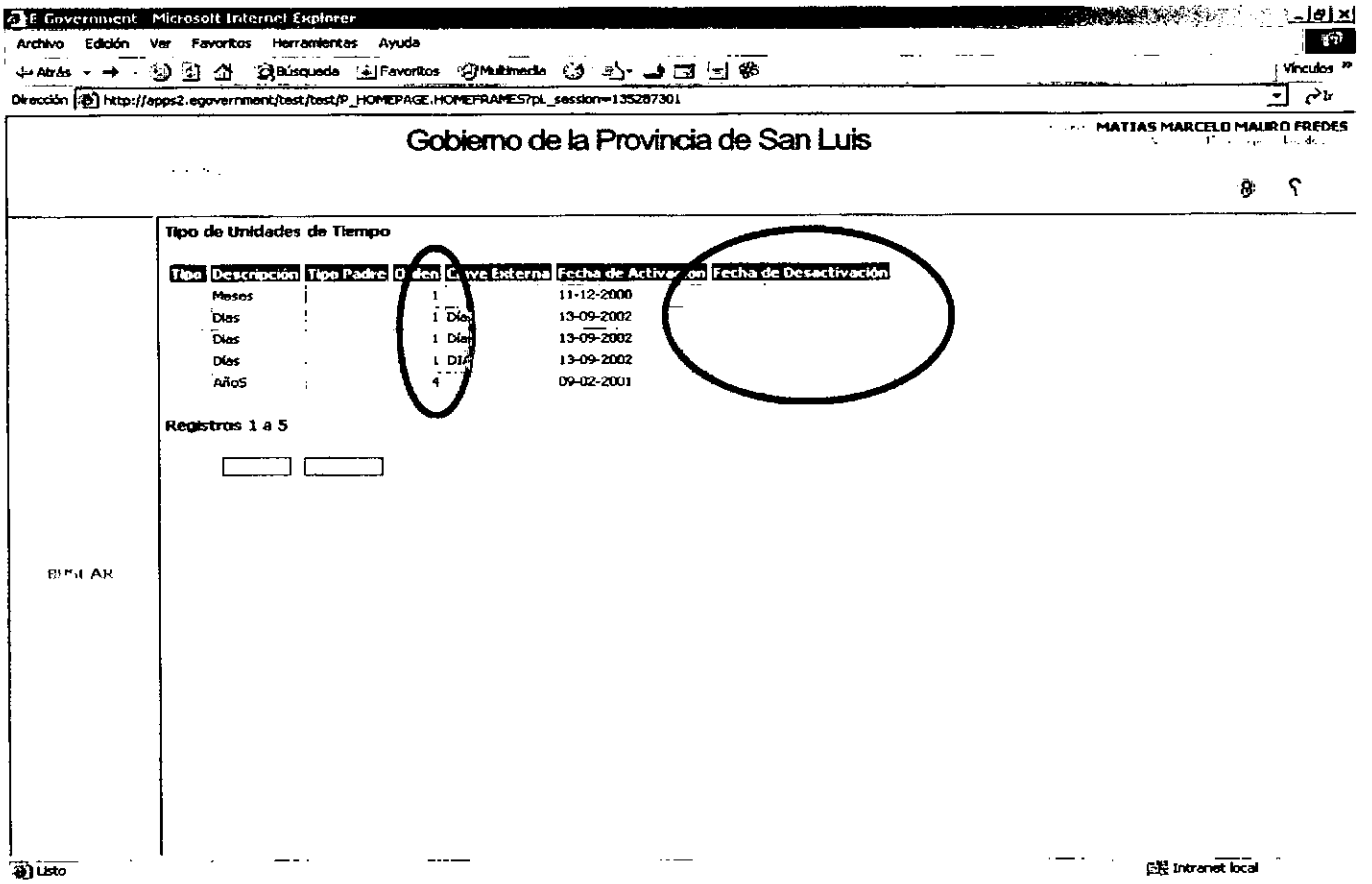
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-055 **CP-EG-SC-TA-056**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

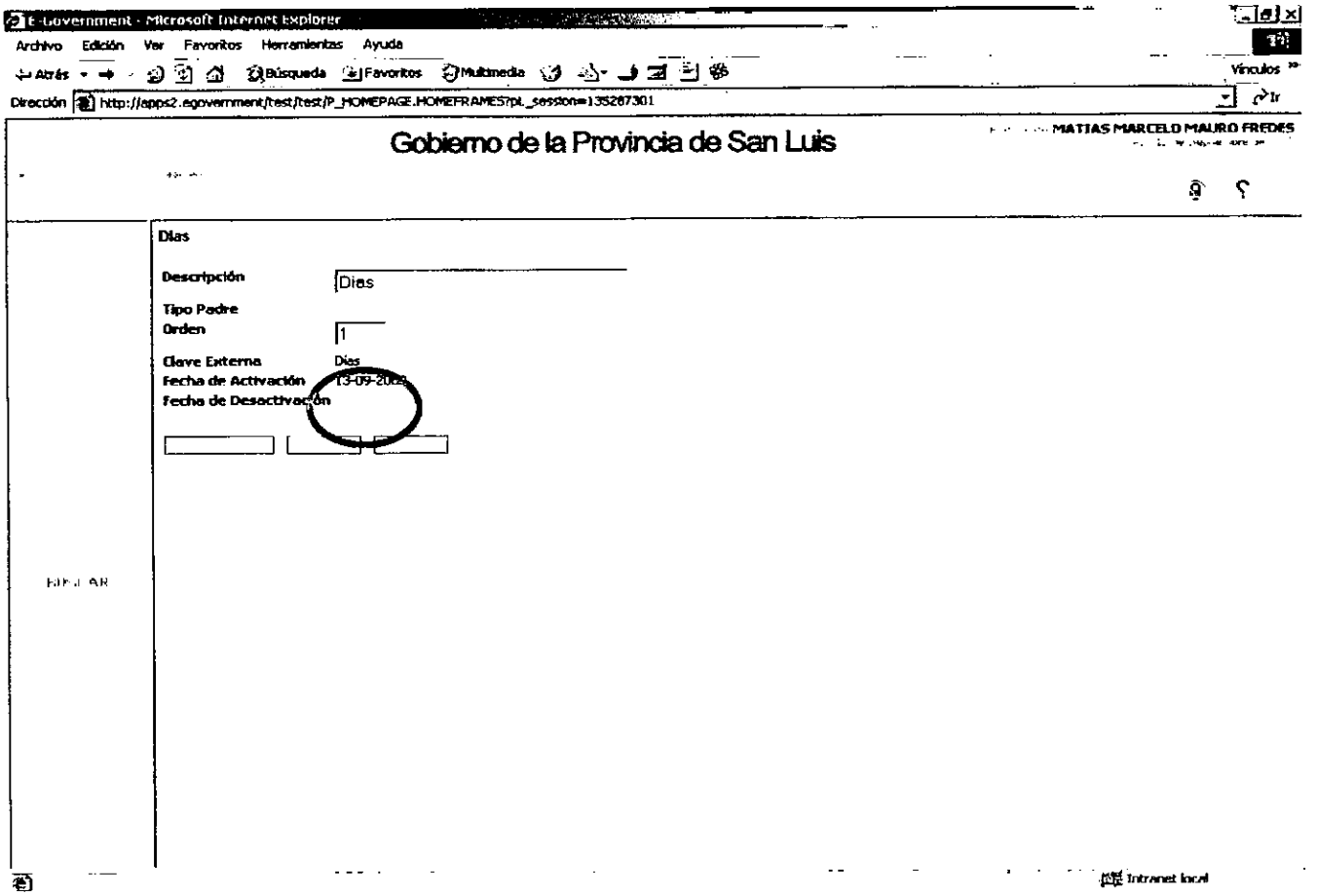
OBSERVACIONES

NO ADMITE REACTIVAR UNA UNIDAD DE TIEMPO DADA DE BAJA.
BORRAR UNA UNIDAD DE TIEMPO ELIMINA DIRECTAMENTE EL REGISTRO, EN VEZ DE COLOCARLE UNA FECHA DE DESACTIVACIÓN.
¿POR QUÉ PERMITE IGUAL ORDEN EN LAS UNIDADES DE TIEMPO?

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAMES?pl_session=135295301

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Tipo de Unidades de Tiempo

Tipo	Destripción	Tipo Padre	Orden	Clave Externa	Fecha de Activación	Fecha de Desactivación
	Meses		1		11-12-2000	
	Días		1	DIA	13-09-2002	
	Años		4		09-02-2001	

Registros 1 a 3

Listo

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Capítulos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Capítulos de un Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

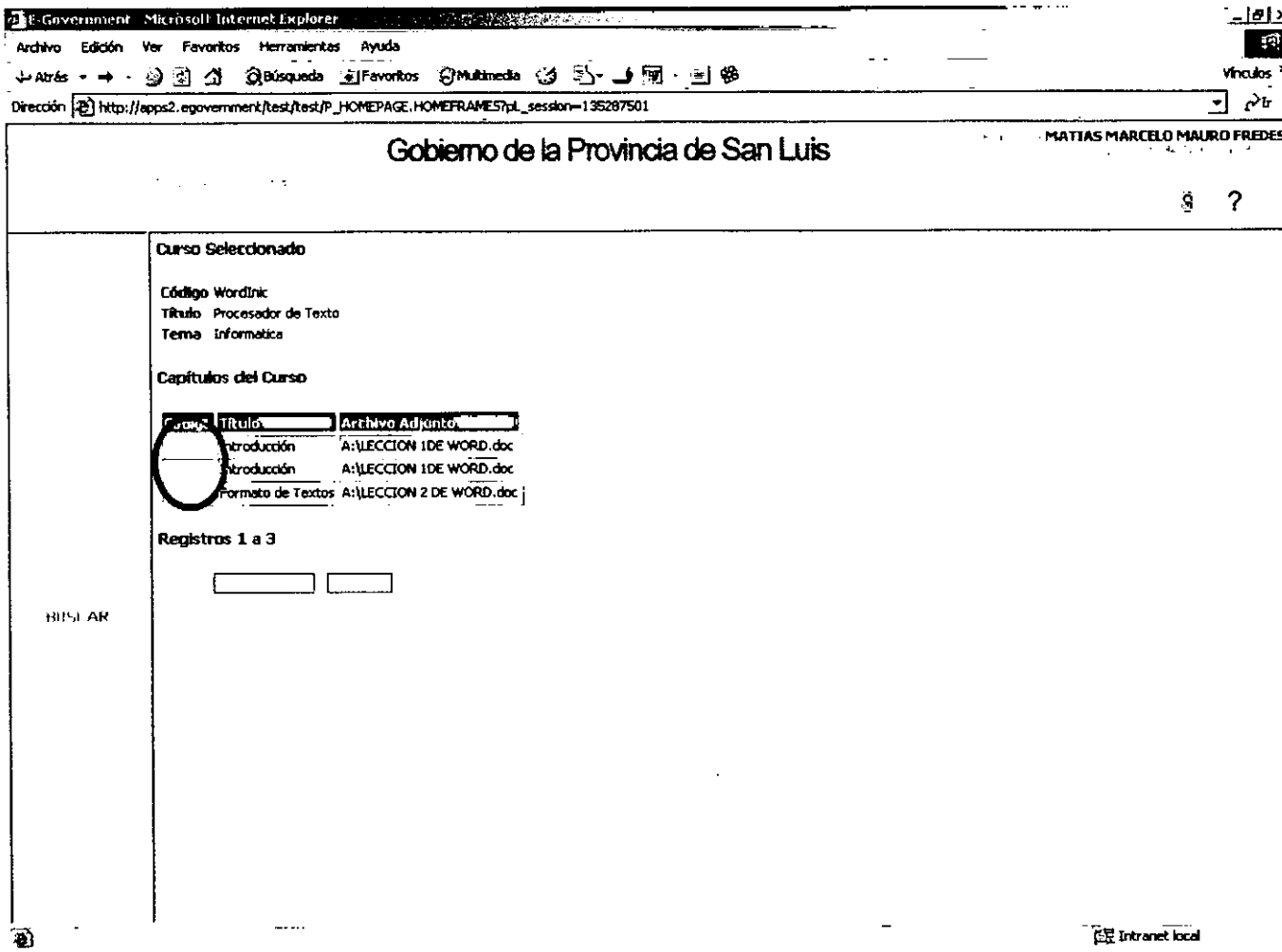
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-015

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

EL CÓDIGO DE CAPÍTULO DEBERÍA SER ÚNICO.
ADMITE MULTIPLICIDAD DE REGISTROS DEL MISMO CAPÍTULO PARA UN MISMO CURSO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

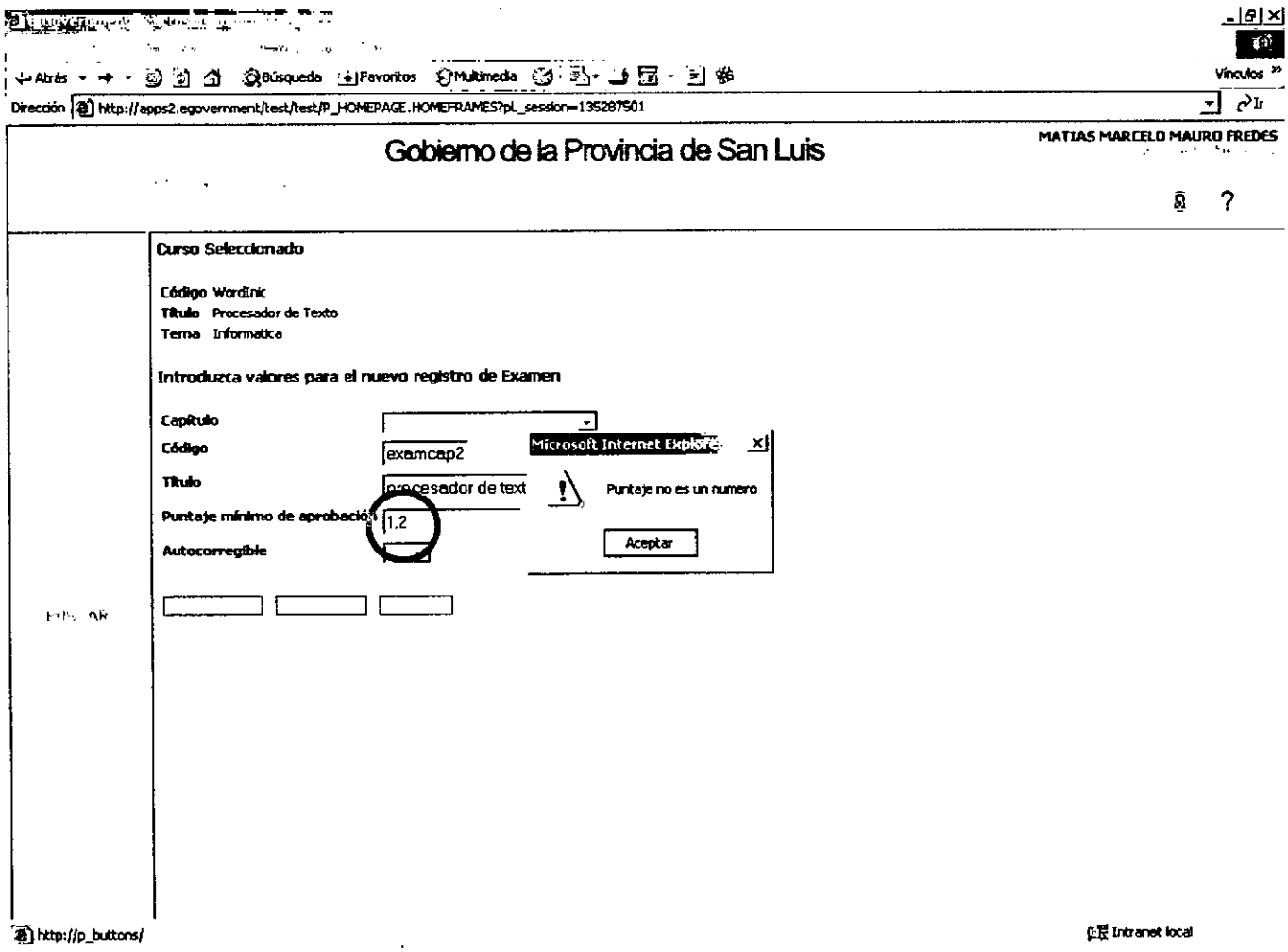
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **ABM Cursos**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Curso - Actualizar o Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-002 **CP-EG-SC-TA-005**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EN LA CARGA DE UN NUEVO CURSO COMO EN LA ACTUALIZACIÓN NO ADMITE NÚMERO REALES POSITIVOS PARA EL PUNTAJE MÍNIMO DE APROBACIÓN.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

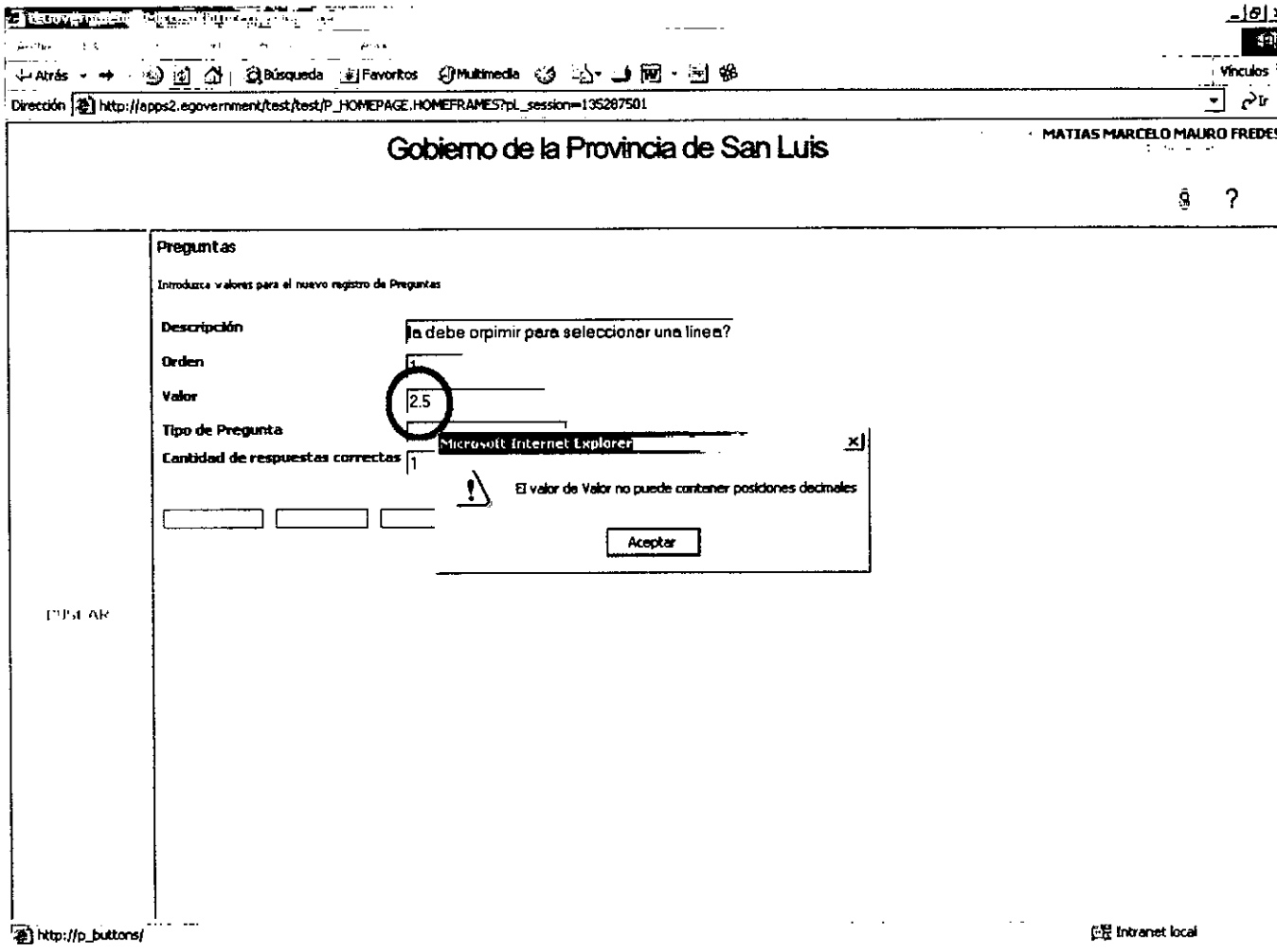
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Pregunta - Actualizar o Borrar preguntas**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-024 **CP-EG-SC-TA-025**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
NO ADMITE VALORES DECIMALES PARA EL VALOR DE LA PREGUNTA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

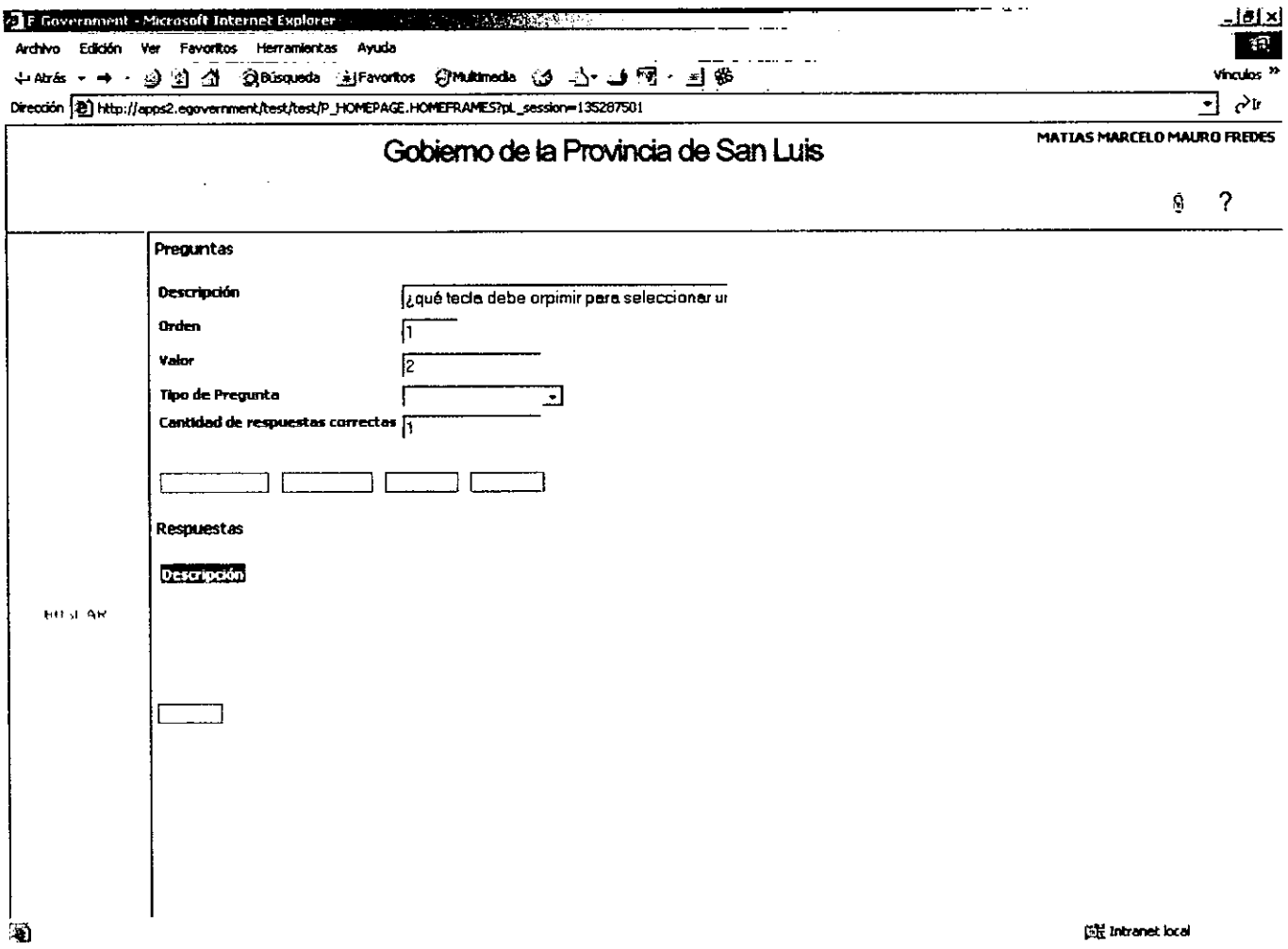
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Respuesta – Actualizar o Borrar Respuestas**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

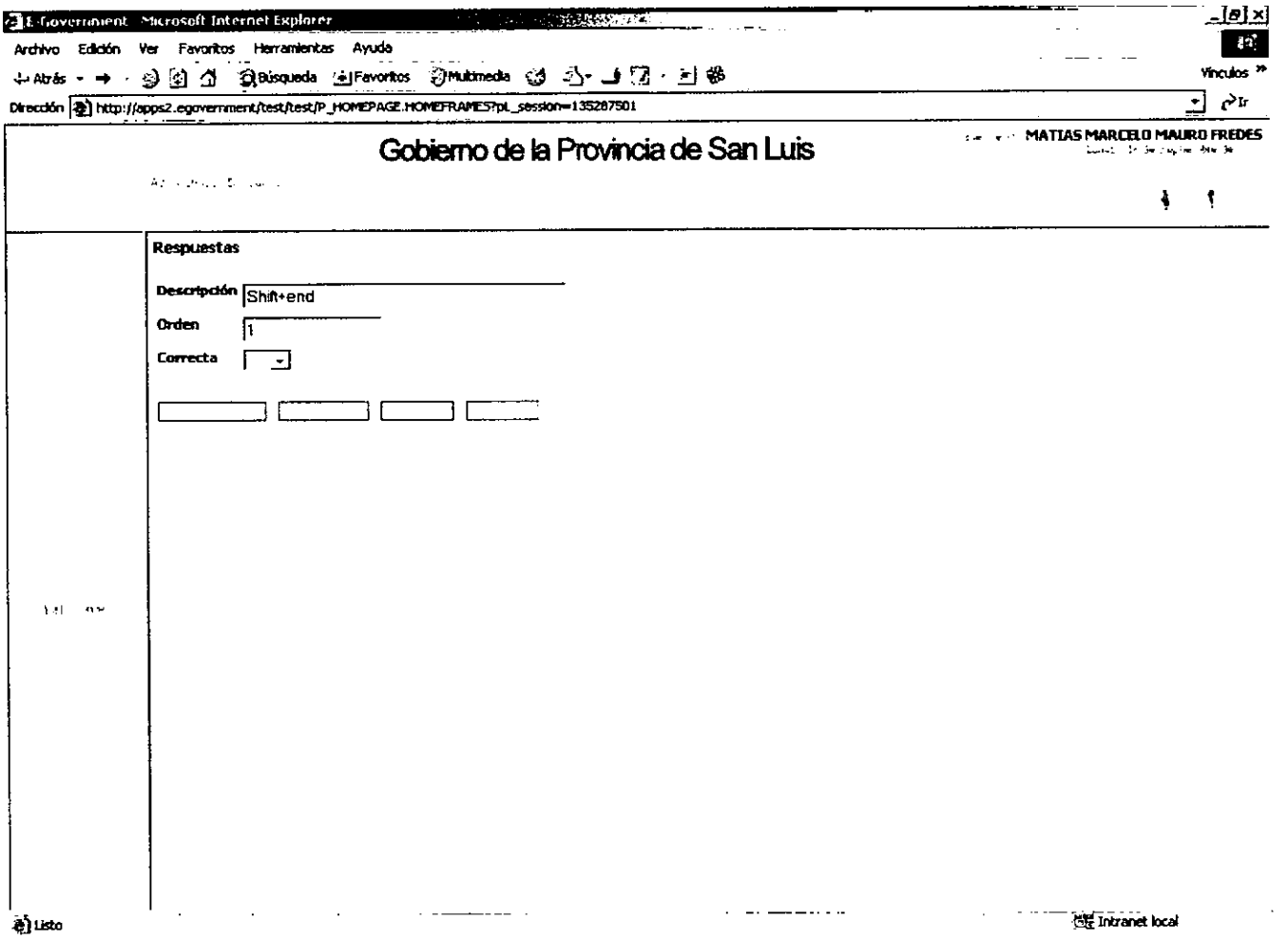
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-027 **CP-EG-SC-TA-028**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

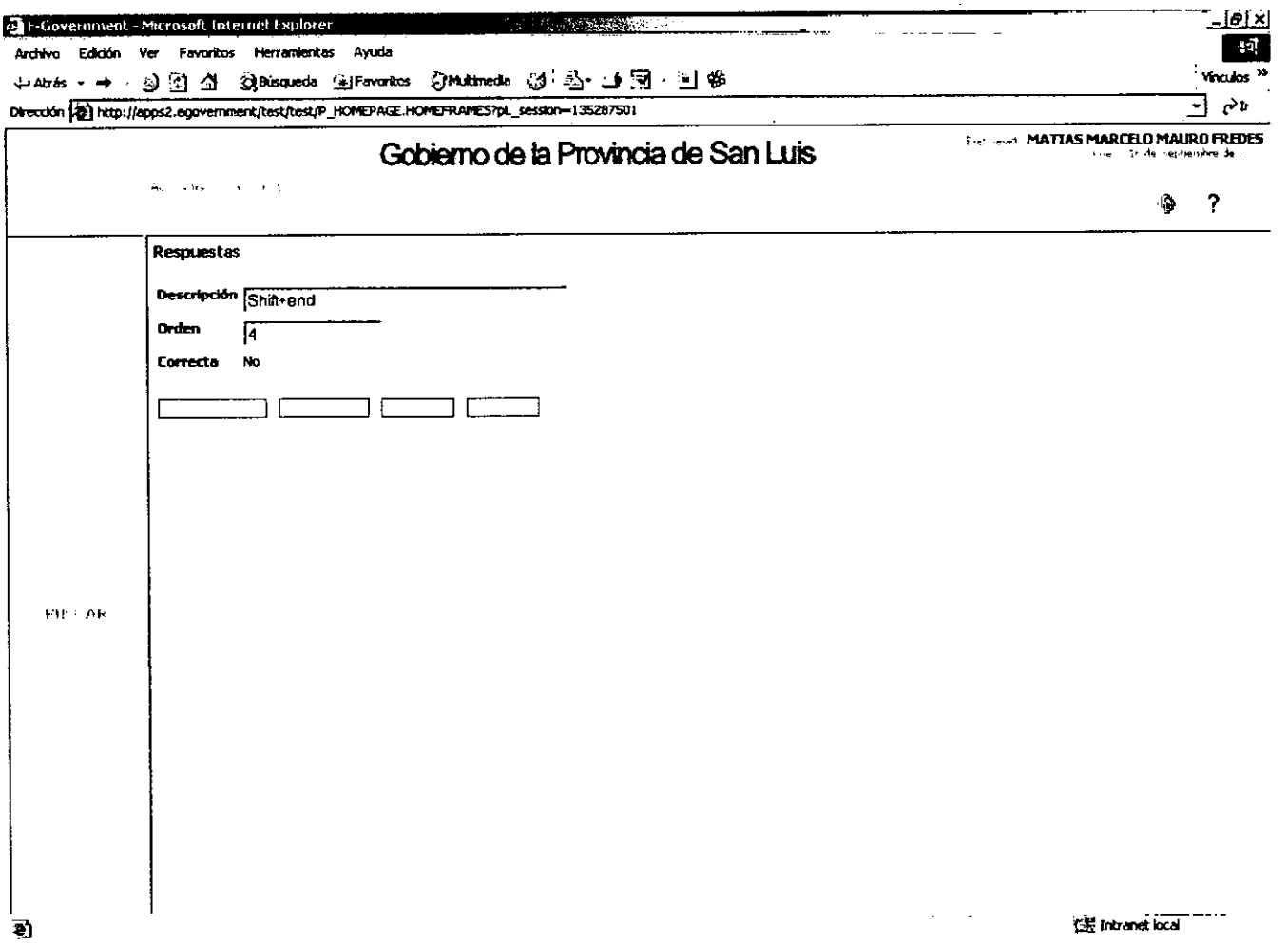
OBSERVACIONES
EN LA CARGA Y / O ACTUALIZACIÓN ADMITE IGUAL RESPUESTAS TANTO PARA "SI" COMO PARA "NO".
IMAGEN



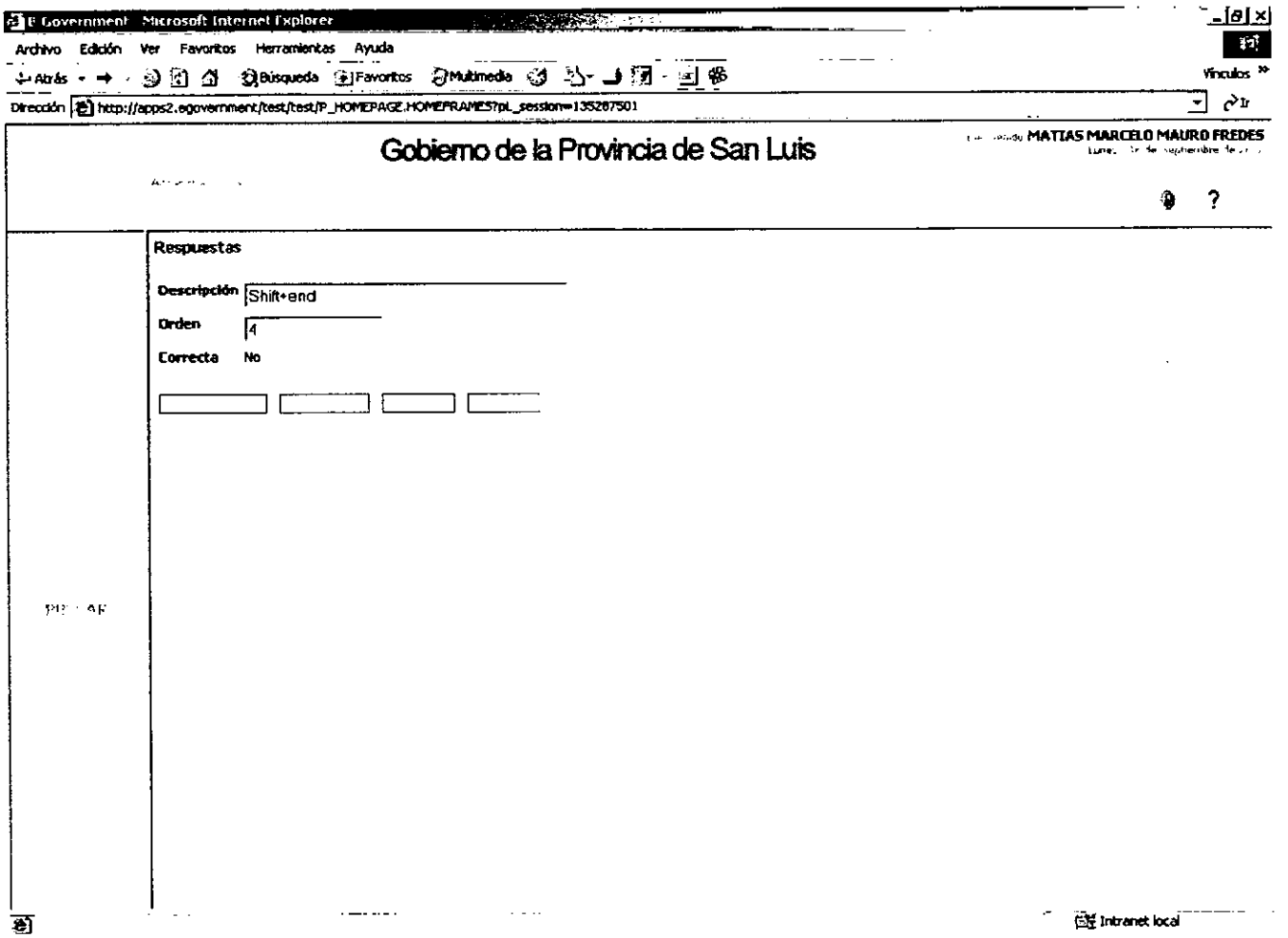
PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar respuesta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

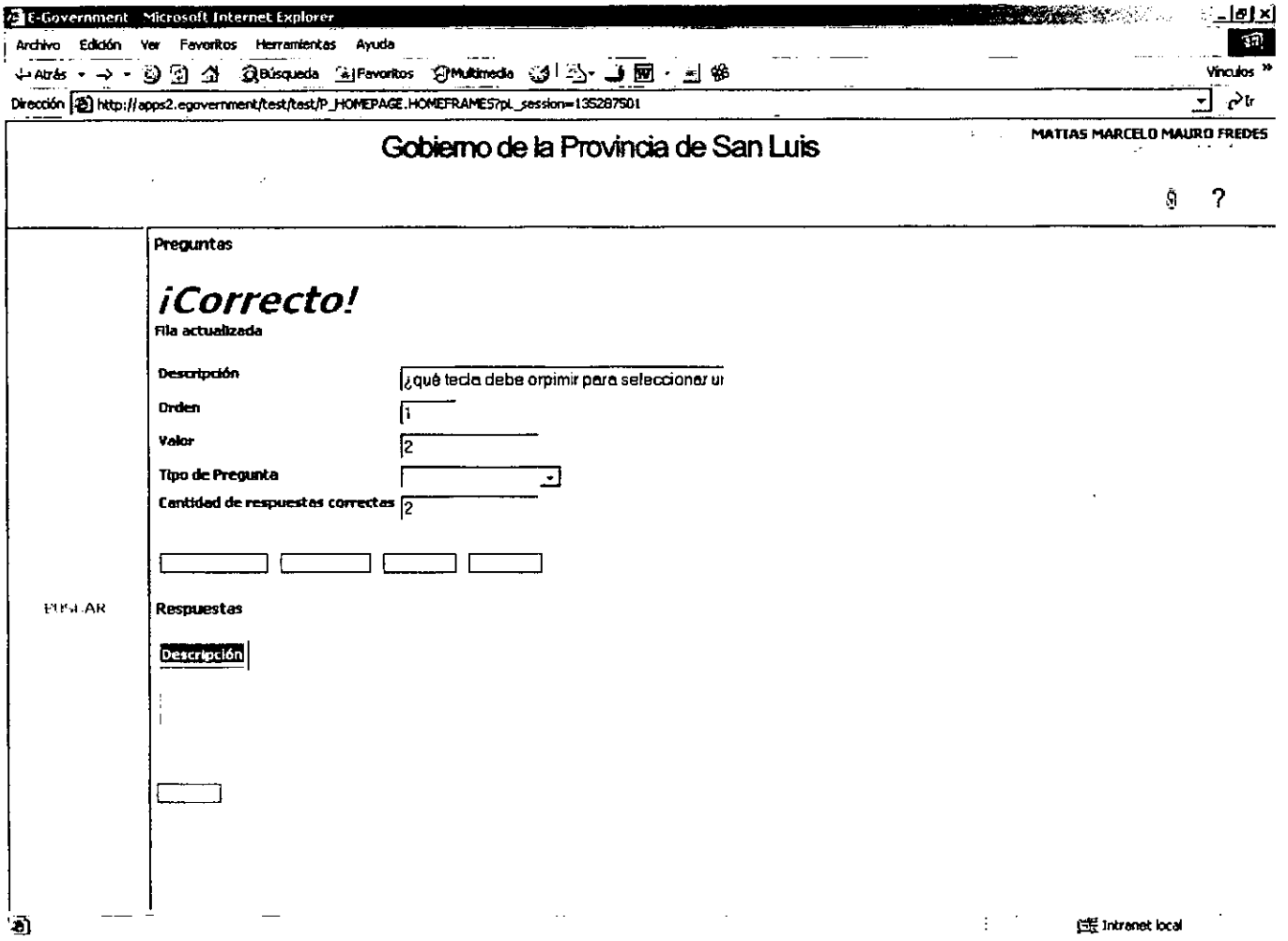
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-028

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

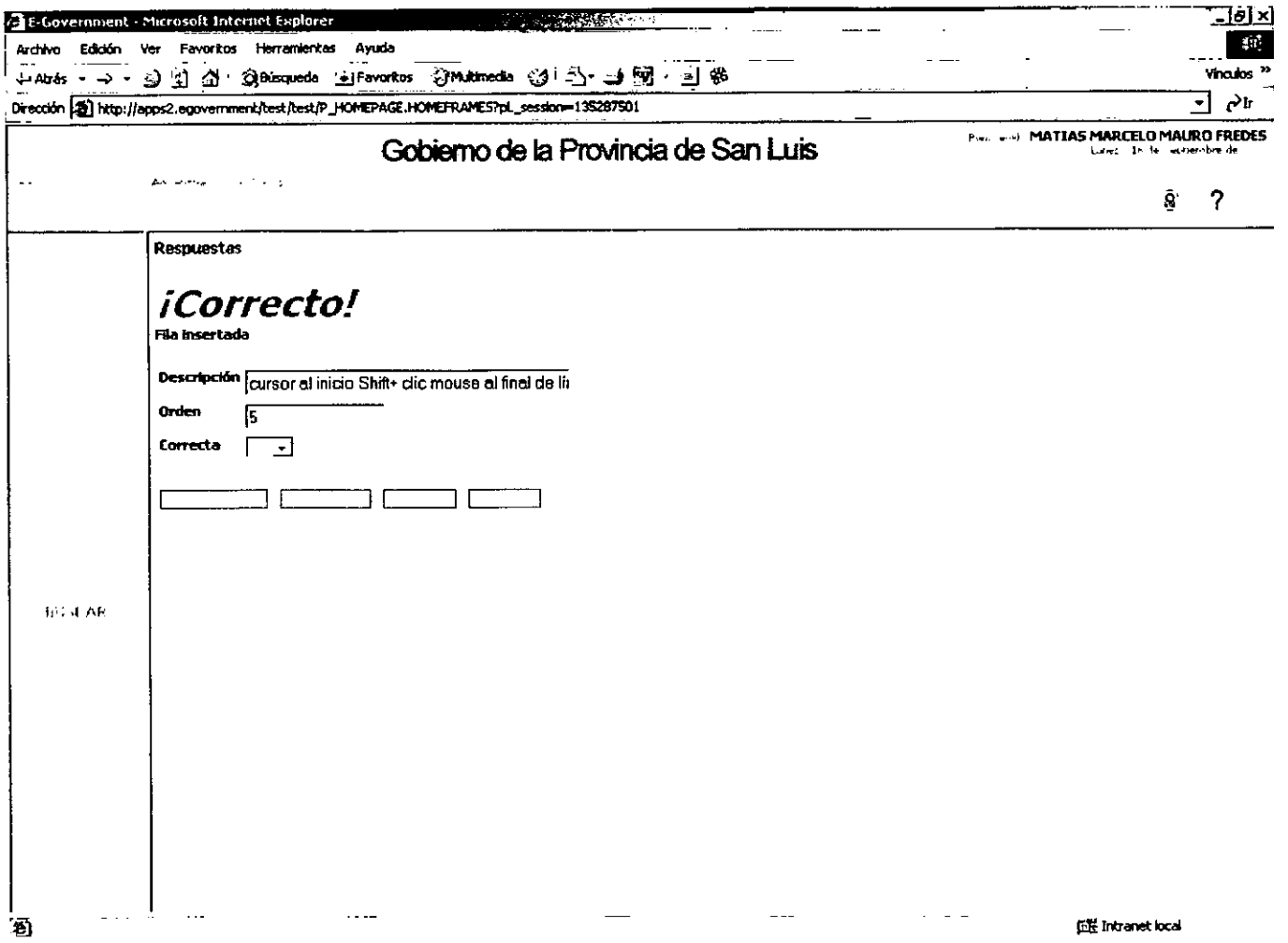
OBSERVACIONES

DESPUÉS DE CARGAR VARIAS RESPUESTAS CORRECTAS PERMITE MODIFICAR LA CANTIDAD DE RESPUESTAS CORRECTAS A UN NÚMERO MENOR AL ESPECIFICADO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAME5?pL_session=135287501

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Capacitación

Preguntas

¡Correcto!

Fila actualizada

Descripción	¿qué tecla debe apimir para seleccionar ur
Orden	1
Valor	2
Tipo de Pregunta	
Cantidad de respuestas correctas	1

Respuestas

Descripción

http://apps2.egovernment/test/test/P_Homepage.Menu?p_page=100158p_path=8pL_session=135287501

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

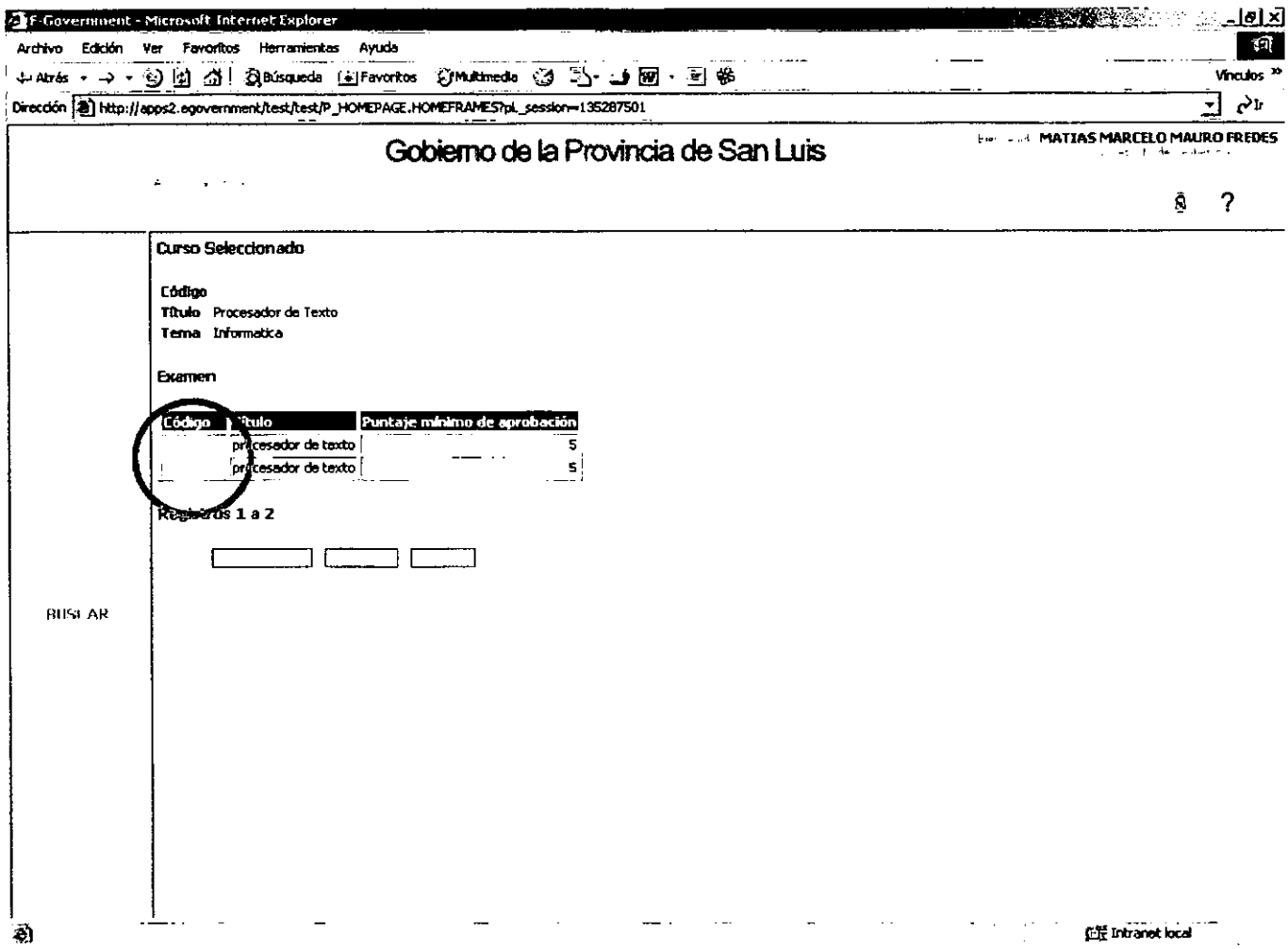
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Buscar Exámen**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-019

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
EL CÓDIGO DE EXÁMEN DEBERÍA SE ÚNICO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

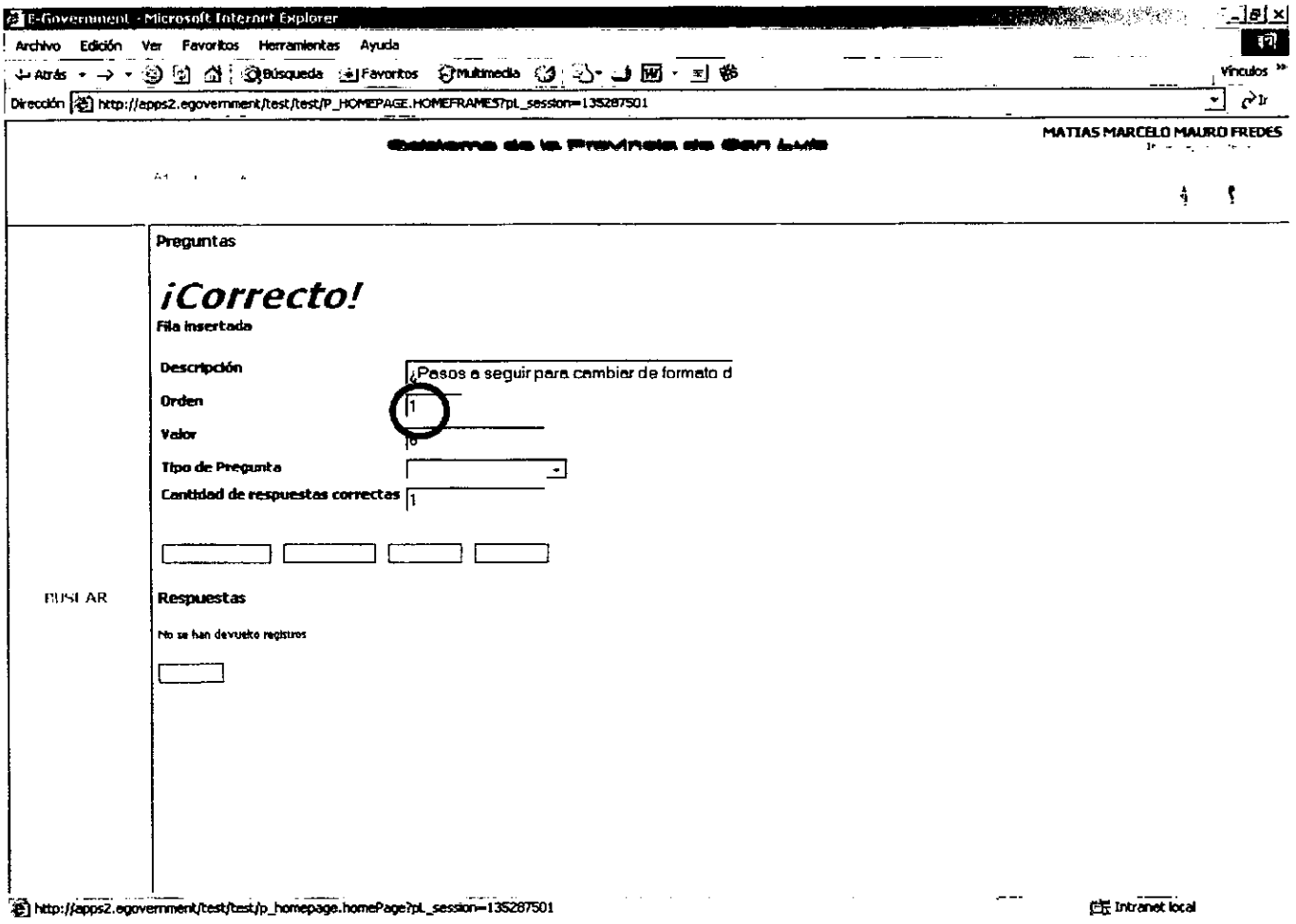
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva respuesta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

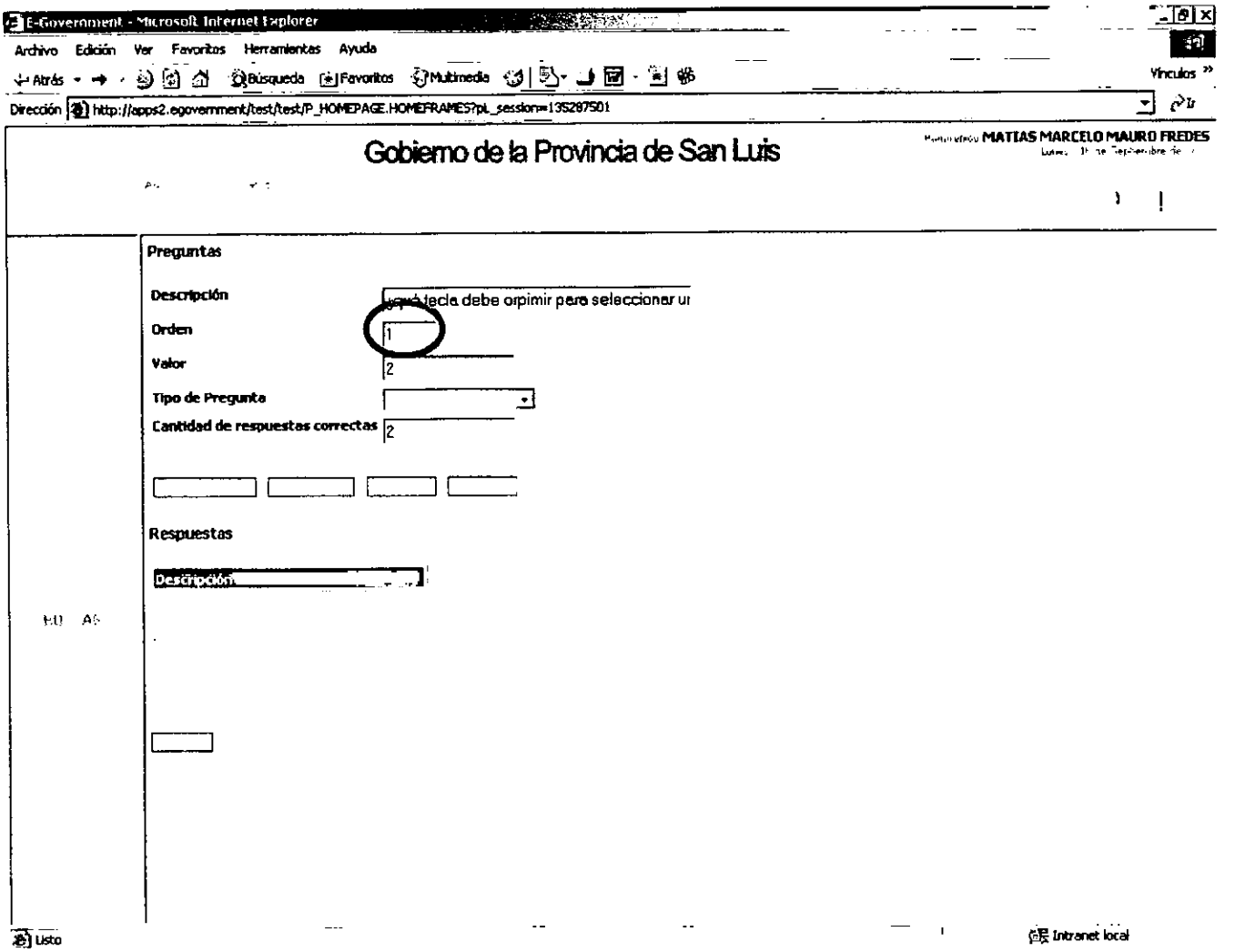
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-027

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
ADMITE DOS PREGUNTAS CON EL MISMO ORDEN.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen Presencial**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nuevo Exámen Presencial**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

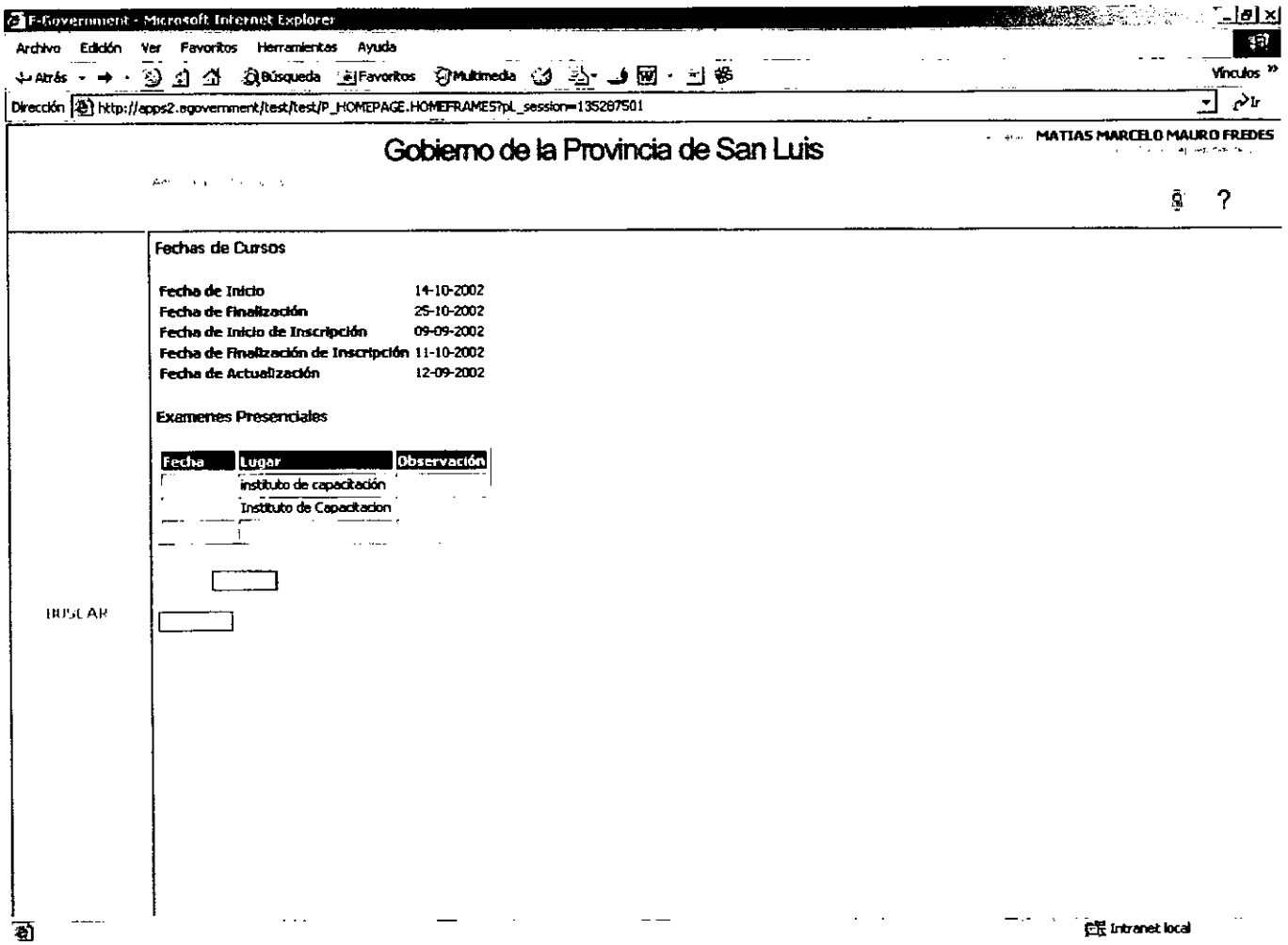
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-031

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

ADMITE CARGAR UNA FECHA DE EXAMEN ANTERIOR AL DICTADO DEL CURSO.
ADMITE CARGAR FECHAS DE EXÁMENES SIN ESPECIFICACIÓN DEL LUGAR.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Pregunta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

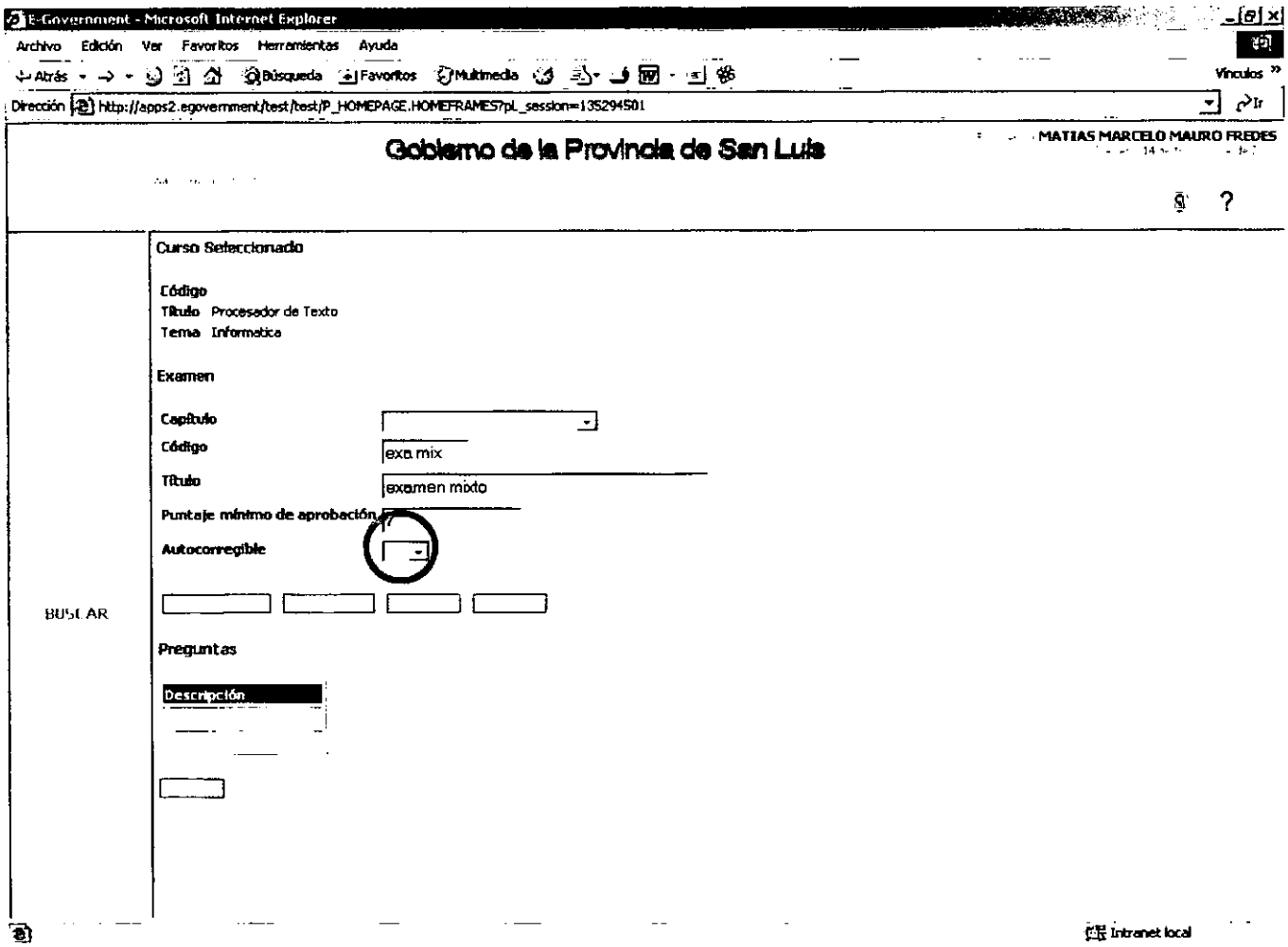
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-024

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

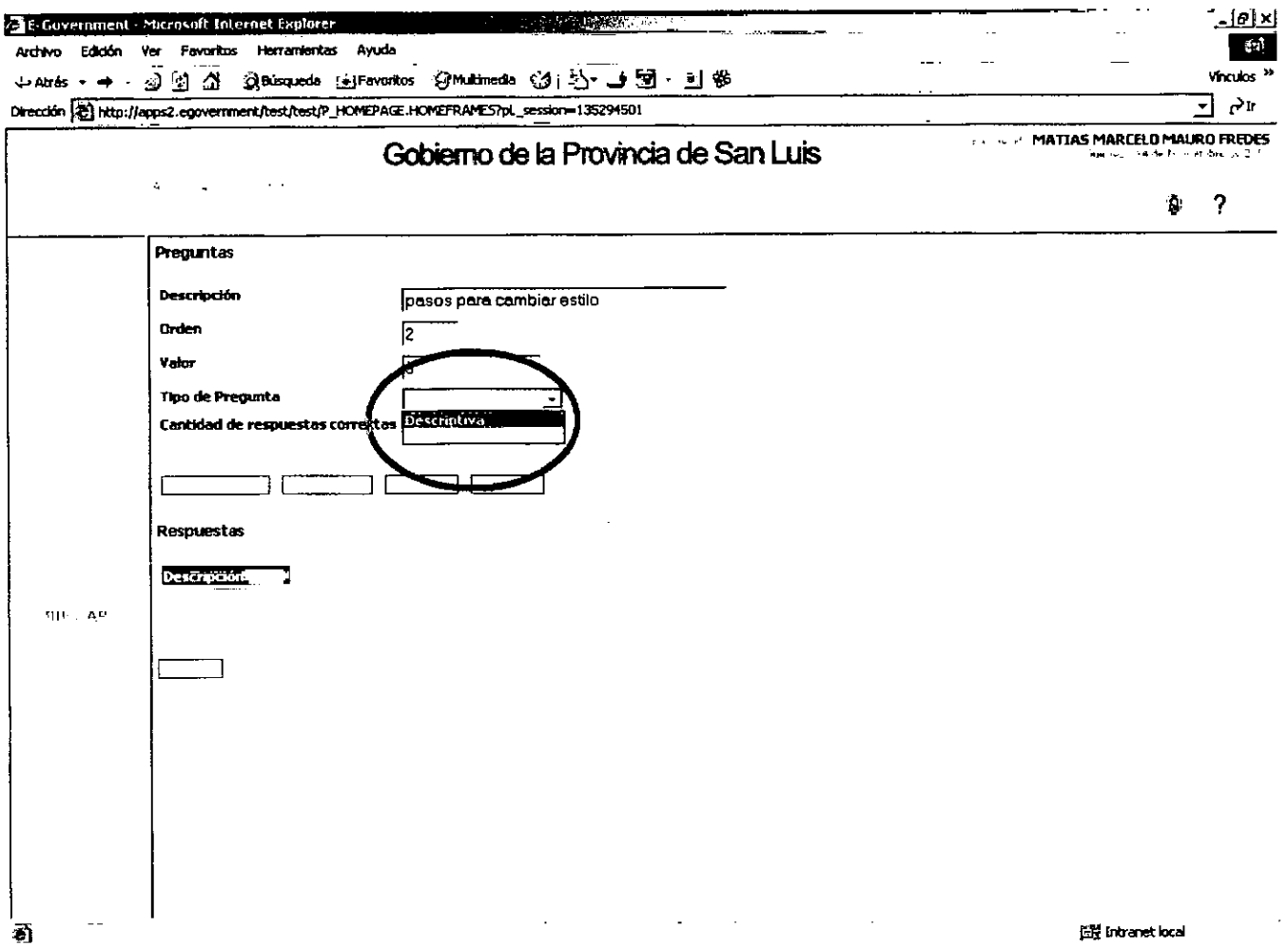
OBSERVACIONES

EN EXÁMENES AUTO-CORREGIBLES NO DEBERÍA PERMITIR NI HABILITAR LA OPCIÓN DE PREGUNTAS DESCRIPTIVAS.
(PARA LOS EXÁMENES MIXTOS DE ENTRADA DEBERÍA ELEGIRSE QUE NO SEAN AUTOCORREGIBLES Y DEBERÍA ESTAR ESPECIFICADO EN EL MANUAL).

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:	FUNCIÓN: **Alumnos por Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

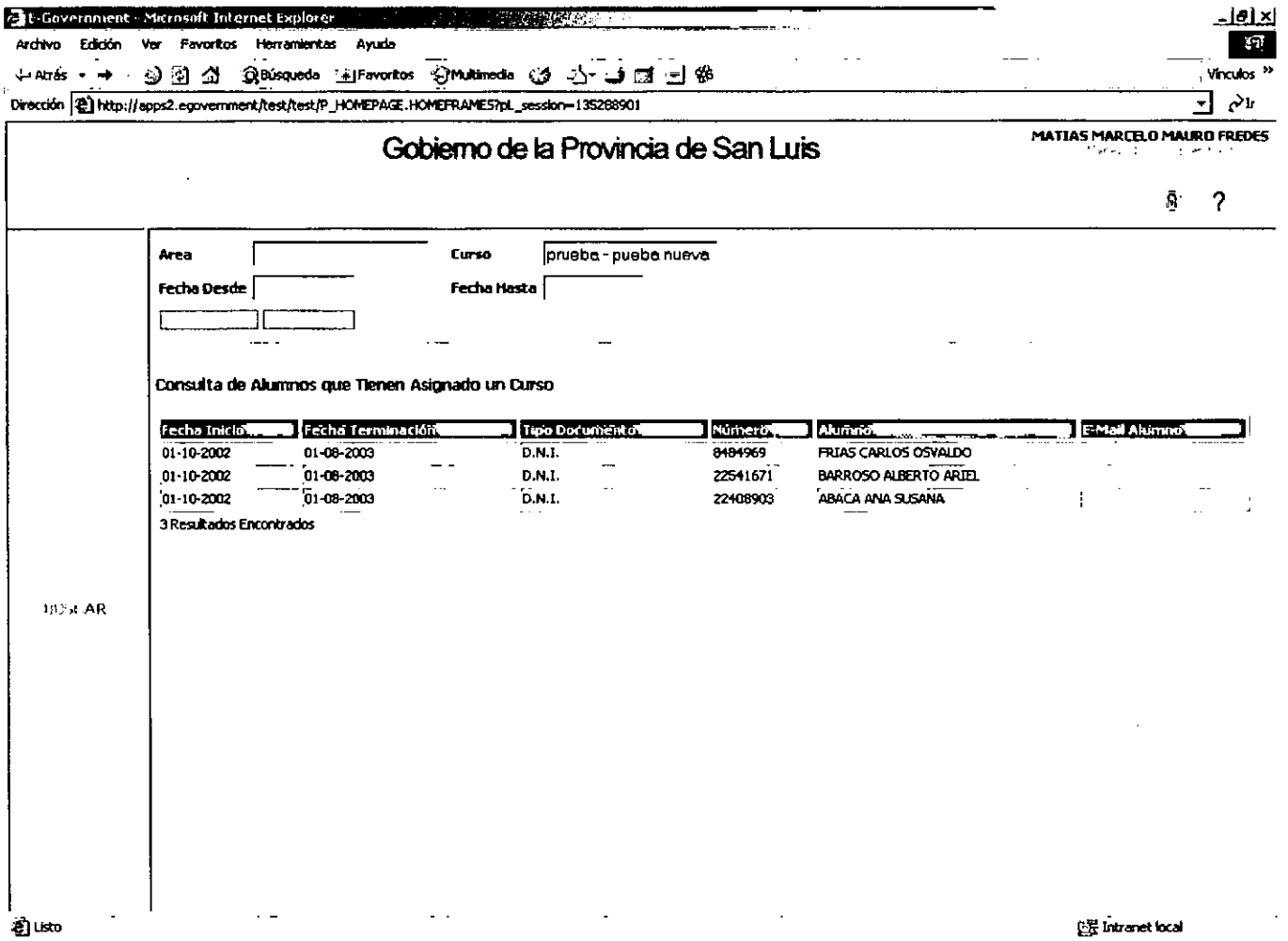
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-064

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

NO DECREMENTA EL NÚMERO DE VACANTES A MEDIDA QUE SE INSCRIBEN LOS ALUMNOS AL CURSO EN EL MÓDULO ADMINISTRACIÓN DE CURSOS / DICTADO.
O LA CONSULTA NO DEVUELVE ESTE DATO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAMES?pL_session=135288901 Ir

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Dictado

Datos del Curso

Código prueba
Título pueba nueva
Tema economia

Fechas del Curso

Fecha de Inicio	Fecha de Finalización	Fecha de Inicio de Inscripción	Fecha de Finalización de Inscripción	Vacantes
	01-09-2003	16-09-2002	30-09-2002	100

BUSCAR

Listo Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

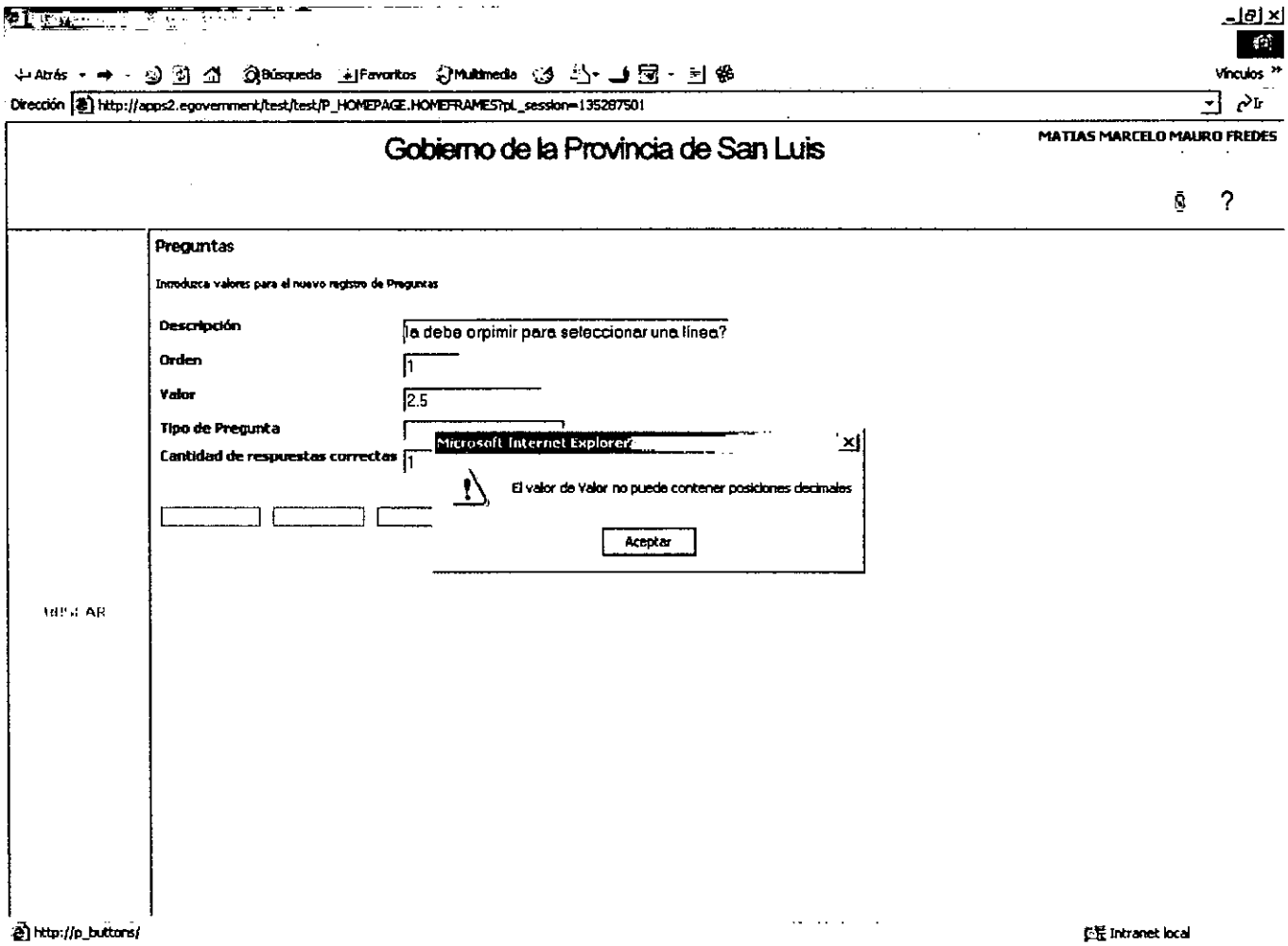
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Pregunta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

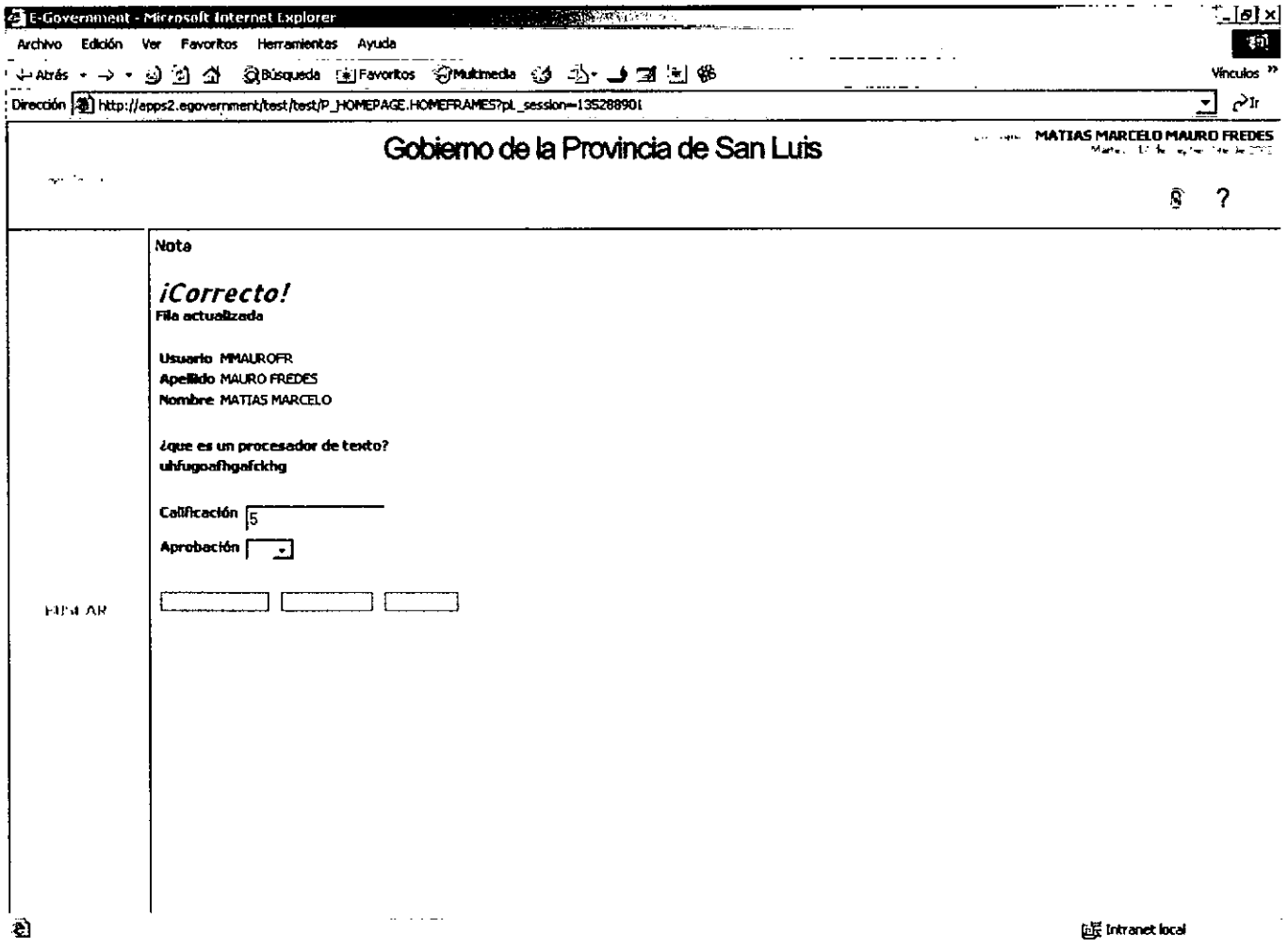
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-024

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

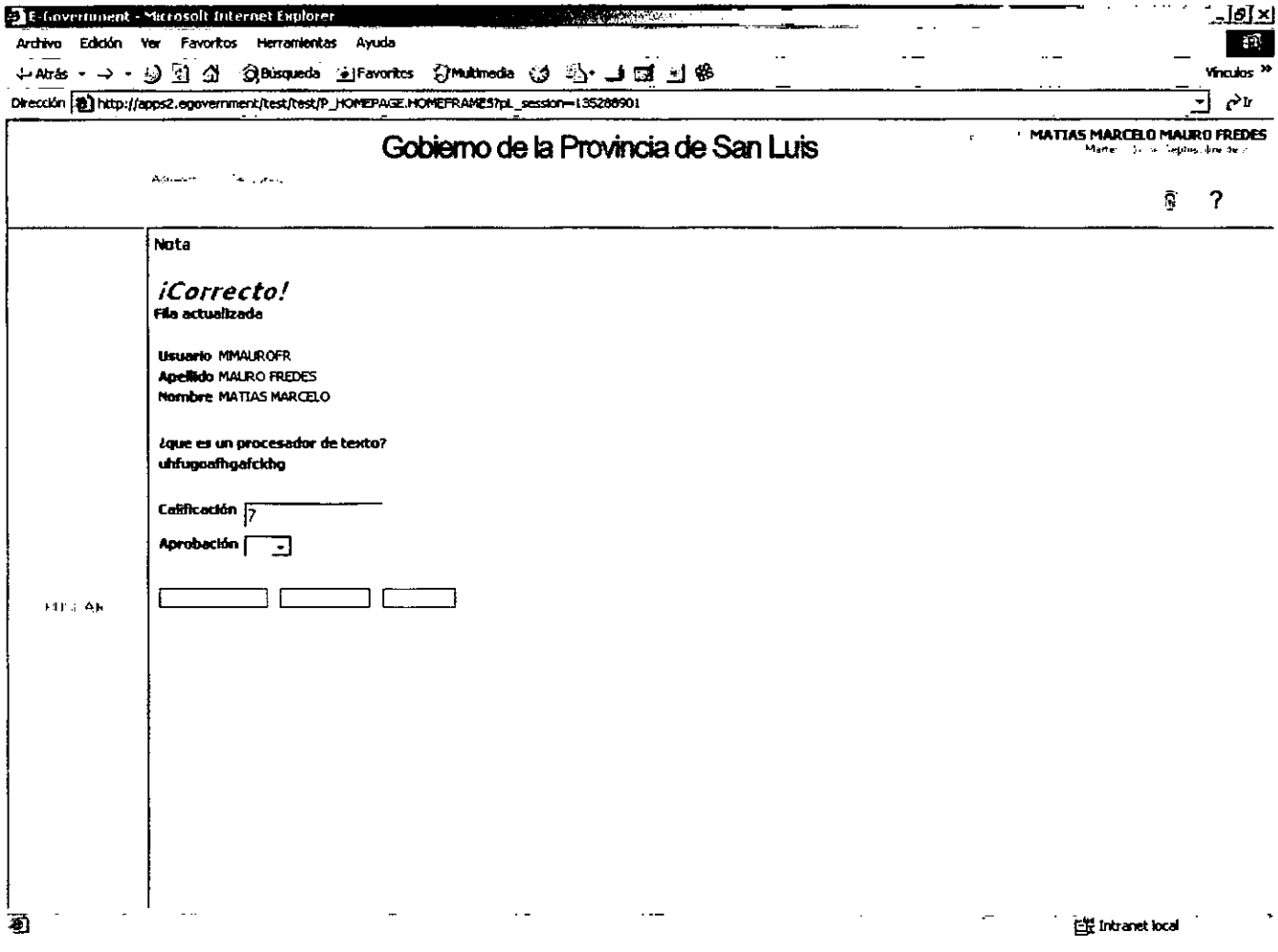
OBSERVACIONES
NO ADMITE VALORES DECIMALES PARA EL VALOR DE LA PREGUNTA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

E-Government - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Dirección http://apps2.egovernment/test/test/P_HOMEPAGE.HOMEFRAME5?pL_session=135288901

Gobierno de la Provincia de San Luis

MATIAS MARCELO MAURO FREDES

Curso Seleccionado

Código
Título Procesador de Texto
Tema Informatica

Examen

Código	Título	Puntaje mínimo de aprobación
	procesador de texto	5
	examen descriptivo	5
	examen mixto	7

Registros 1 a 3

Intranet local

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

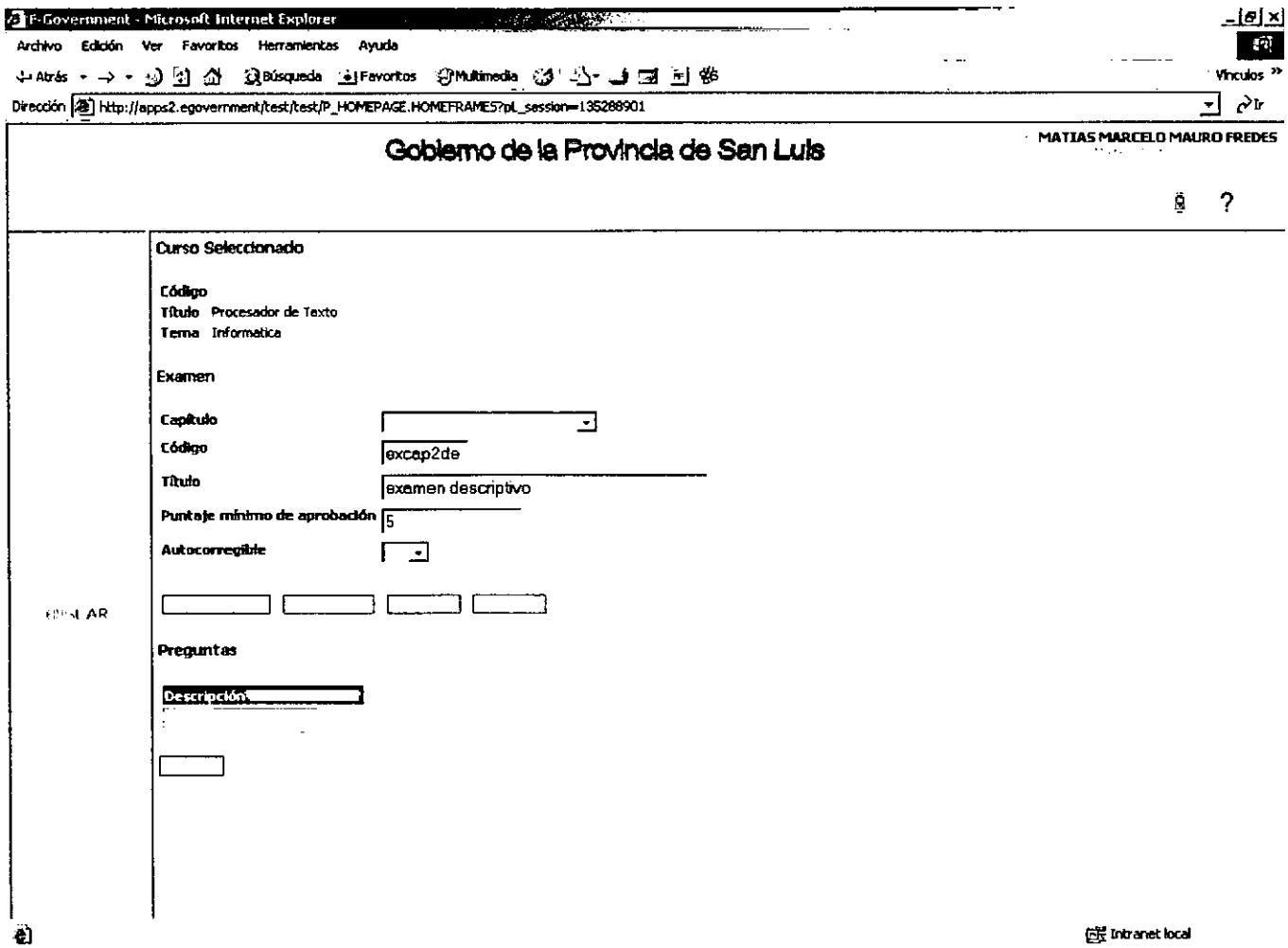
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Actualizar o Borrar Exámen**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

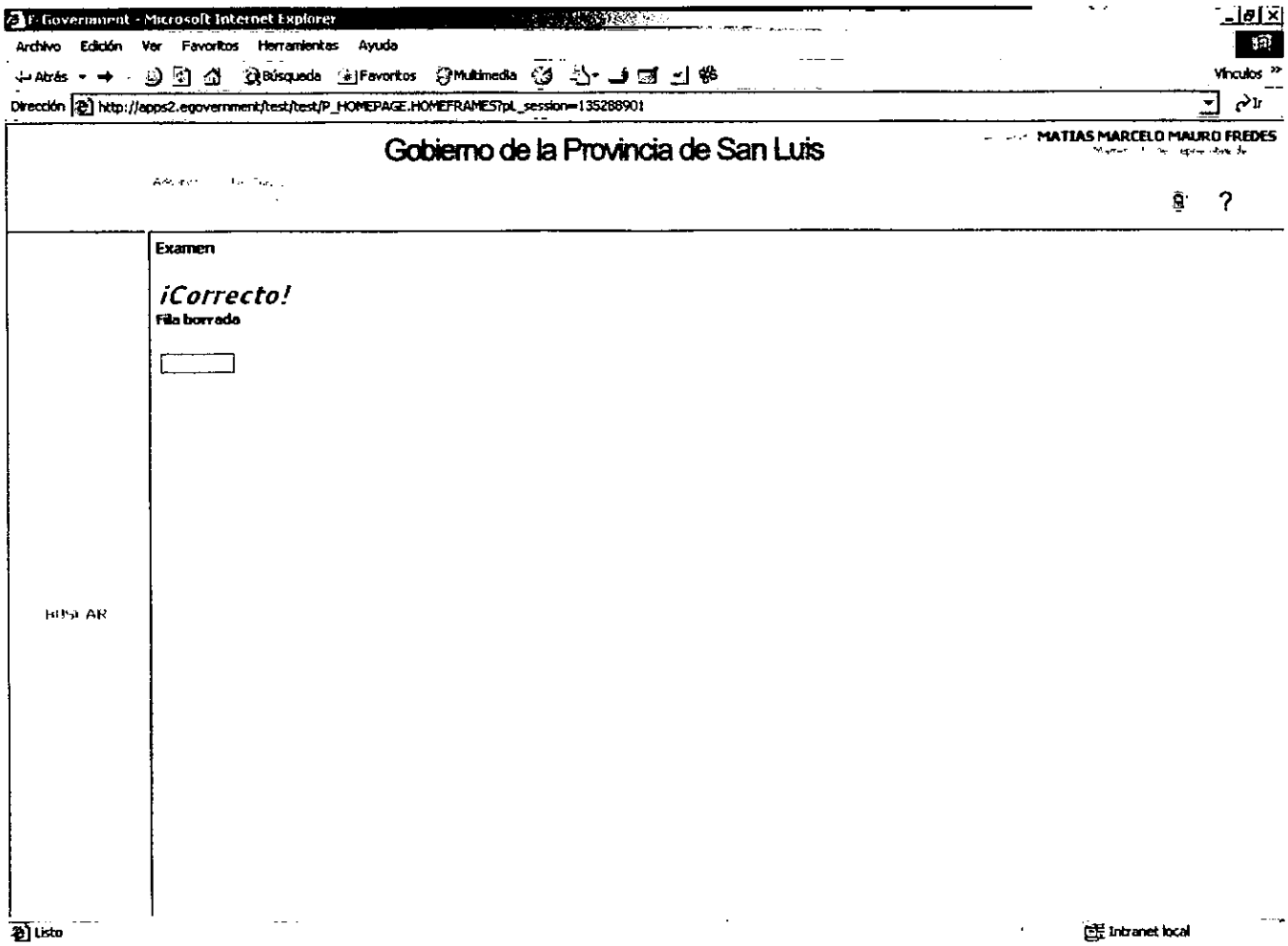
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-062

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
ADMITE BORRAR EXÁMENES QUE YA HAN SIDO RENDIDOS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

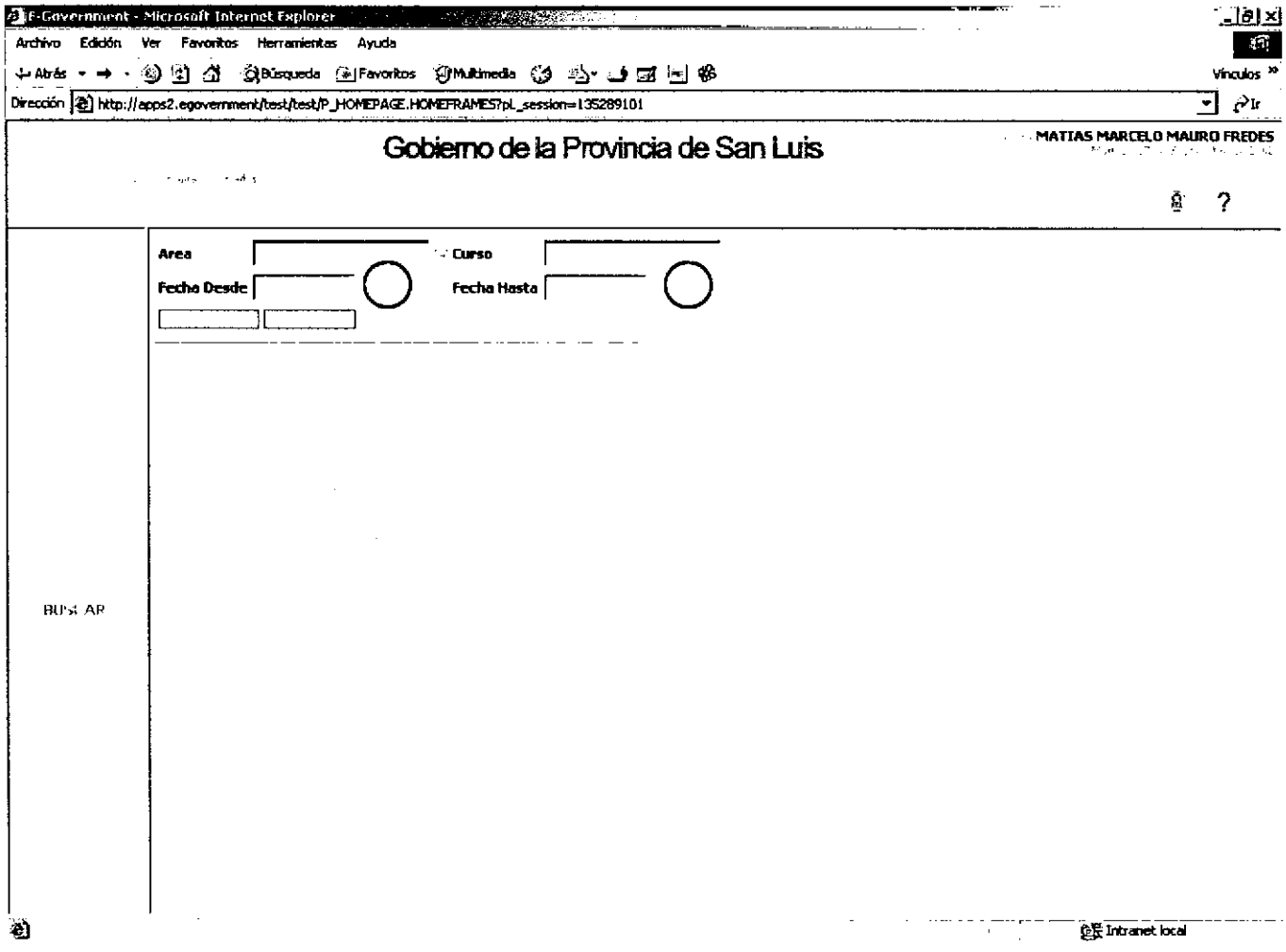
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Listado de Usuarios**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-062

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE USUARIOS NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS**
MÓDULO: **ABM Cursos**	FUNCIÓN: **Actualizar o Borrar Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

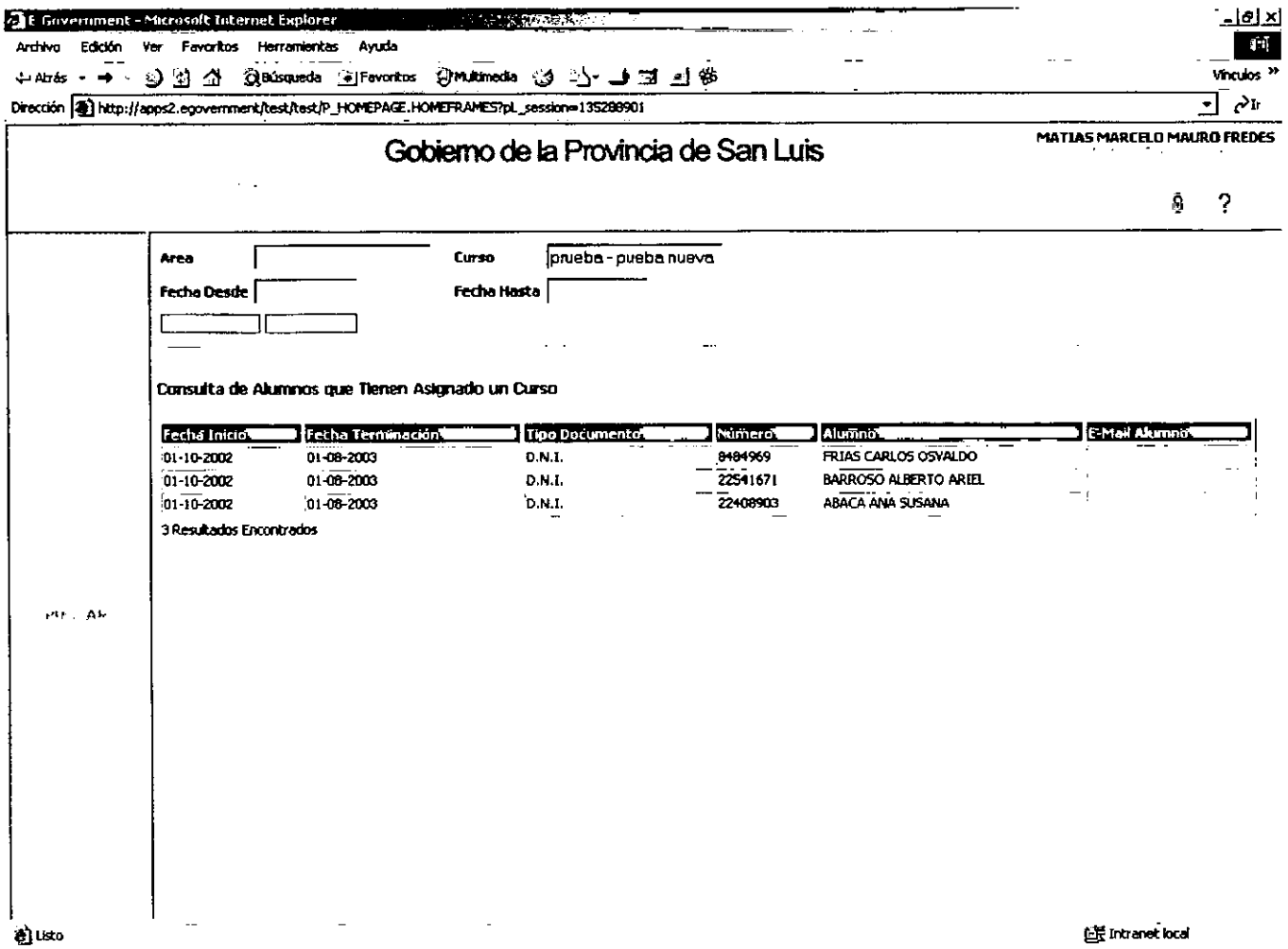
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-005

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

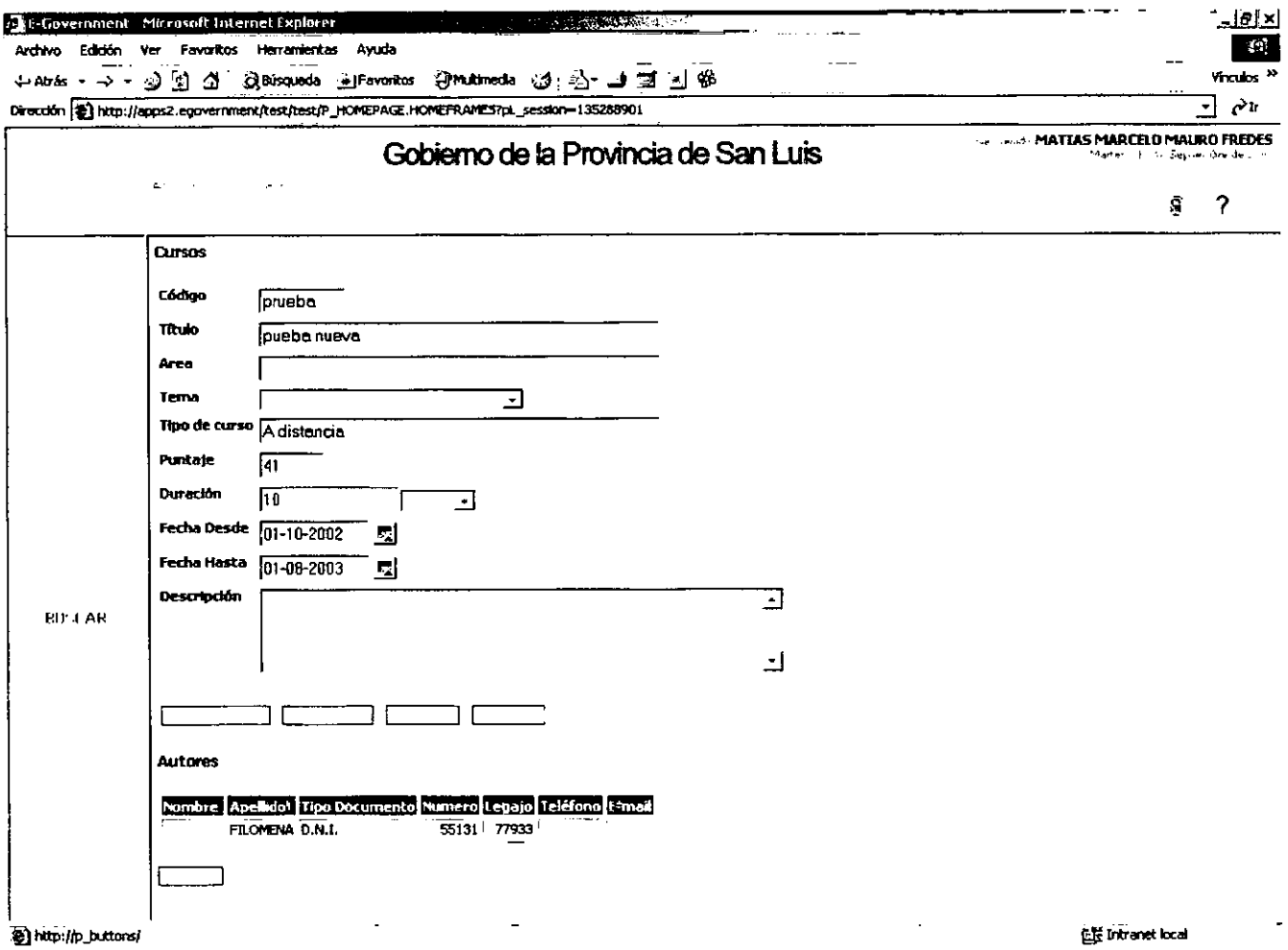
OBSERVACIONES

PERMITE BORRAR CURSOS COMPLETOS CON ALUMNOS CURSANDO Y CON EXÁMENES RENDIDOS PERDIÉNDOSE LA INFORMACIÓN (SIN CARTELES DE ADVERTENCIA).
NO SE OBSERVAN RESTRICCIONES DE INTEGRIDAD.

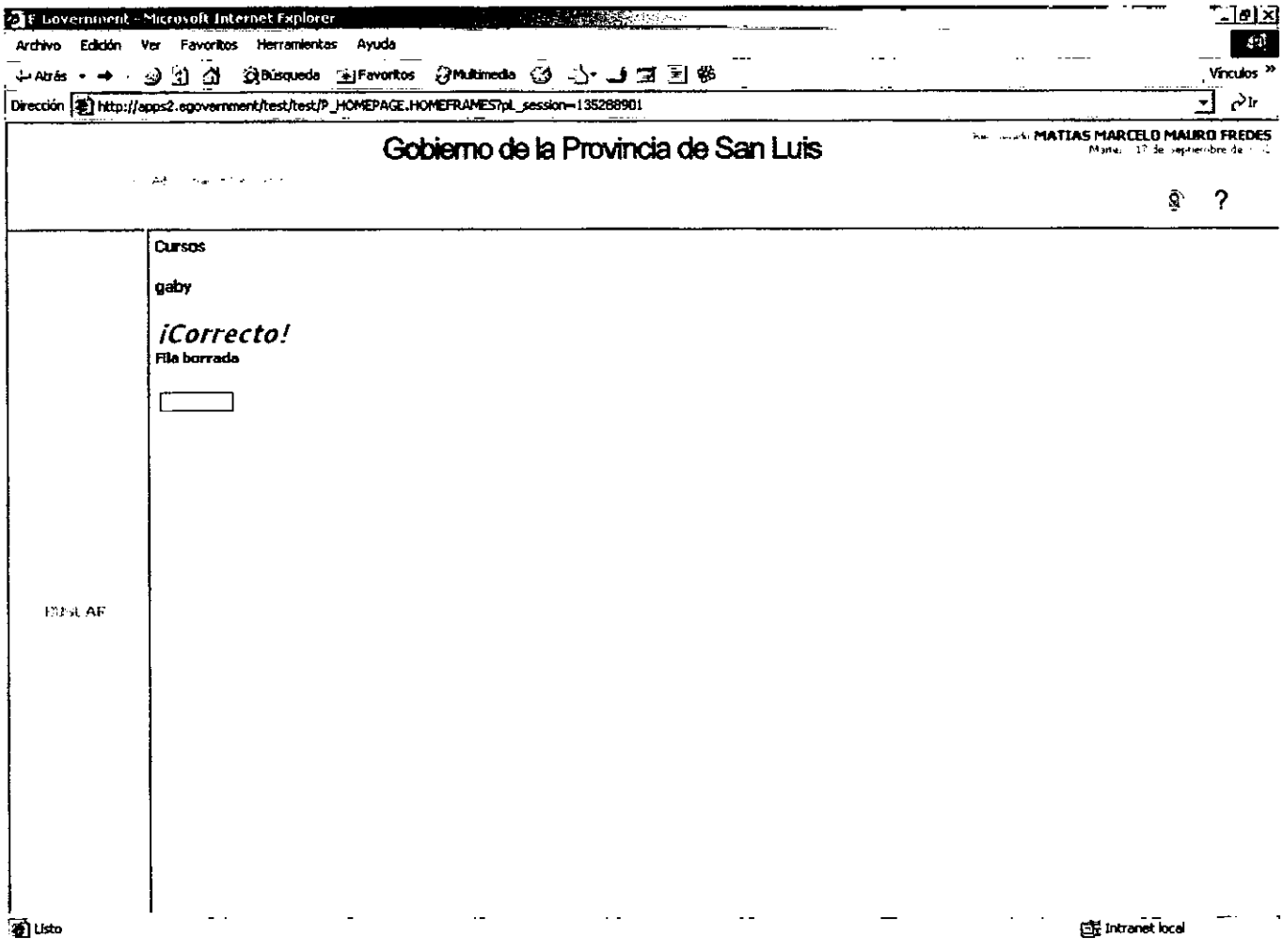
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

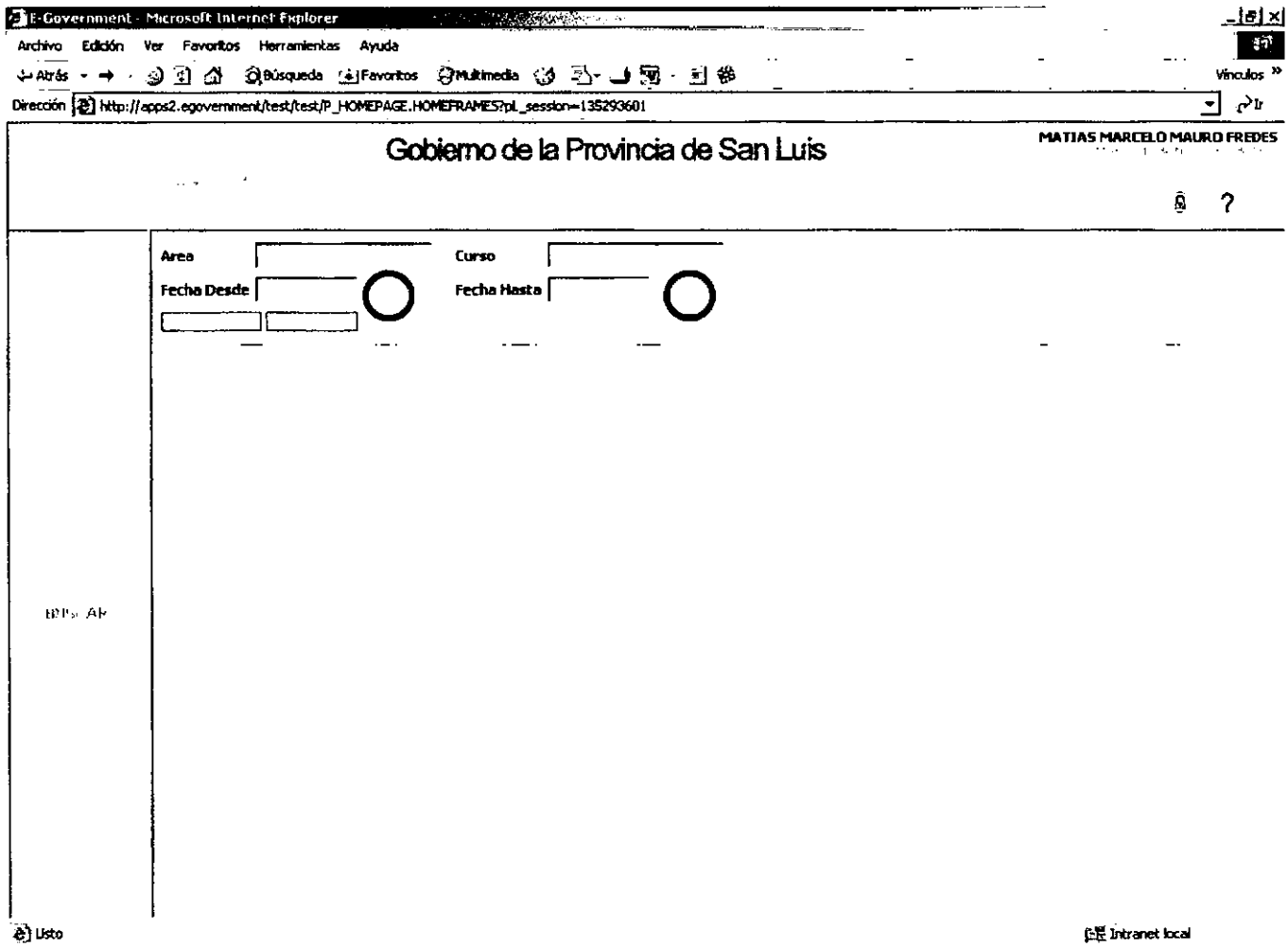
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Alumnos por Curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-064

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE ALUMNOS POR CURSO NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

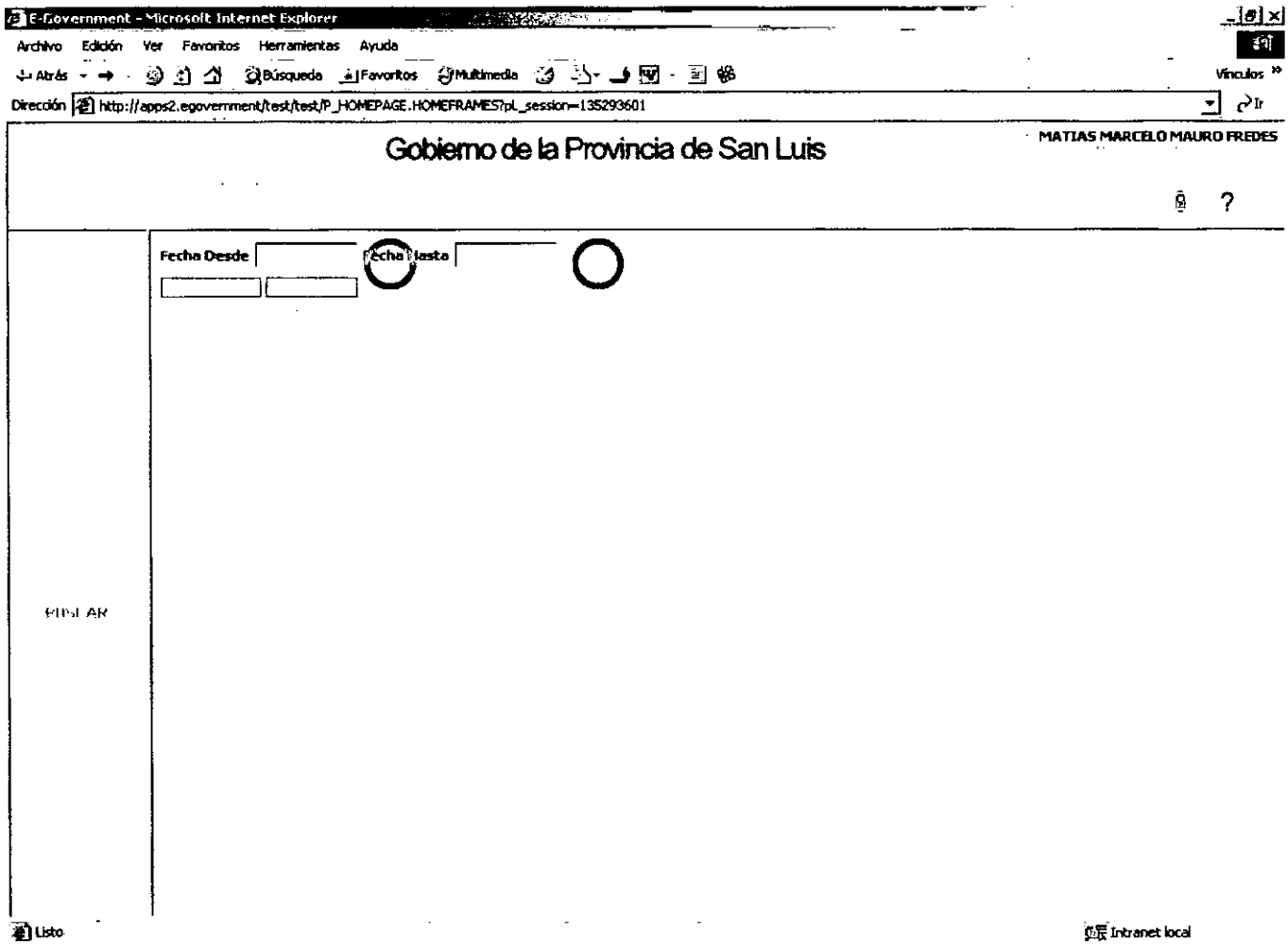
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Cursos por Alumno**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-058

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE CURSOS POR ALUMNO NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

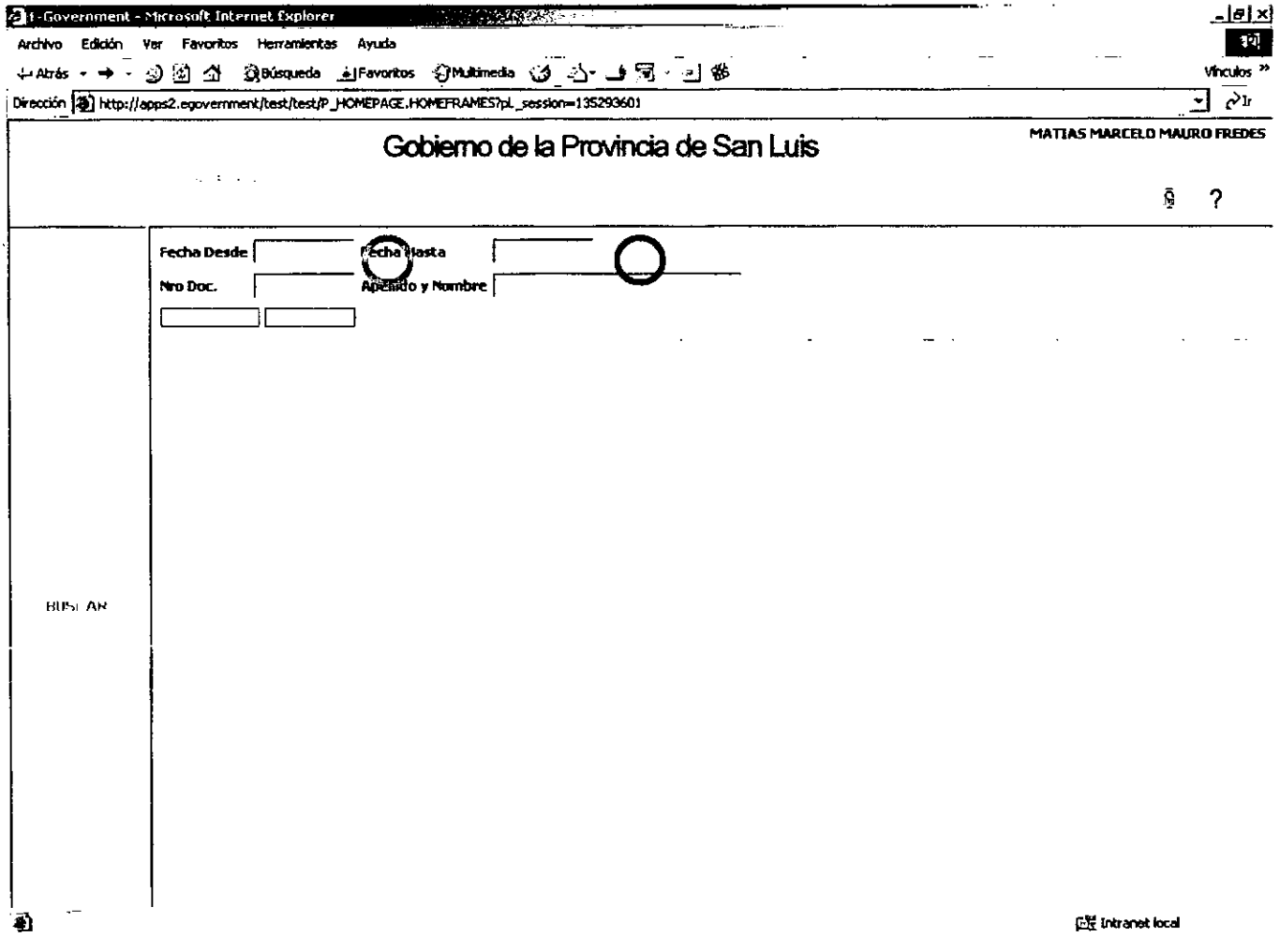
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:	FUNCIÓN: **Cursos que Realiza un Alumno**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-060

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE CURSOS QUE REALIZA UN ALUMNO NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

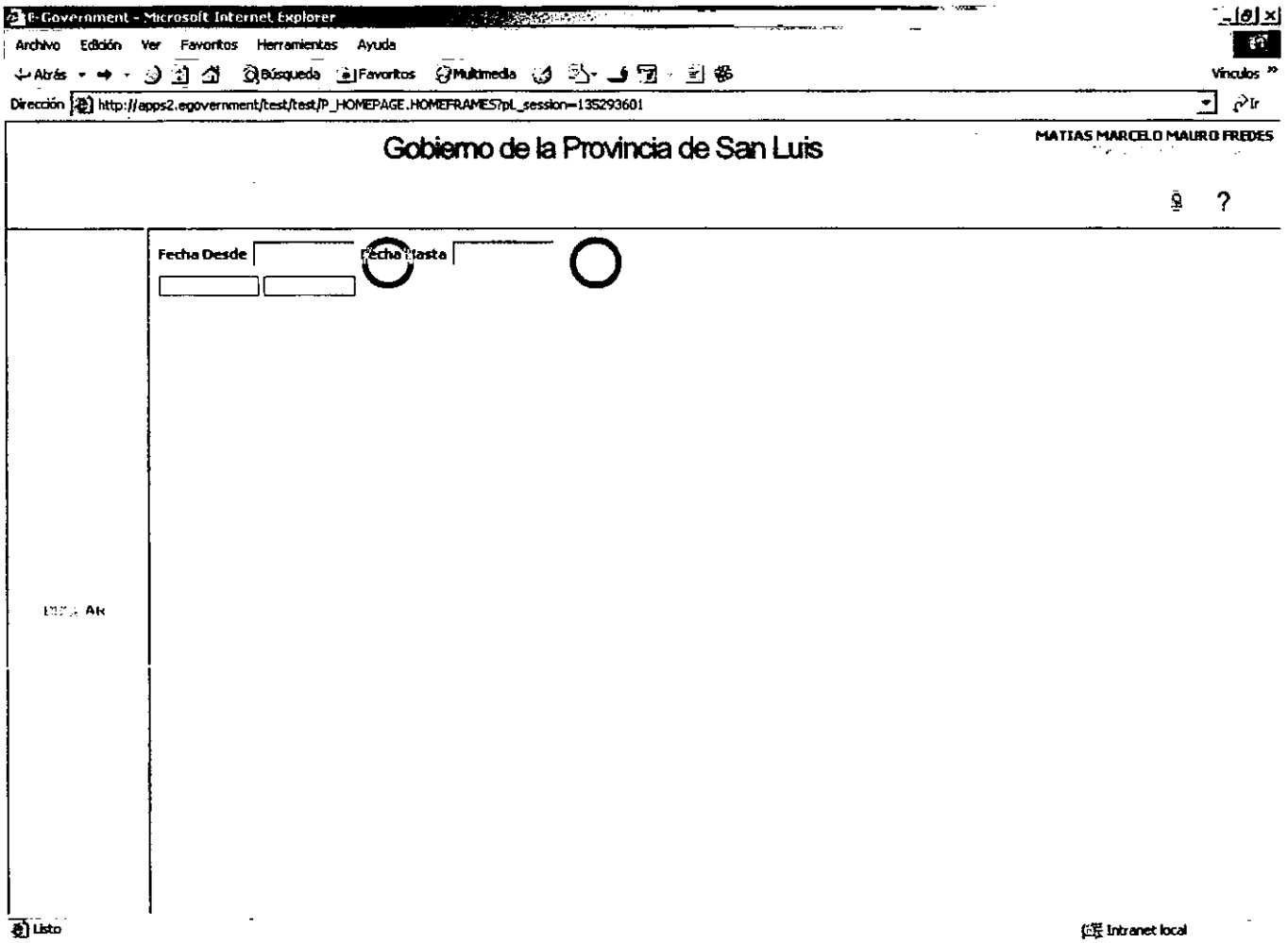
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Cursos por Tutor**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-063

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE CURSOS POR TUTOR NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

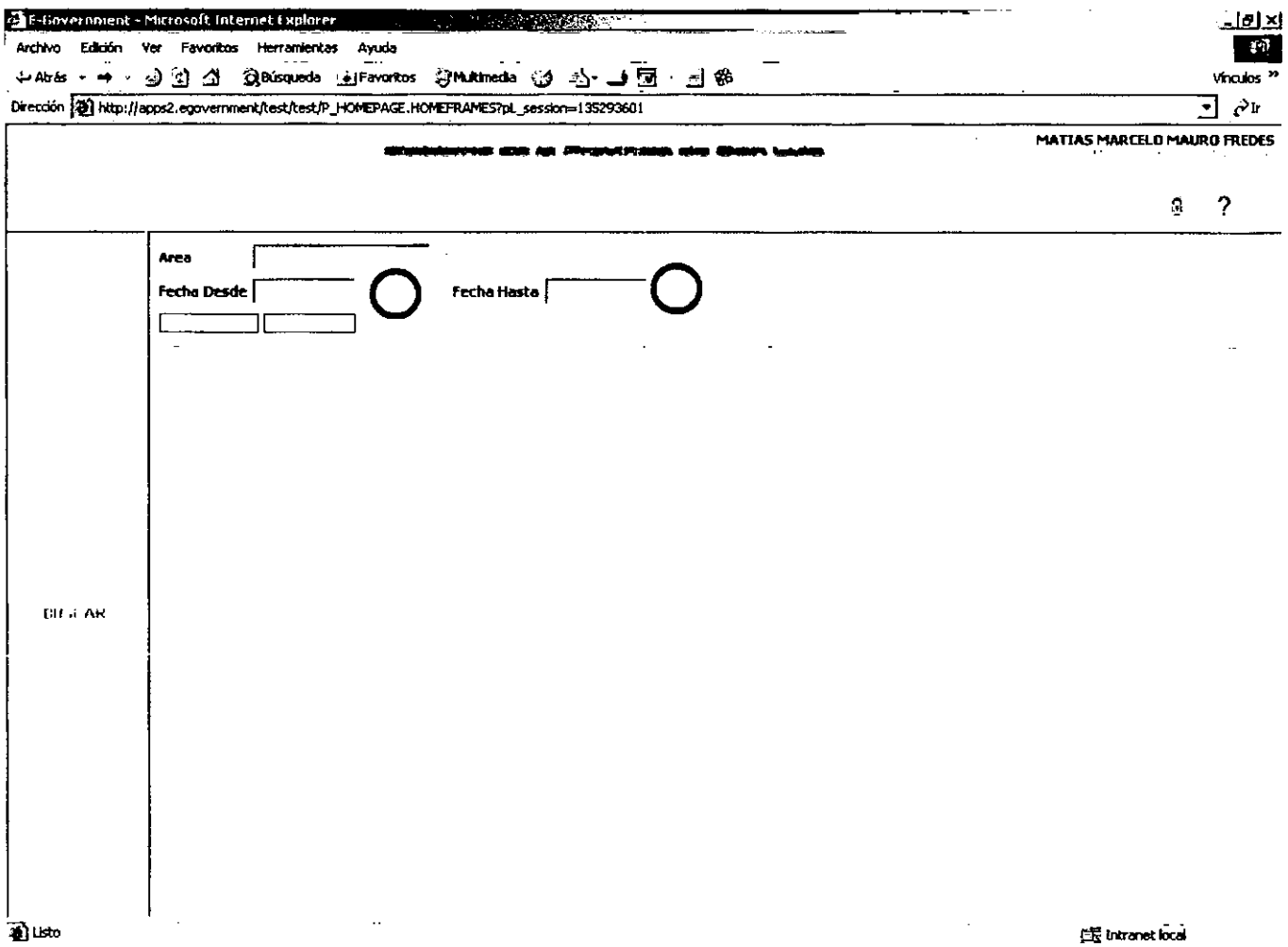
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:	FUNCIÓN: **Estadísticas**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-066

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
EN LA PANTALLA DE LA CONSULTA DE LISTADO DE ESTADÍSTICAS NO SE OBSERVAN LOS ÍCONOS DE CALENDARIO PARA LOS CAMPOS DE FECHA.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

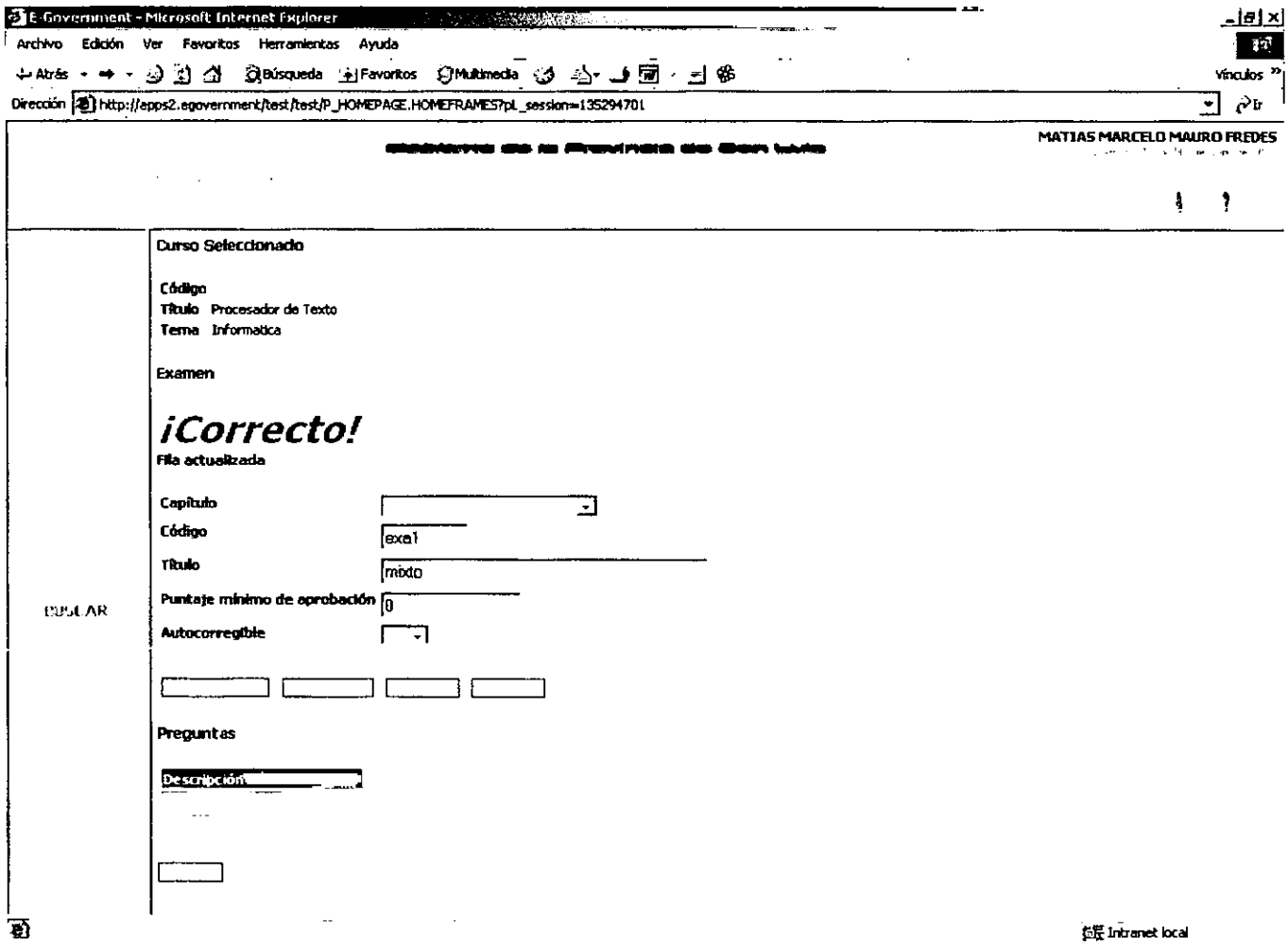
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS**
MÓDULO: **Exámen**	FUNCIÓN: **Nuevo Exámen – Actualizar o Borrar Exámen**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-021 **CP-EG-SC-TA-022**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
PERMITE CARAGAR O ACTUALIZAR EL PUNTAJE MÍNIMO DE APROBACIÓN CON "0".
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

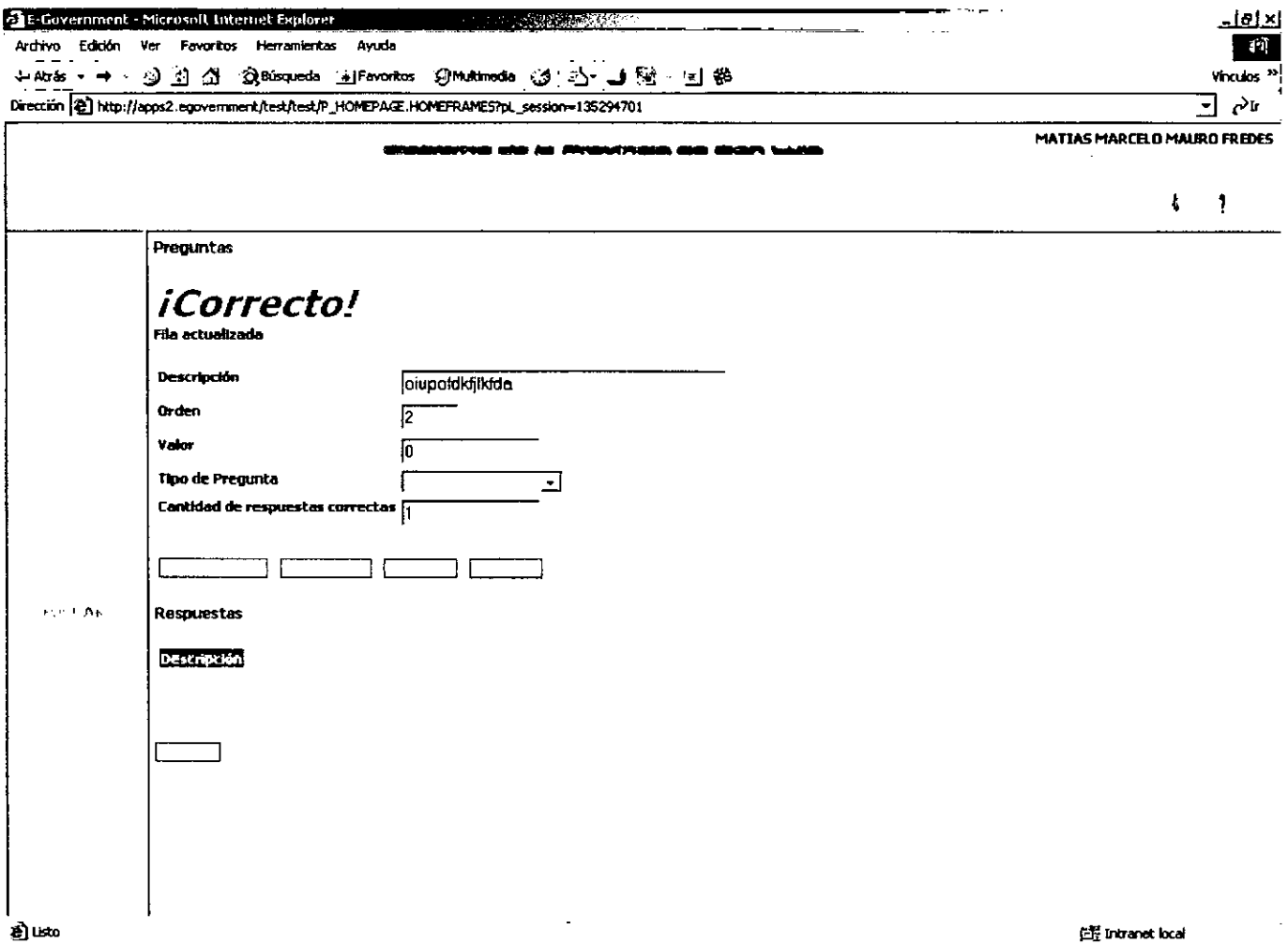
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Exámen**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Nueva Pregunta – Actualizar o Borrar Pregunta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-024 **CP-EG-SC-TA-025**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
PERMITE CARGAR O ACTUALIZAR EL VALOR DE UNA PREGUNTA CON "0".
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS**
MÓDULO: **Exámen**	FUNCIÓN: **Actualizar o Borrar Pregunta**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

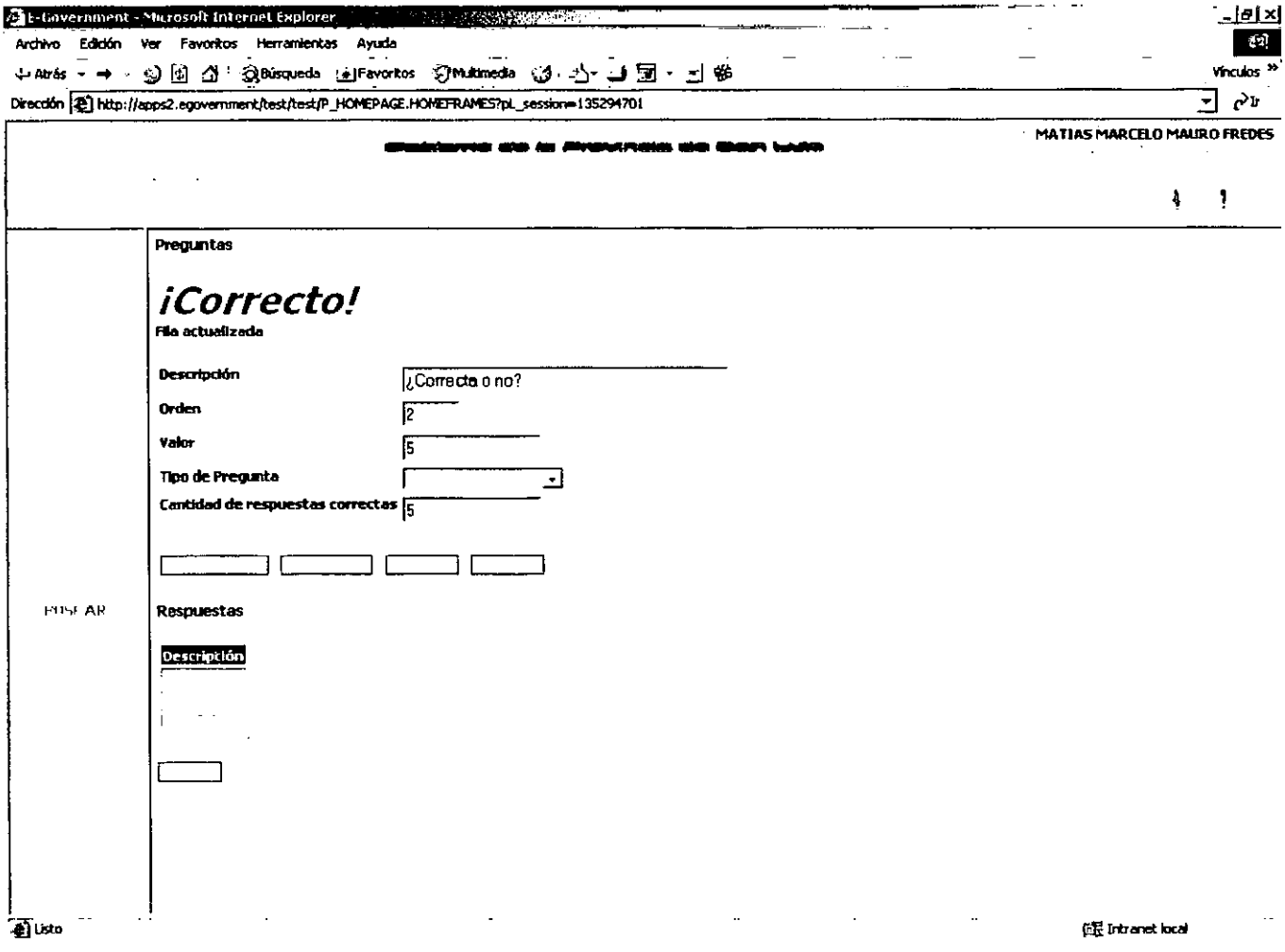
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-025

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

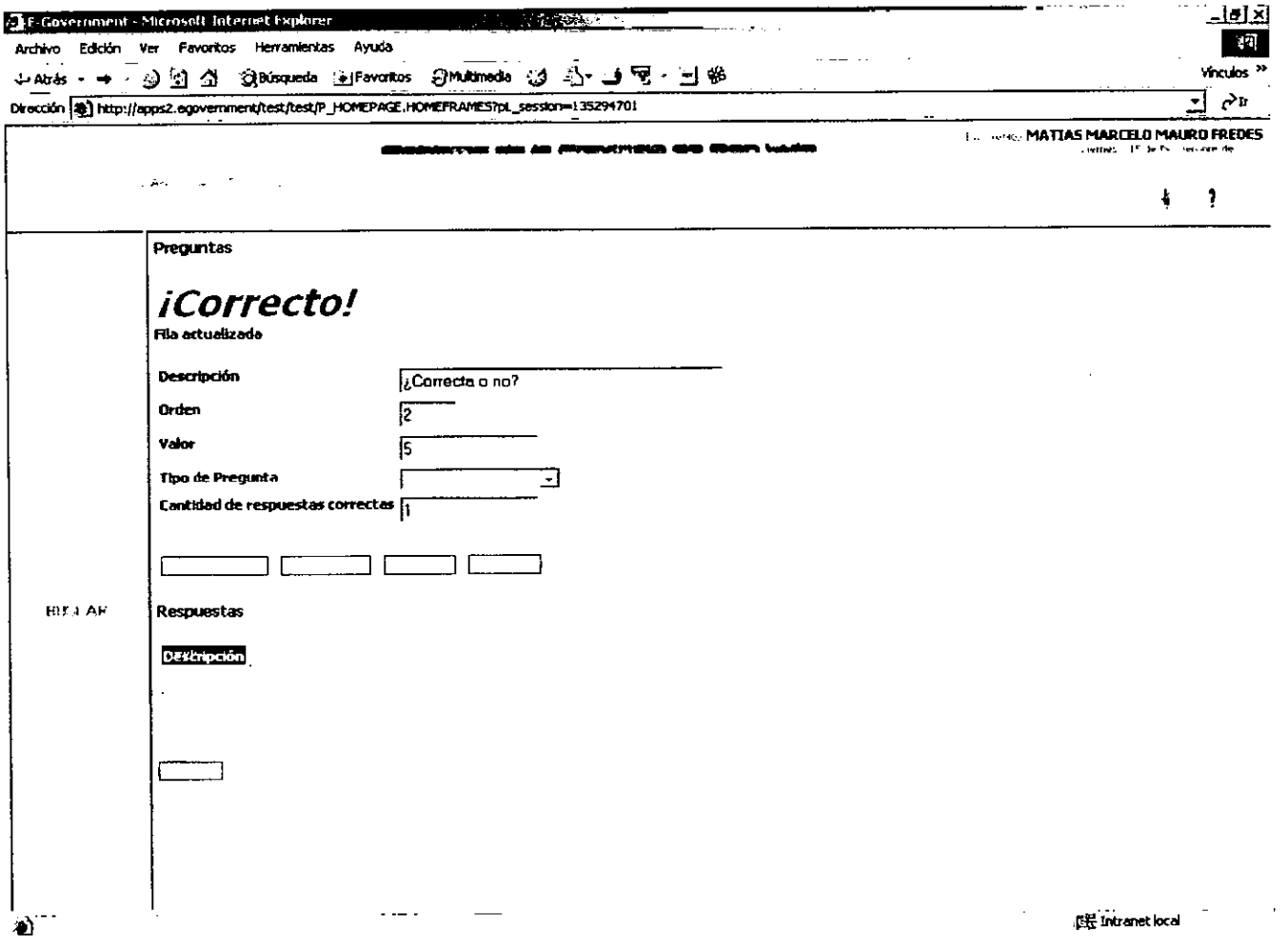
OBSERVACIONES

PERMITE ACTUALIZAR O CARGAR UN NÚMERO MAYOR QUE EL DE RESPUESTAS CORRECTAS EXISTENTES.
PERMITE ACTUALIZARA A UN NÚMERO MENOR AL DE RESPUESTAS CORRECTAS YA CARGADAS.
ESTE NÚMERO DEBERÍA DEPENDER DE LA CANTIDAD DE REGISTROS CORRECTOS CARGADOS.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

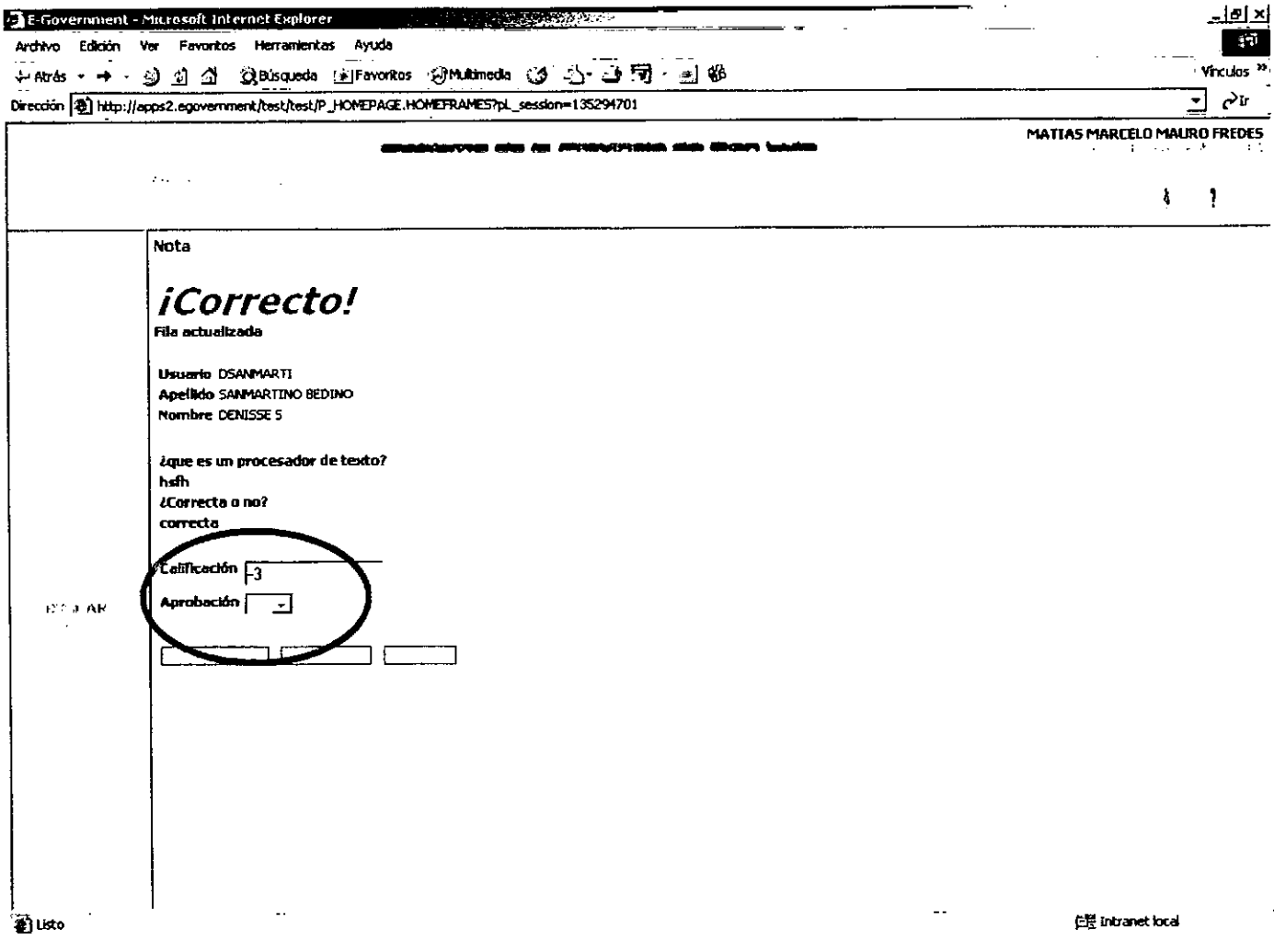
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **Notas**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Cargar o Actualizar Nota**

PLAN DE PRUEBAS ASOCIADO	
ID: **001**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-035

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES
PERMITE CARGAR Y ACTUALIZAR APROBACIONES DE CURSO CON CALIFICACIONES NEGATIVAS.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Cursos que realiza un Alumno**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

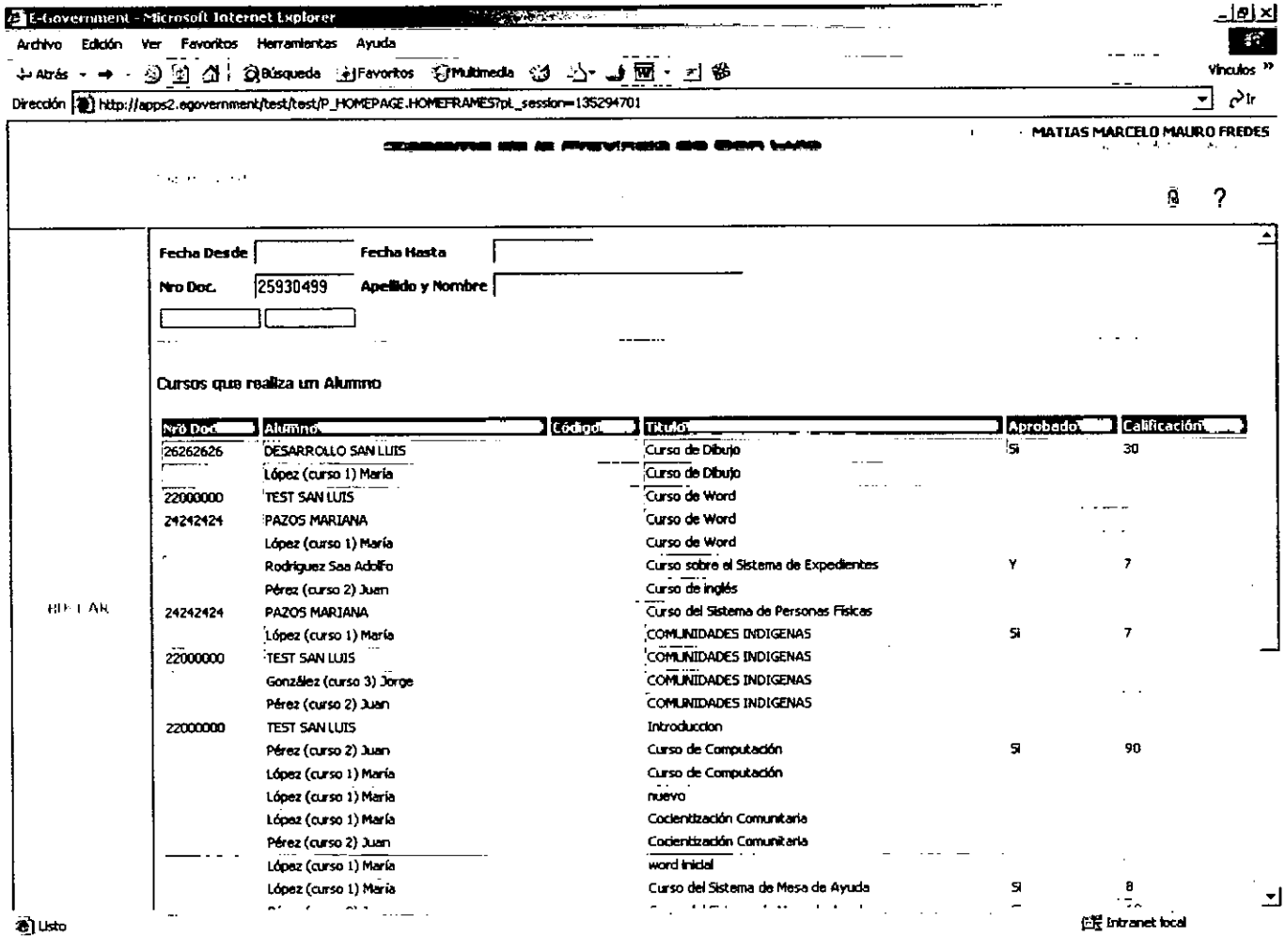
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-060

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

EN LA CONSULTA CURSOS QUE REALIZA UN ALUMNO NO FUNCIOLA BÚSQUEDA POR NÚMERO DE DOCUMENTO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Cursos por alumno**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

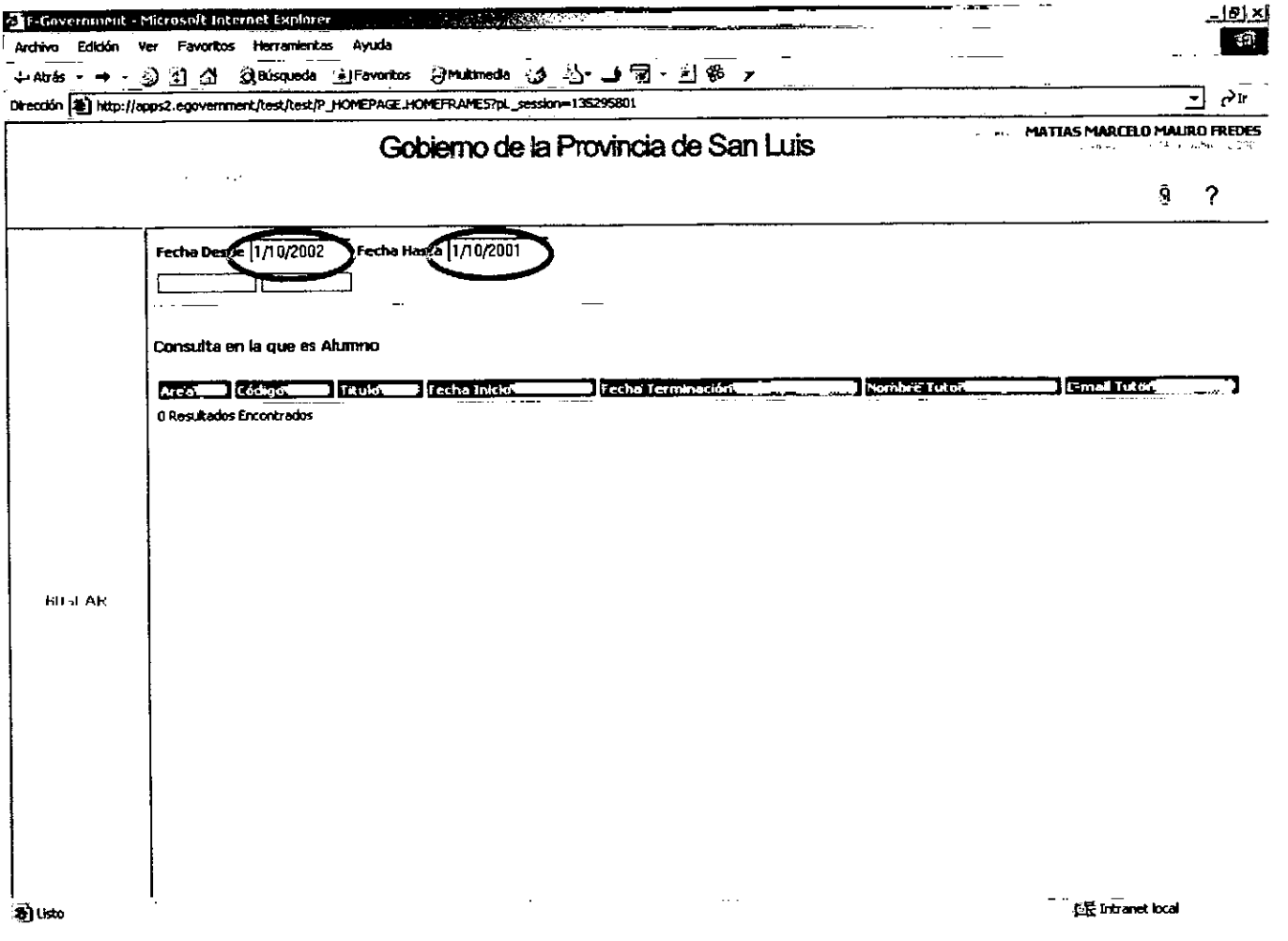
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-058

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

PERMITE REALIZAR CONSULTAS DE CURSOS POR ALUMNOS CON FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:	FUNCIÓN: **Cursos que realiza un alumno**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

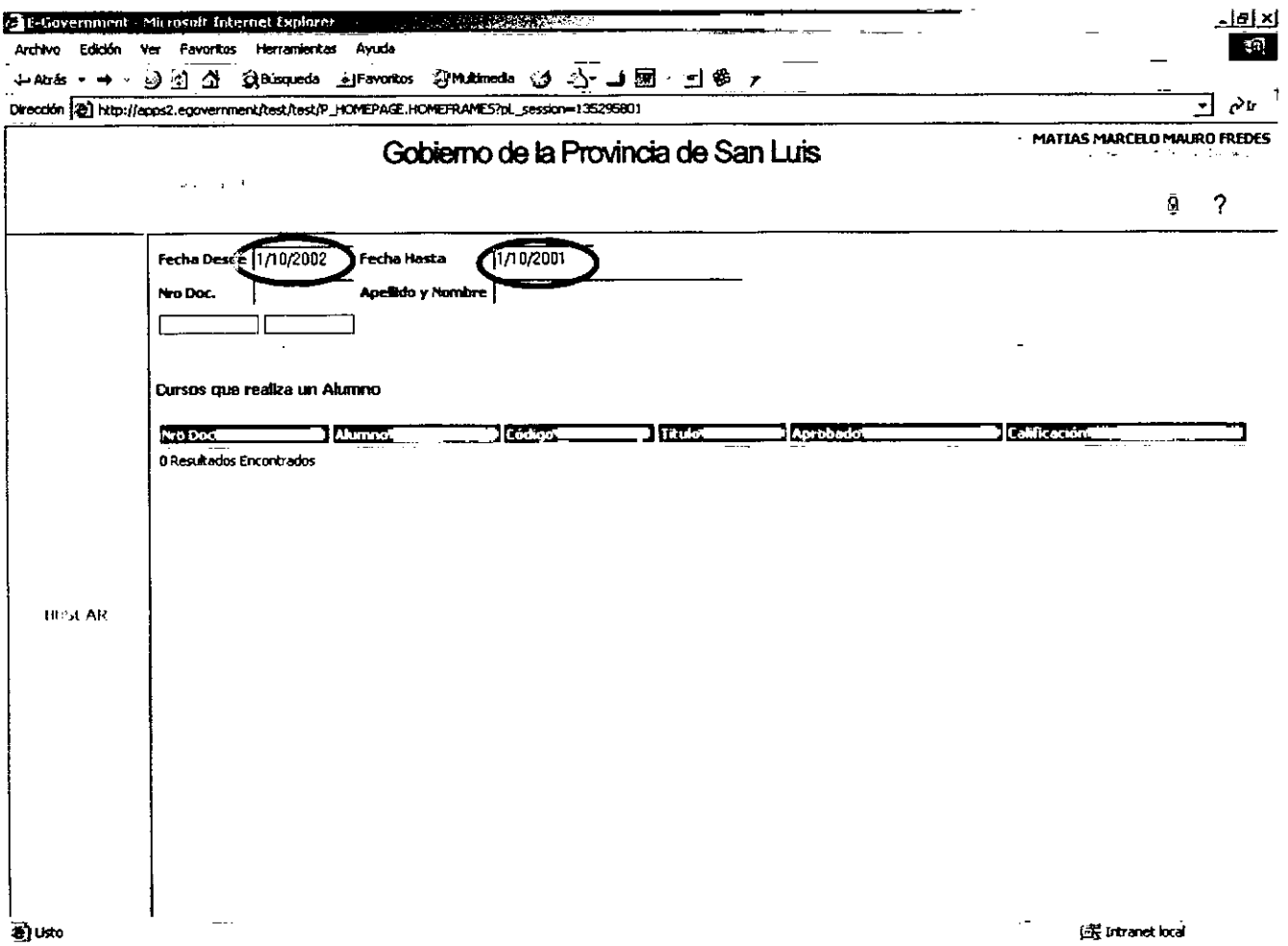
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-059

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES

PERMITE REALIZAR CONSULTAS DE CURSOS QUE REALIZA UN ALUMNO CON FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes**	TELÉFONO: **02652 - 451076**
ORGANIZACIÓN: **S. E. T. I.**	E-MAIL: **mfredes@sanluis.gov.ar**
FECHA: **06 / 11 / 02**	

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT**	SUBSISTEMA: **CONSULTAS Y LISTADOS**
MÓDULO:	FUNCIÓN: **Listado de Usuarios**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

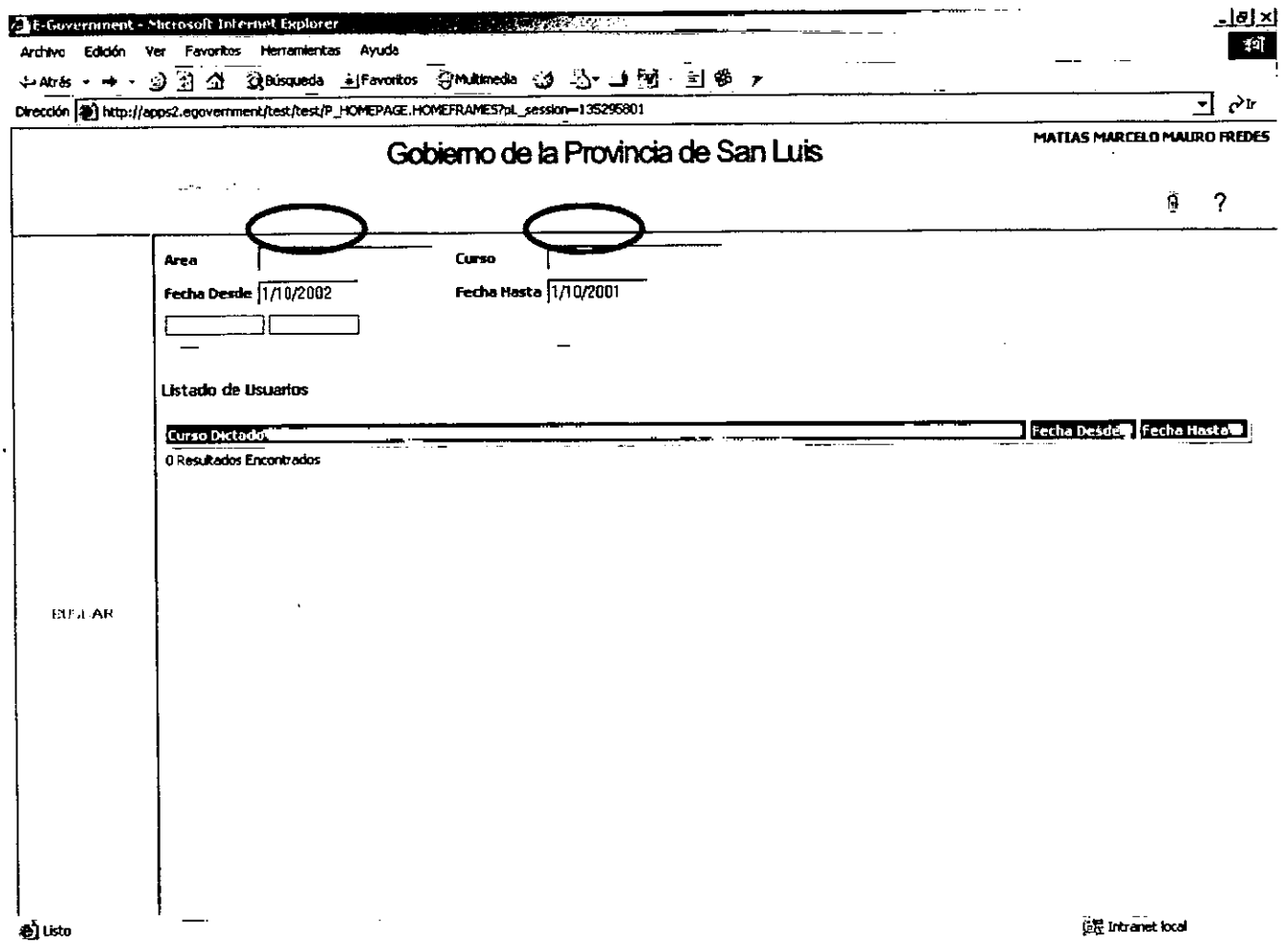
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-062

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

PERMITE REALIZAR CONSULTAS DE LISTADO DE USUARIOS CON FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

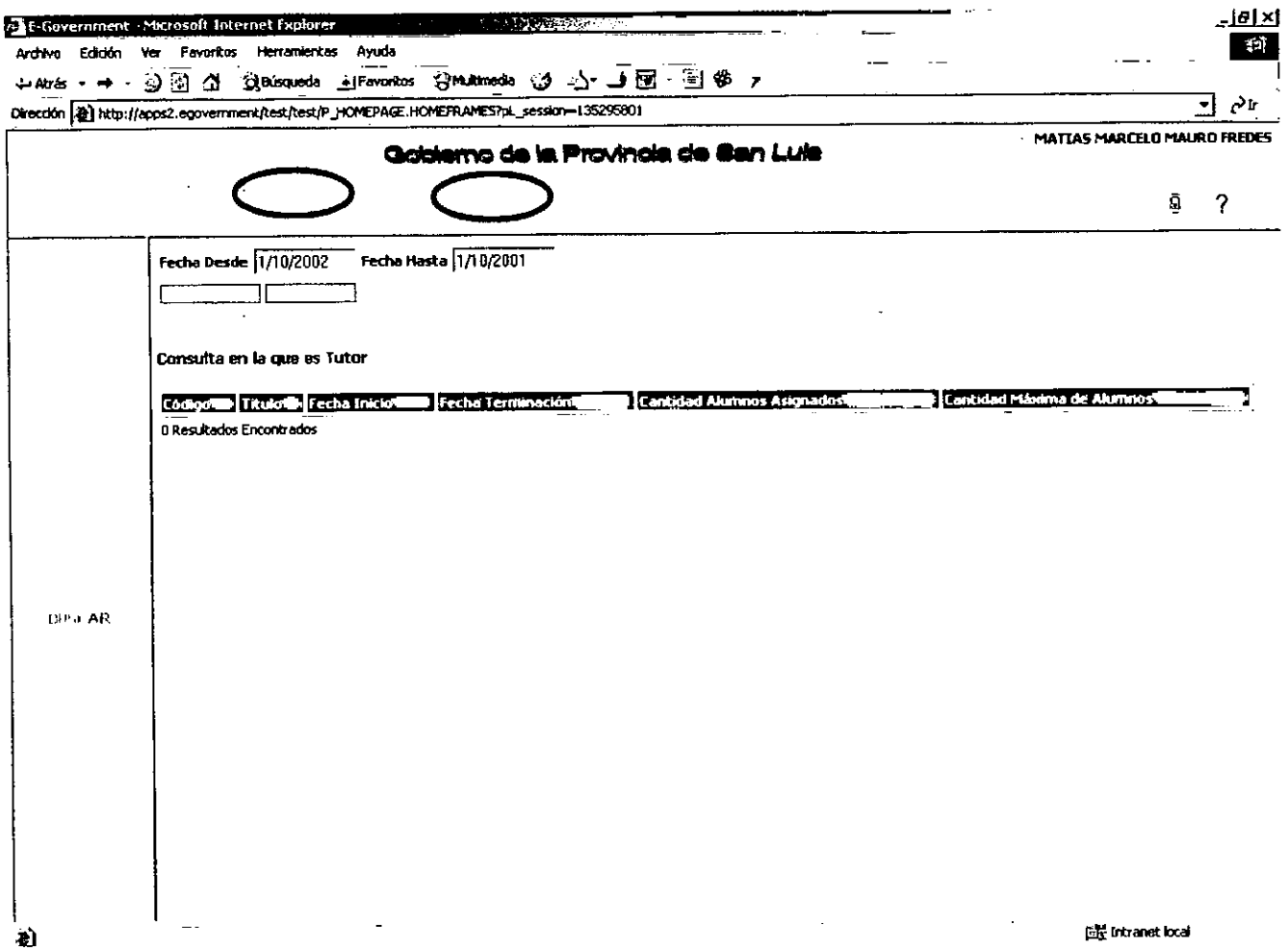
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Cursos por Tutor**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-063

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
PERMITE REALIZAR CONSULTAS DE CURSOS POR TUTOR CON FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

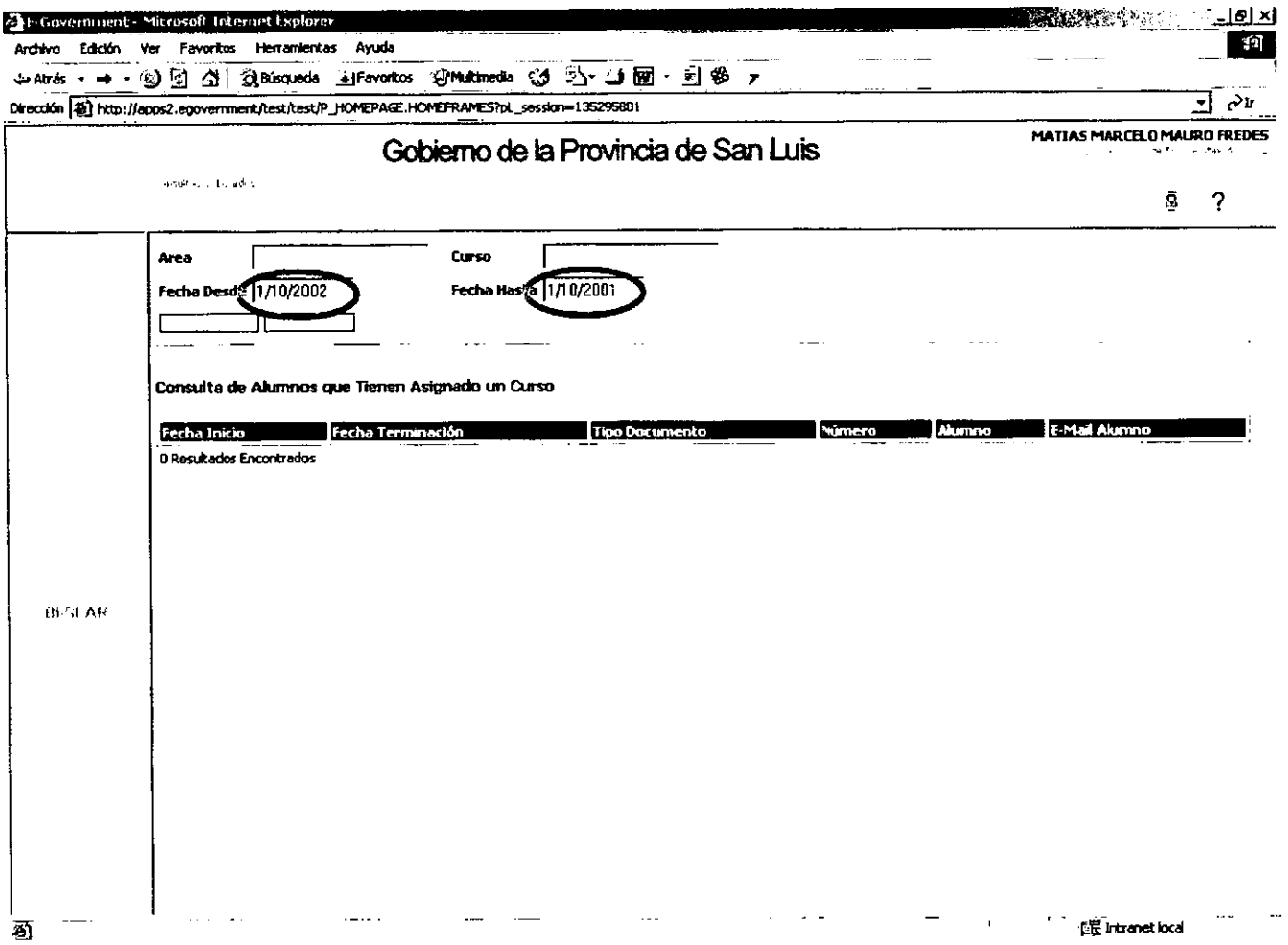
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Alumnos por curso**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-064

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

OBSERVACIONES
PERMITE REALIZAR CONSULTAS DE ALUMONOS POR CURSO FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO:	SUBSISTEMA: **CONSULTAS Y LISTADOS** FUNCIÓN: **Estadísticas**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

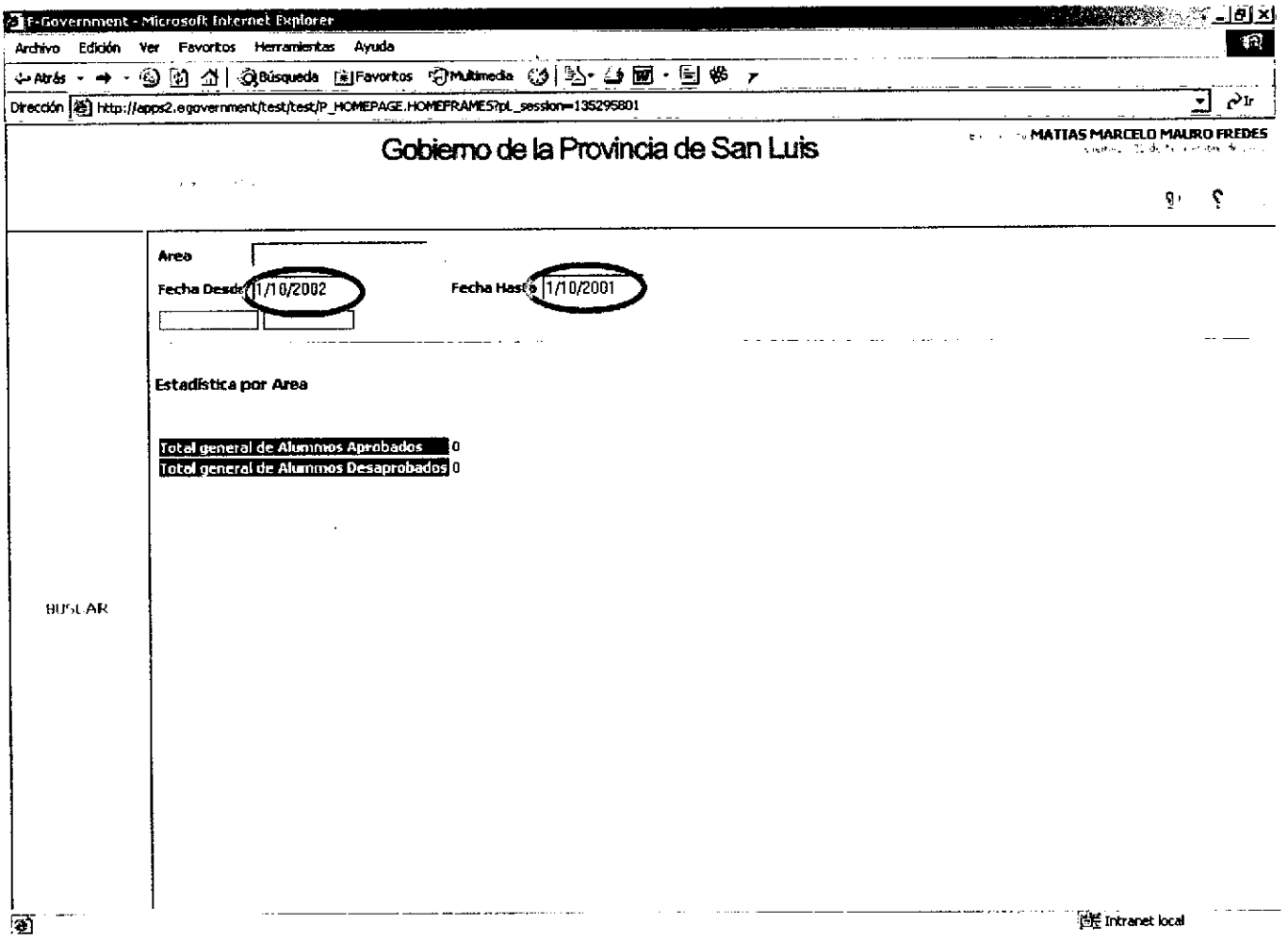
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-066

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

OBSERVACIONES

PERMITE REALIZAR CONSULTAS DE ESTADÍSTICAS CON FECHA DESDE MAYOR QUE FECHA HASTA SIN CONTROL DE QUE ESTO DEVUELVE SIEMPRE UN LISTADO VACÍO.

IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: **E-GOVERNMENT**	

DATOS DEL TESTER	
APELLIDO Y NOMBRE: **Matías Mauro Fredes** ORGANIZACIÓN: **S. E. T. I.** FECHA: **06 / 11 / 02**	TELÉFONO: **02652 - 451076** E-MAIL: **mfredes@sanluis.gov.ar**

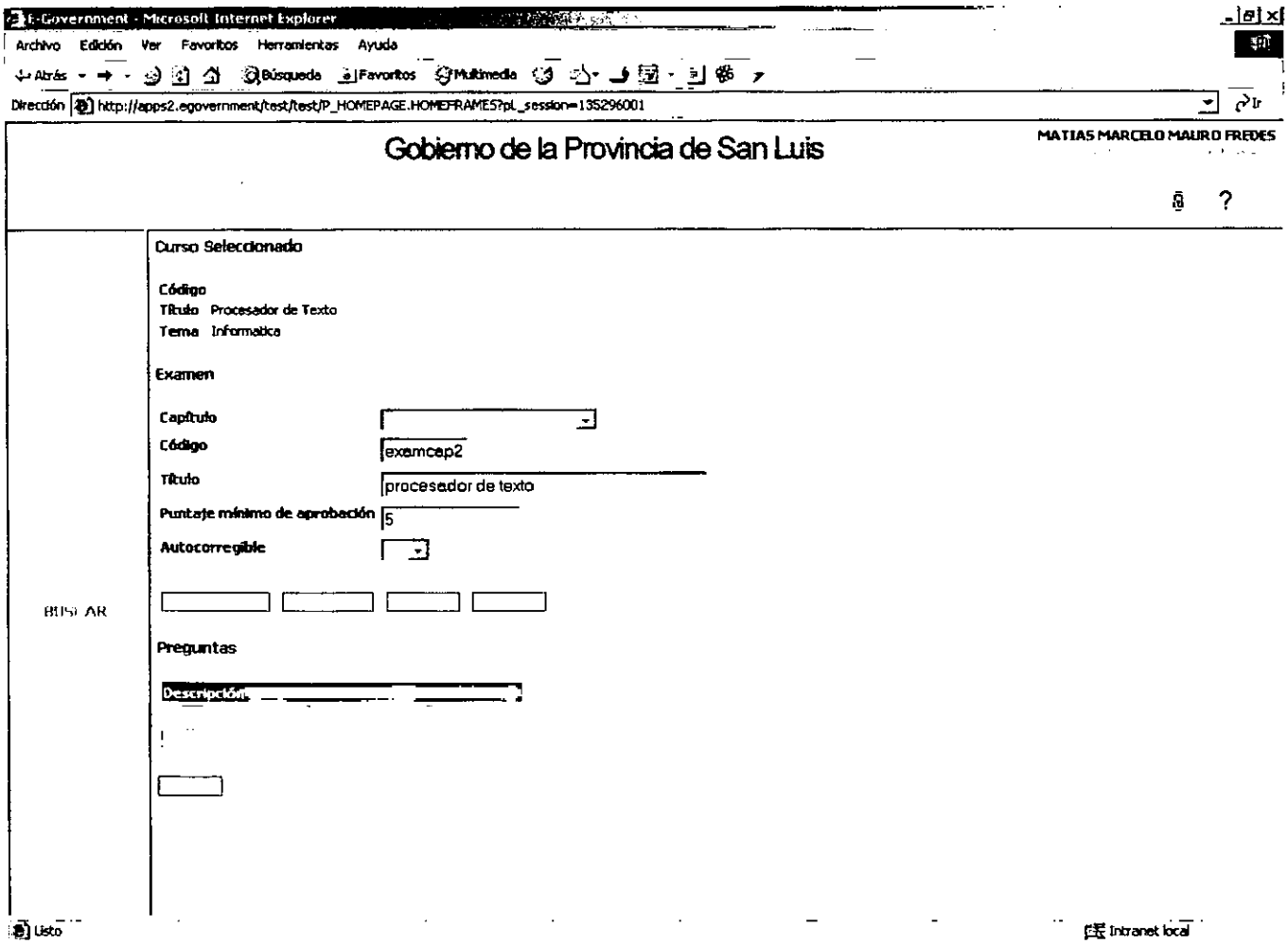
DATOS DE LA PANTALLA	
SISTEMA: **E_GOVERNMENT** MÓDULO: **EXÁMEN**	SUBSISTEMA: **ADMINISTRACIÓN DE CURSOS** FUNCIÓN: **Cargar / Actualizar orden de pregunta**

PLAN DE PRUEBAS ASOCIADO	
ID: **003**	VERSIÓN: **01**

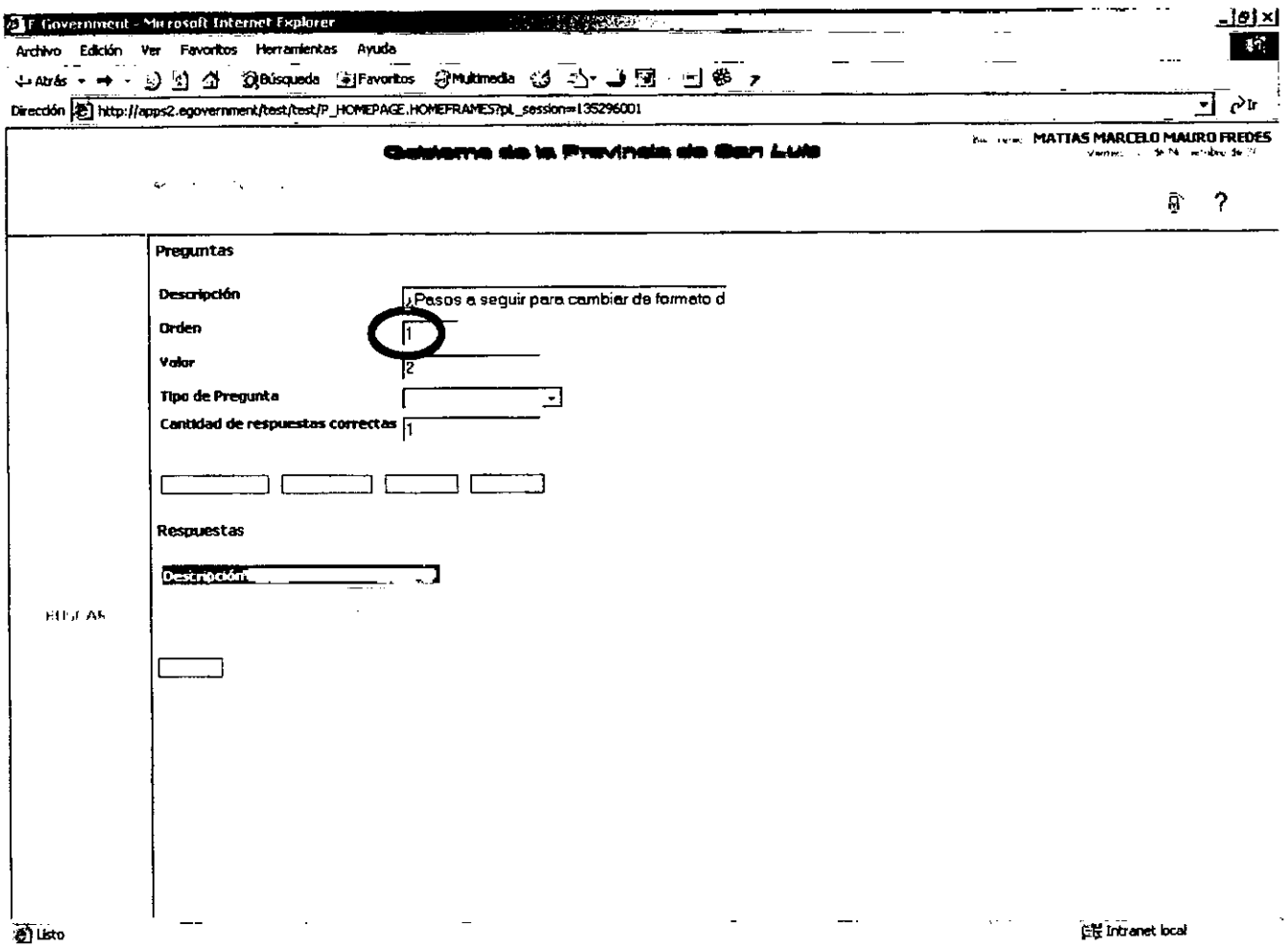
CASO DE PRUEBA ASOCIADO
CP-EG-SC-TA-024 **CP-EG-SC-TA-025**

PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	

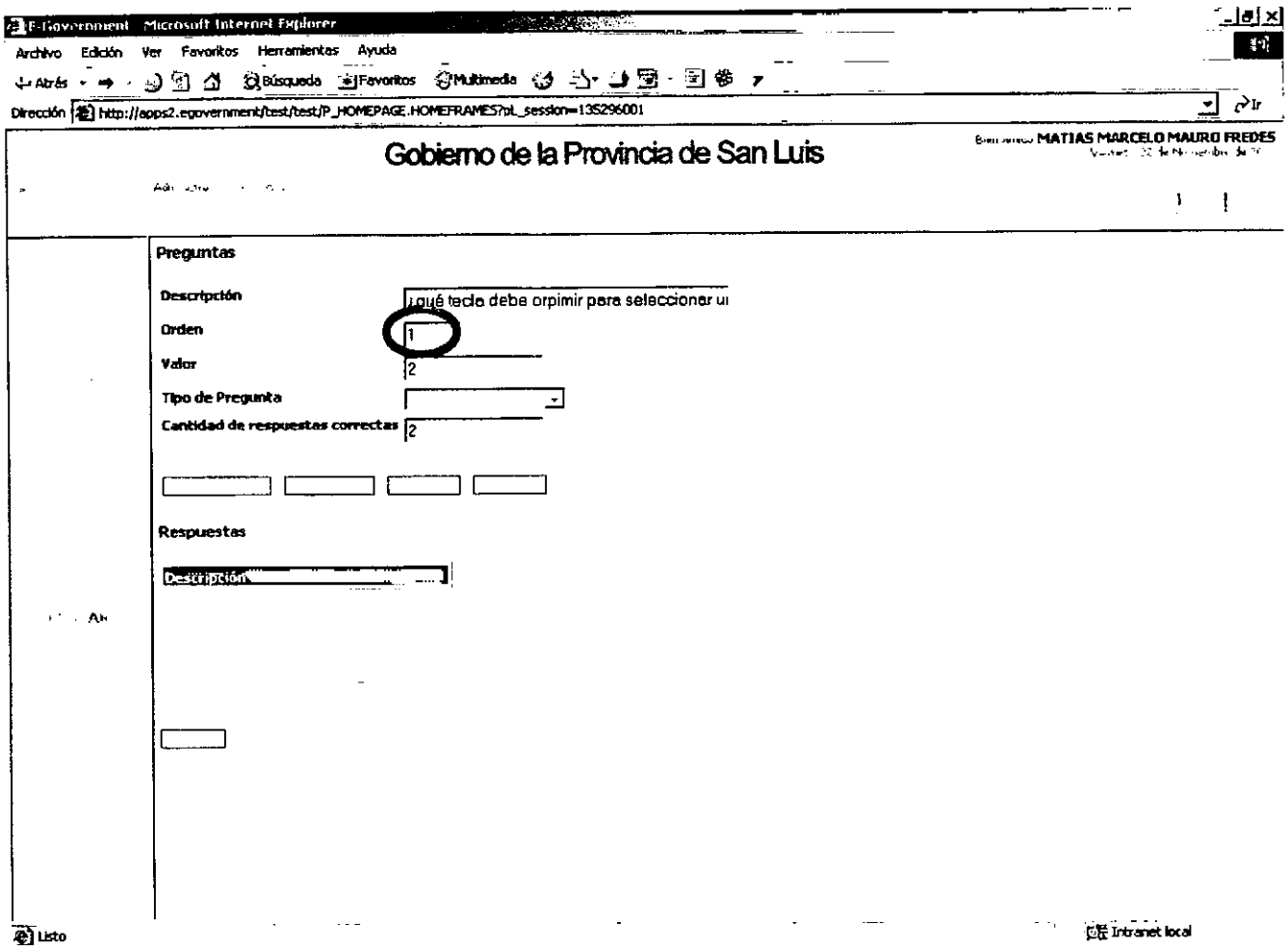
OBSERVACIONES
PERMITE CARGAR VARIAS PREGUNTAS CON EL MISMO ORDEN.
IMAGEN



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



PROYECTO	CAPTURA DE PANTALLAS
NOMBRE: E-GOVERNMENT	



Lic. Silvina R. Acuña
GERENTE DE
CONCIENTIZACION COMUNITARIA
SECRETARÍA DE TECNOLOGÍA DE LA INFORMACION

Secretaría de Estado de Tecnologías de la Información
San Luis