

Informe



SOWTIC

Informe de final sobre actualización tecnológica de IPS

Proyecto: Sistemas RUPA
Revisión 1.0

Contenido

- Informe Final..... 3
 - Propósito.....3
 - Abordaje del Proceso de Migración de Bases de Datos..... 3
 - Evolución de infraestructura - DevOps..... 5
 - Análisis, diseño y desarrollo de nuevo Rupa.....5
 - Avances en la migración de script de Natural a Python..... 6
- Planificación realizada según lo acordado..... 6
 - Plan de migración estratégico..... 7

Informe Final

Propósito

El presente documento se refiere al trabajo realizado sobre el requerimiento planteado por el Instituto de Previsión Social (IPS) para llevar a cabo una actualización tecnológica integral del sistema RUPA.

El proyecto tuvo foco en 4 enfoques centrales: migración de base de datos, implementación de cluster de contenedores, migración de Script y Desarrollo de nueva versión de aplicativo RUPA

Abordaje del Proceso de Migración de Bases de Datos

La migración de la base de datos de Adabas D a PostgreSQL fue un proyecto de envergadura, guiado por la directriz fundamental de preservar el modelo de datos original en su totalidad. Este principio aseguró la continuidad operativa y la integridad de la información crítica, minimizando los riesgos asociados a cambios estructurales profundos.

Fases Clave del Proceso de Migración:

1. Relevamiento Inicial y Planificación Estratégica:

El punto de partida fue un relevamiento exhaustivo de la estructura general de la base de datos de origen, Adabas D. Esta fase incluyó la identificación de esquemas, tablas, índices, vistas, procedimientos almacenados y cualquier otro objeto de base de datos. Se analizaron las relaciones entre las tablas, los tipos de datos utilizados y las restricciones de integridad. Este análisis pormenorizado fue crucial para obtener una dimensión precisa del estado actual de la base de datos, identificar posibles complejidades y desarrollar una planificación detallada de las etapas subsiguientes. Se generaron diagramas entidad-relación (ERD) actualizados y se documentaron todas las dependencias para evitar sorpresas durante la ejecución.

2. Replicación del Esquema y Carga de Datos:

Una vez finalizado el relevamiento y diseñado el esquema objetivo en PostgreSQL, se procedió con la replicación de dicho esquema. Este paso implicó la creación de las tablas, índices y demás objetos en la nueva base de datos de PostgreSQL, asegurando que su estructura reflejara con exactitud la de Adabas D. Posteriormente, se dio inicio a la fase crítica de extracción y carga de datos. Para garantizar la consistencia, se implementaron mecanismos de validación de datos en cada etapa, verificando que los datos extraídos de Adabas D coincidieran con los cargados en PostgreSQL, tanto en cantidad como en formato.

Estrategia de Migración Adaptativa y Eficiente:

La estrategia de migración se diseñó para ser flexible, adaptándose al volumen y las características específicas de cada tabla a transferir, buscando optimizar los tiempos:

- **Proceso ETL Generalizado para la Mayoría de las Tablas:**

Para la gran mayoría de las tablas, se implementó un proceso ETL (Extracción, Transformación y Carga) estandarizado. Este proceso consistió en extraer los datos de Adabas D, realizar transformaciones mínimas y estrictamente necesarias para adaptarlos al formato de PostgreSQL (por ejemplo, ajustes de tipos de datos si fuera indispensable, pero siempre manteniendo la integridad del dato original), y finalmente cargar la tabla completa en la base de datos de destino. Se utilizaron herramientas y scripts personalizados para automatizar este proceso, garantizando la eficiencia y reduciendo la posibilidad de errores manuales.

- **Carga Anticipada para Grandes Volúmenes de Datos:**

En el caso de tablas que contenían un volumen excepcionalmente grande de datos, se optó por una estrategia de carga anticipada. Esta medida implicó transferir estos datos a la base de datos de PostgreSQL con antelación a la fecha de migración principal. El objetivo primordial de esta acción fue reducir significativamente el tiempo total del proceso de migración, minimizando la ventana de inactividad del sistema y los posibles impactos operativos. Se estableció un proceso de sincronización incremental para estas tablas, asegurando que cualquier cambio en la base de datos de origen durante el período de precarga se replicara en PostgreSQL antes del corte final.

- **Desacople del Esquema de Autorías para Gestión Histórica:**

Una decisión estratégica fue desacoplar el esquema de autorías del modelo de datos principal. Esto significó que, en lugar de migrar sus datos, se migró únicamente su estructura. El esquema original de autorías en Adabas D fue tratado como un esquema histórico. Esta aproximación permitió simplificar el nuevo entorno, mantener la información histórica accesible para auditorías o consultas específicas y evitar la carga innecesaria de datos que no serían activos en el nuevo sistema. Se definió un plan para el acceso y la consulta de esta información histórica cuando fuera necesario.

Consideraciones Críticas sobre el Formato y la Integridad de la Información:

Durante la fase de extracción y transformación, fue necesario realizar un saneamiento de datos. Este proceso implicó la eliminación de datos no deseados o inconsistentes que pudieron haber surgido de la extracción o que eran obsoletos en el contexto del nuevo sistema. Sin embargo, se mantuvo una premisa estricta: una vez finalizada la migración y cargados los datos en PostgreSQL, no se permitiría ninguna modificación del esquema ni de los datos transferidos. Esta política rigurosa fue fundamental para asegurar la integridad total de la información migrada, garantizando que los datos en PostgreSQL fueran una representación fiel y precisa de los datos de Adabas D en el momento de la migración final. Se implementaron procedimientos de control de calidad y auditoría post-migración para validar la consistencia y la exactitud de los datos.

Problemas y soluciones en la carga de datos

Se identificó un problema recurrente de errores durante la carga de un grupo de tablas: la causa principal fue el uso de ^_ como separador en los archivos de origen, un formato no estándar. Se ajustaron los scripts de descarga y se añadió uno nuevo para eliminar valores NULL, lo que permitió cargar archivos de ETL con mínima intervención manual y mayor estabilidad.

Evolución de infraestructura - DevOps

La estrategia de abordaje para la migración de IPS se centró en la implementación de una solución integral de infraestructura que garantizara la gobernanza de los sistemas. Para ello, se desplegaron dos clústeres de contenedores, uno en el datacenter de DPSIT y otro en el datacenter de IPS, ambos diseñados para asegurar alta disponibilidad, tolerancia a fallos y escalamiento automático. Esta arquitectura se complementó con el desarrollo e implementación de un workflow de integración y despliegue continuo (CI/CD), que estandarizó los flujos, garantizando consistencia, trazabilidad y repetibilidad en la entrega de software, y automatizando la construcción y el despliegue en todos los ambientes (desarrollo, testing, pre-producción y producción) mediante Pipelines de CI/CD, Charts de Helm y ArgoCD.

Durante el período de implementación, se priorizó la estabilización, optimización y mejora del pipeline de CI/CD, enfocándose en la automatización de procesos y la reducción de fallos operacionales. Se avanzó significativamente en la estandarización de los flujos de integración y despliegue continuo, lo que aseguró una mayor consistencia y trazabilidad en los procesos de entrega de software. Se mantuvo una estrecha coordinación con los equipos de desarrollo, realizando reuniones clave para habilitar accesos, alinear el estado de los merge requests del backend y validar la integración de cambios en los distintos entornos.

Asimismo, se llevaron a cabo presentaciones y revisiones de artefactos técnicos, como Pipelines de CI/CD, Charts de Helm y configuraciones de ArgoCD, con los equipos involucrados. Esta actividad reforzó la trazabilidad, transparencia y automatización de los despliegues. Se logró la resolución de incidencias críticas, incluyendo la estabilización de los healthchecks en OKD que afectaban la disponibilidad de servicios y la corrección de fallos relacionados con OAuth, mejorando la confiabilidad de la autenticación. La colaboración con el equipo de Infraestructura del organismo fue constante, participando en reuniones estratégicas para alinear criterios operativos y definir flujos de procesamiento de archivos. Finalmente, se realizó una revisión del backend de la aplicación RUPA, asegurando su correcta integración a través del pipeline de CI/CD para un despliegue seguro y controlado.

Se han configurado cuatro entornos distintos para la nueva aplicación RUPA: Desarrollo, Testing, Pre-Producción y Producción.

Los entornos de Desarrollo y Testing se encuentran alojados en el clúster de DPSIT. Por otro lado, los entornos de Pre-Producción y Producción residen en el clúster del centro de datos de IPS.

Cada uno de estos entornos opera de manera independiente y cuenta con sus propios artefactos exclusivos de la aplicación RUPA, tales como el backend, el frontend, la base de datos, Keycloak, Redis y el almacenamiento S3.

Análisis, diseño y desarrollo de nuevo Rupa

El proyecto RUPA (Registro Único de Prestadores de Actividad) del Instituto de Previsión Social (IPS) de la Provincia de Buenos Aires representa una modernización integral de la gestión de

empleadores, afiliados y relaciones laborales mediante el desarrollo de una solución de software robusta y escalable. Esta aplicación se compone de un *Backend* basado en una API REST construida con **Django** y **PostgreSQL**, y un *Frontend* desarrollado en **Next.js** y **React**, asegurando una interfaz moderna y responsiva. La seguridad y el control de acceso se implementan rigurosamente a través de la integración con **Keycloak**, estableciendo un sistema de autenticación y autorización basado en roles. Las funcionalidades principales de la plataforma se centran en la gestión integral de cuatro áreas críticas: **Empleadores** (CRUD completo), **Afiliados** (listado y gestión vinculada a empleador), **Relaciones Laborales** (gestión de 'Revistas') y la **Liquidación de Salarios** (consulta con filtros avanzados). El sistema dispone de una arquitectura modular que facilita el mantenimiento y la escalabilidad. Finalmente, el frontend incluye un *Dashboard* personalizado por rol, capacidades de búsqueda, filtros y la **exportación de datos** a CSV, proporcionando una herramienta eficiente y segura para la administración de la información previsional.

Avances en la migración de script de Natural a Python

El proyecto de migración de scripts de Natural a Python exhibe un **sólido enfoque inicial en la planificación y el desarrollo técnico**, lo cual es un factor crítico de éxito. La actividad más intensiva fue la **traducción de código**, lo que asegura el avance en la conversión del *core* del sistema. Este esfuerzo se complementó con una planificación rigurosa para la **adquisición de insumos mínimos**, como entornos de prueba y datos fuente (archivos .DAT y .TXT), lo cual subraya la prioridad otorgada a la fase de preparación y obtención de recursos. En este contexto, se identificaron **tareas de Prioridad Alta**, centradas en la disponibilidad de entornos y la traducción de scripts clave, como el de carga SAP.

Adicionalmente, se observa un **énfasis primordial en la validación y la gestión de código** para garantizar la integridad funcional. La metodología adoptada prioriza la **validación de "caja negra"**, requiriendo la equivalencia funcional **idéntica** entre los *outputs* de Python/Postgres y Natural/Adabas. El flujo de trabajo incorporó la **entrega temprana de scripts traducidos** al equipo de pruebas, lo que permite una validación continua. La **Gestión de Código**, con el uso de **Gitlab** para el control de versiones, asegura la trazabilidad y la colaboración eficiente en el desarrollo.

Planificación realizada según lo acordado

Definición de modelo de infraestructura

- Implementación de clúster de Kubernetes
- Desarrollo de solución de integración continua
- Propuesta de charlas técnicas relacionadas a HELM

ITEM 1 - Migración de base de datos RUPA

- Análisis de modelo y submodelos
- Definición de roles y privilegios
- Exportación e importación de procesos de DBSCHemas

- Especificación funcional de procesos de migración
- Desarrollo de proceso de migración
- Exportación de datos
- Migración inicial
- Modificación de sistemas existentes en entornos de prueba
- Validación y pruebas
- Pruebas de la aplicación RUPA: Se finalizó la corrección de la versión 5.

ITEM 2 - Desarrollos de procesos de Natural a Python

- Especificación funcional procesos de Natural a Python
- Desarrollo de procesos de natural a Python
- Validación y pruebas
Pruebas de script Natural a Python
- Cargas en paralelo (tercera etapa)

ITEM 3 - Desarrollo de software para carga de información

- **Historias de usuario:** modelo aprobado.
- **Diseño UX/UI finalizado:** mockups y prototipos de la interfaz de usuario aprobados.
- **Frontend Desarrollado:** Código y funcionalidades del frontend completados.
- **Backend desarrollado:** Código y funcionalidades del backend completados.
- **Software Testeado:** Reporte final de pruebas que confirma que el software está listo para producción.
- **Sistema en Producción (Go Live):** Sistema lanzado oficialmente y operando en el entorno de producción.
- **Ajustes Post Go Live Realizados:** Arreglos críticos y arreglos menores

Plan de migración estratégico

Los documentos relacionados con el plan de migración se encuentran centralizados en una [carpeta específica](#). Estos archivos corresponden a versiones en desarrollo y comprenden, principalmente, el [Plan de Migración](#) y el [Plan de Gestión de Riesgos](#).

El objetivo de estos documentos es identificar, evaluar y establecer estrategias de mitigación frente a los riesgos asociados al proyecto de migración. Este incluye la transición de la base de datos desde ADABAS a PostgreSQL, la conversión de scripts de Natural a Python, el desarrollo de la nueva aplicación RUPA y la implementación de un clúster de Kubernetes para su soporte. Se detallan los principales riesgos potenciales, junto con las acciones propuestas para su monitoreo y control.