

ProyectoAviones

David Gomez Rivas

31 de enero de 2024

Índice

1. Introducción	3
2. Análisis del Problema	3
3. Solución Propuesta	4
4. Pseudocódigo	5
5. Análisis del Código	6
6. Diagrama	7
7. Conclusión	8

1. Introducción

Este documento presenta el desarrollo parcial de una aplicación de escritorio destinada a consumir, procesar y almacenar datos de una API seleccionada. La aplicación, diseñada para operar en múltiples plataformas, se encuentra en una etapa intermedia, con funcionalidades clave ya implementadas en el escritorio y planes para una futura expansión a plataformas móviles. Hasta el momento, hemos desarrollado la capacidad de la aplicación para interactuar con datos de una API, procesarlos en diferentes formatos (SQL, XML, JSON), y ofrecer una interfaz de usuario intuitiva a través de Java Swing, junto con la generación de informes estructurados.

2. Análisis del Problema

El proyecto se enfrenta al desafío de crear una solución cohesiva que pueda integrar datos de diversas fuentes y formatos, asegurando una funcionalidad robusta tanto en aplicaciones de escritorio como móviles. La necesidad de una aplicación multiplataforma surge de la diversidad de los entornos de usuario y los requisitos de accesibilidad de datos. La solución debe ser capaz de obtener datos de una API específica, transformar estos datos para su almacenamiento en una base de datos SQL y proporcionar interfaces interactivas para la manipulación y visualización de datos, todo mientras se mantiene una arquitectura coherente y escalable.

3. Solución Propuesta

La solución propuesta aborda estos desafíos mediante una arquitectura modular que consta de:

- **Conexión a la API:** Un módulo dedicado a establecer una conexión con la API seleccionada, extraer datos y procesarlos para su uso en la aplicación.
- **Procesamiento y Almacenamiento de Datos:** Funcionalidades para transformar los datos extraídos en formatos adecuados (SQL, XML, JSON) y almacenarlos en una base de datos local o archivos según corresponda.
- **Interfaz de Usuario Java Swing:** Una aplicación de escritorio desarrollada con Java Swing que ofrece una interfaz gráfica intuitiva para interactuar con los datos almacenados. Incluye capacidades para leer y escribir en la base de datos y en archivos XML y JSON, y para exportar datos a informes predefinidos.
- **Generación de Informes:** Un componente que utiliza iReports para crear informes estructurados basados en los datos visibles dentro de la aplicación, facilitando la interpretación y el análisis de los datos.

Esta arquitectura no solo satisface los requisitos actuales del proyecto sino que también establece una base sólida para la futura integración de la aplicación móvil.

4. Pseudocódigo

Inicio

```
// Obtener Datos de la API
Definir APLURL como String
Definir datosAPI como Lista
Conectar a APLURL
datosAPI = ExtraerDatos(APLURL)
Para cada dato en datosAPI
    TransformarDato a formato SQL
    InsertarDatoEnBaseDeDatos(dato)

// Interfaz de Usuario Java Swing
Iniciar InterfazUsuario
Mientras usuario interactua con la InterfazUsuario
    Si usuario selecciona 'Leer-Base-de-Datos'
        MostrarDatosDesdeBaseDeDatos()
    Fin Si
    Si usuario selecciona 'Leer-Archivo-XML'
        datosXML = LeerDatosDesdeXML('rutaArchivo.xml')
        MostrarDatos(datosXML)
    Fin Si
    Si usuario selecciona 'Leer-Archivo-JSON'
        datosJSON = LeerDatosDesdeJSON('rutaArchivo.json')
        MostrarDatos(datosJSON)
    Fin Si
    Si usuario selecciona 'Exportar-Informe'
        datosVisibles = ObtenerDatosVisibles()
        GenerarInforme(datosVisibles)
    Fin Si
Repetir

// Generacion de Informes
Definir GenerarInforme como Funcion(datos)
    Utilizar iReports para crear informe basado en 'datos'
    GuardarInforme('rutaInforme')
Fin Funcion

Fin
```

5. Análisis del Código

- Estructura y Organización El proyecto parece estar bien organizado en múltiples archivos, cada uno enfocado en una tarea específica (conexión API, lectura de datos, interfaz de usuario, etc.), lo cual es una buena práctica para mantener el código modular y fácil de mantener.
- Conexión a la API (API.java) Este archivo probablemente maneja la conexión con la API externa, extrayendo los datos necesarios. Sería clave evaluar cómo se manejan las excepciones y los errores de conexión para garantizar la robustez del código.
- Interfaz de Usuario (Inicial.java, AgregarDatos.java, VerDatos.java, MenuXML.java, MenuJSON.java) Estos archivos parecen estar relacionados con la interfaz de usuario Java Swing, cada uno posiblemente representando diferentes ventanas o paneles dentro de la aplicación. Es importante asegurarse de que la interfaz sea intuitiva y que se sigan las mejores prácticas de diseño de UI/UX.
- Manipulación de Datos (LeerXML.java, LeerJSON.java) Estos archivos indican que el sistema puede leer y procesar datos desde archivos XML y JSON, lo cual es esencial para la flexibilidad en el manejo de diferentes formatos de datos. Sería recomendable revisar la eficiencia del parsing y la transformación de estos datos.
- Exportación de Informes (Reporte.java) Este archivo sugiere funcionalidad para generar informes, posiblemente utilizando iReports. Sería crucial examinar cómo se estructuran los datos en los informes y garantizar que la generación de informes sea dinámica y adaptable a los datos en tiempo real.
- Entrada del Programa (Main.java) Este archivo probablemente actúa como el punto de entrada de la aplicación, inicializando la interfaz de usuario y las dependencias necesarias. Es fundamental que este archivo esté bien organizado para facilitar la comprensión del flujo general de la aplicación.

6. Diagrama



7. Conclusión

Hasta el momento, la solución implementada ha demostrado ser eficaz en el manejo de datos de múltiples fuentes y formatos, ofreciendo una interfaz de usuario coherente y funcionalidades de generación de informes. Sin embargo, aún queda trabajo por hacer, particularmente en el desarrollo de la aplicación móvil y en la mejora continua de la arquitectura existente para asegurar su escalabilidad y mantenibilidad. La reflexión sobre los logros actuales y las áreas de mejora servirá de guía para las siguientes fases del proyecto.