

Oneconomy

Drama Queens

Marzo 28 de 2021



Integrantes:

Santiago Alexander Rodríguez Becerra

Osman Beltrán Murcia

Santiago Fernández Becerra

David Santiago Meneses Cifuentes

Sonia Carolina Molina Moreno

Miguel Ángel Uribe García

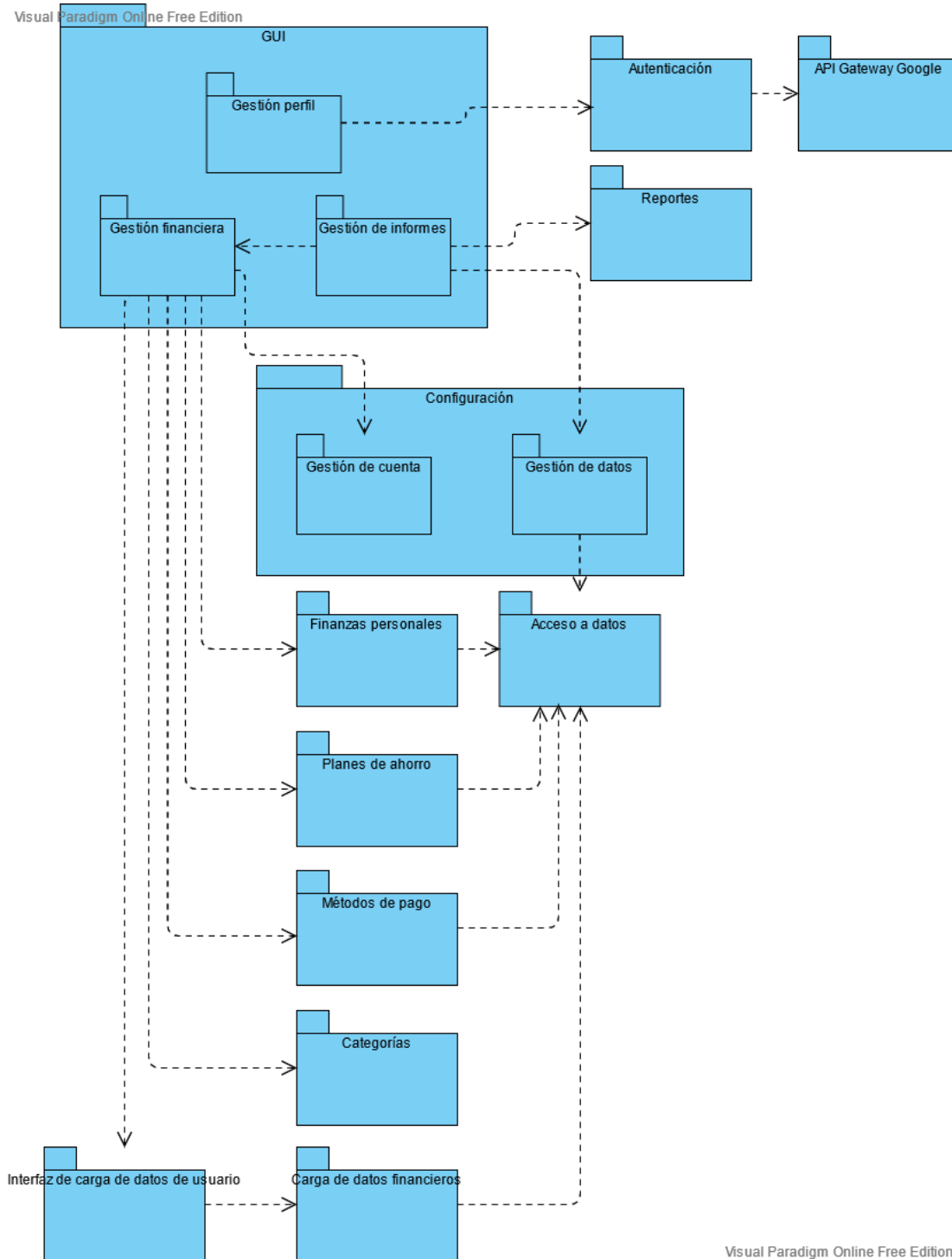
Docente:

Anabel Montero Posada

Fundamentos de Ingeniería de Software

Laboratorio 2

1. Diagrama de paquetes inicial
Ver anexo “Diagrama de paquetes inicial”



2. Investigación y descripción de la arquitectura a usar en el proyecto

Propuesta MVC

El paradigma modelo-vista-controlador se utiliza principalmente para separar la presentación del funcionamiento interno de un software. Este estilo de arquitectura de software destaca por la facilidad de separación de componentes en desarrollo, de tal forma que tanto los datos, la lógica de negocio y la vista tengan mayor completitud, sin necesidad de que el acoplamiento se vea entorpecido. De igual forma, su facilidad de adaptación al cambio y capacidad de desarrollo en paralelo en diferentes componentes permite la asignación de diferentes roles especializados para la ejecución de estos. Por ende, el uso del modelo MVC se consolida como una opción óptima a tomar en cuenta en el proyecto Oneconomy.

Como objetivo principal del uso de este estilo de arquitectura de software es conservar la separación de la interfaz de usuario con el manejo de los datos, ya que Oneconomy tratará con información personal confidencial de cada usuario. Asimismo, la evaluación de seguridad de los datos es un aspecto fundamental en el uso del aplicativo, característica que se asegurará por mantener los datos de forma distribuida según el paradigma MVC.

Librerías o tecnologías adicionales para incorporar:

Para el correcto desarrollo e implementación del proyecto Oneconomy es necesario implementar el uso de librerías especializadas para el cumplimiento de funciones de autenticación, mantenimiento de seguridad de la información y acceso a datos. Cabe resaltar que se proveerá mucho más nivel de detalle en el siguiente apartado relacionado con descripción de paquetes en el modelo de arquitectura a trabajar.

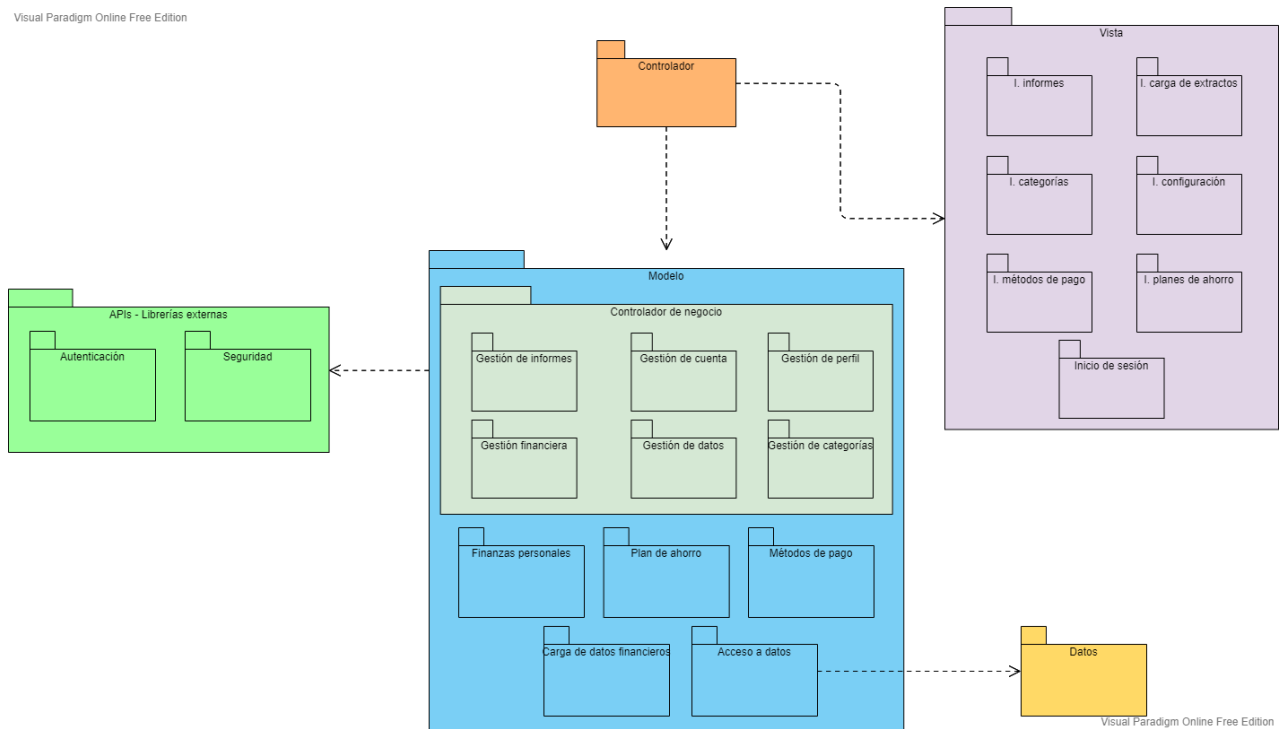
En este apartado destaca el uso de Firebase, plataforma tecnológica provista por Google que permite el manejo, almacenamiento y mantenimiento de los datos de identidad en las sesiones de los usuarios que usarán el aplicativo. De igual forma, este aspecto es fundamental y altamente significativo para el funcionamiento del aplicativo, ya que proporciona al usuario seguridad en el almacenamiento de sus datos personales, asimismo, flexibilidad entorno a la cuenta a usar para obtener acceso a las funcionalidades de Oneconomy.

En el caso del aseguramiento de seguridad de los datos se optará por implementar algoritmos de encriptación robustos, acorde a los datos que trataremos, los cuales serán principalmente personales financieros.

Por otro lado, para realizar el debido acoplamiento de la base de datos MySQL se deberá implementar el uso del conector provisto por Oracle, elemento fundamental para la comunicación entre el aplicativo y la base de datos a desarrollar.

3. Diagrama de la arquitectura de sistema

- Ver anexo “Diagrama MVC”



Descripción de los paquetes:

- **API:**

- **Autenticación:** Paquete que tendrá el sistema de autenticación que identificara al usuario de la aplicación, mediante el SDK de Firebase Authentication, haremos uso de la implementación de la biblioteca FirebaseAuthUI, que permitirán el llamado de diferentes métodos de autenticación de manera segura, tal como el acceso por medio de correo electrónico y contraseña o directamente a través de varios proveedores como Google, Facebook o Twitter [1].
- **Seguridad:** Paquete con las librerías y métodos necesarios para mantener la seguridad de la información de los usuarios ante posibles fugas, filtraciones y robo de esta, en el paquete se tendrán métodos que permitan encriptar/desencriptar datos, chequeo de la información, generación de tokens, la idea es que toda la información que se consulte y se envíe a la base de datos pase por este módulo para realizar la aseguración y verificación correspondiente.

En cuanto al uso de algoritmos para la encriptación de los datos, se tiene pensado realizar un hash haciendo uso de SHA-256 y SHA-512 para los datos más sensibles (métodos de pago, contraseñas, etc.) y verificación de checksum. En cuanto a los datos menos sensibles, pero aun importantes (como lo son transacciones, datos personales y de contacto) se tiene pensado realizar una encriptación asimétrica, donde solo el usuario tendrá acceso a la llave privada desde su dispositivo predeterminado [2] [3].

Para el desarrollo de este módulo no se provee que sea necesario hacer uso de librerías externas a las que se encuentran predefinidas dentro del lenguaje, sin embargo, no está completamente descartado por posibles inconvenientes de tiempo de entrega o rendimiento de los algoritmos.

- **Modelo:**
 - **Controlador de negocio:**
 - Gestión de informes: Gestión de los informes que sean generados a partir de los datos subidos previamente, estas estadísticas deben de quedar guardadas en la base de datos.
 - Gestión de cuenta: Gestión de los datos personales del cliente, estos datos deben de quedar guardados.
 - Gestión financiera: Gestiona datos como, la cantidad de dinero en tiempo real, planes de ahorro, finanzas, entre otros, estos datos deben de ser guardados en la base de datos.
 - Gestión de datos de actividad financiera: Gestiona los datos leídos desde el extracto y los ingresados manualmente, estos datos deben de ser guardados en la base de datos.
 - Finanzas personales: Paquete que tendrá todos los métodos necesarios para el manejo de la cuenta frente a finanzas personales y consejos para el usuario.
 - Plan de ahorro: Paquete que tendrá toda la información sobre los planes de ahorro como Nombre, valor y análisis de este.
 - Métodos de pago: Paquete que contendrá todos los métodos de pago del usuario los cuales fueron inscritos por los datos.
 - Carga de datos financieros: se encarga de cargar los datos con los cuales se harán los análisis financieros.
 - Acceso a datos: Se encarga de comunicarse con la base de datos del sistema.
- **Datos:** Base de datos del sistema - donde se almacena todo lo necesario para el correcto funcionamiento de la aplicación.

- **Interfaz:** Se conectan todos los paquetes de control con sus interfaces respectivas
 - I informes: Interfaz de informes.
 - I carga de extractos: Interfaz de carga de datos.
 - I categorías: Interfaz de categorías.
 - I configuración: Interfaz de la configuración de la app.
 - I métodos de pago: Interfaz de los métodos de pago.
 - I planes de ahorro: Interfaz de los planes de ahorro.
 - Inicio de sesión: Interfaz para la autenticación del usuario.

4. Anexos

- Diagrama de paquetes inicial

<https://online.visual-paradigm.com/community/share/oneconomy-diagrama-de-paquetes-hztsygryv>

- Diagrama MVC

<https://online.visual-paradigm.com/community/share/mvc-oneconomy-hztyjim3y>

5. Referencias

- [1] «Firebase Authentication», *Firebase*. <https://firebase.google.com/docs/auth?hl=es> (accedido mar. 28, 2021).
- [2] GlossaryAdmin, «SHA-2», *Pro2col File Transfer Glossary*. <https://glossary.pro2col.com/sha-2/> (accedido mar. 28, 2021).
- [3] «SHA-256 Algorithm Overview», *Solarwinds MSP*, sep. 12, 2019. <https://www.solarwindmsp.com/blog/sha-256-encryption> (accedido mar. 28, 2021).