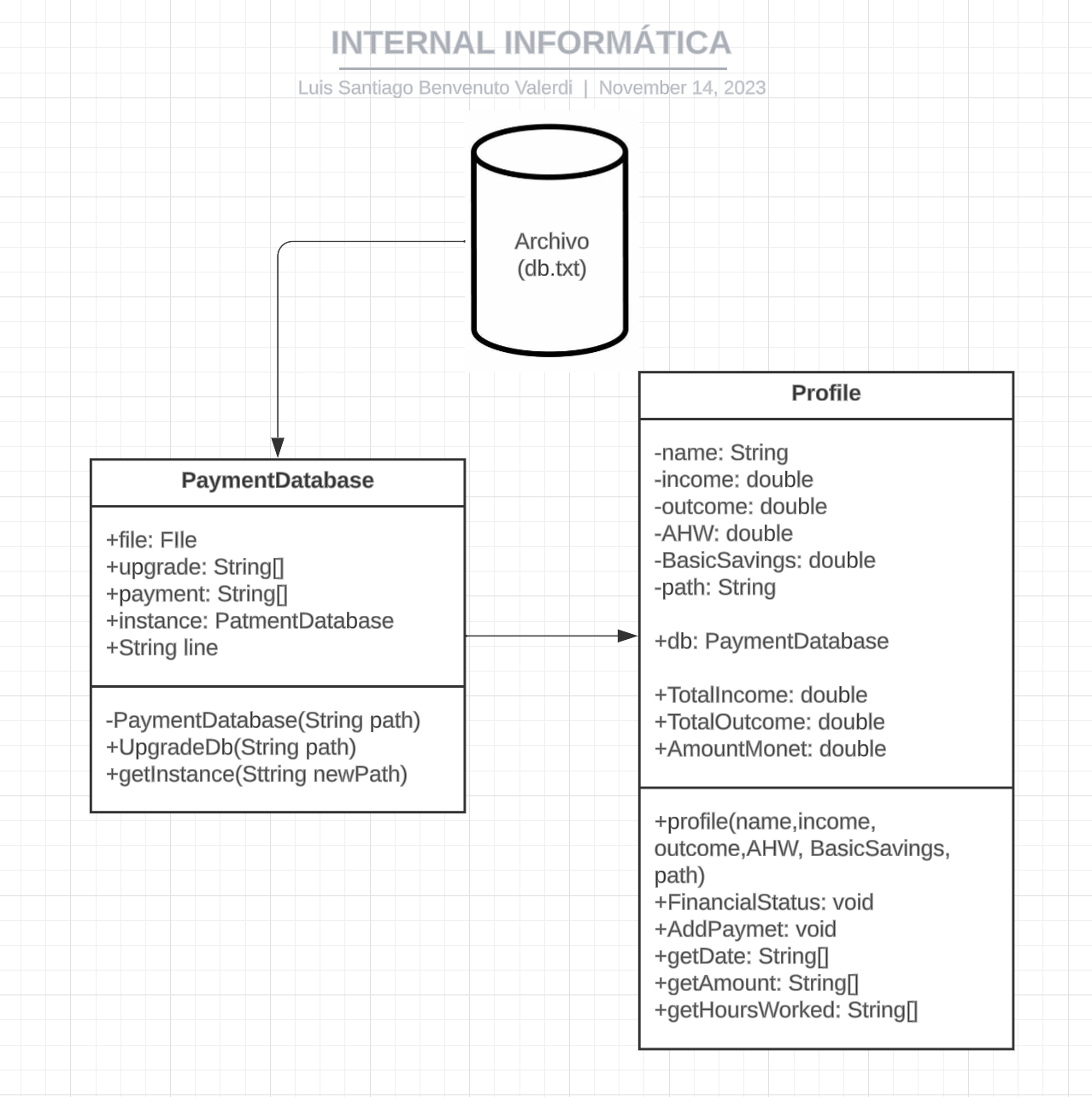
**Criterio B: Diseño**

**Informática**

**Luis Santiago Benvenuto Valerdi A01286400**

**Diagramas UML**

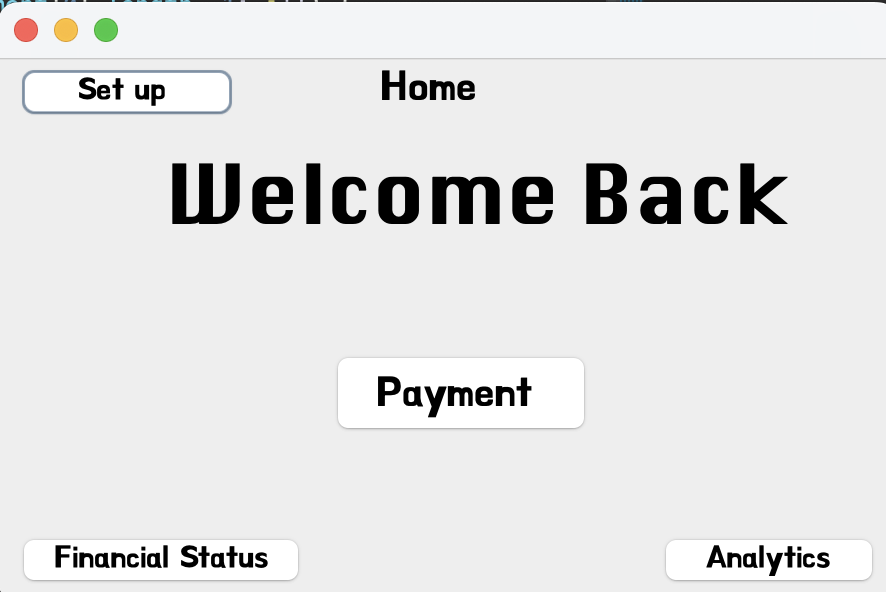
****

**Estructuras**

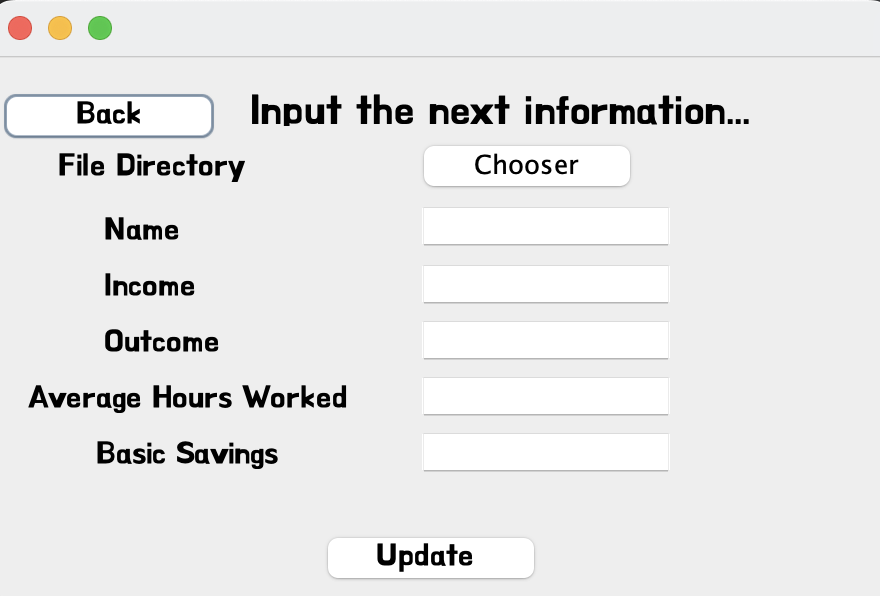
Java ofrece una flexibilidad y facilidad a la hora de utilizarlo con el fin de crear un software orientado a objetos, en el que se busca utilizar características propias de este tipo de programación como lo es la herencia y el polimorfismo. Primeramente, Java posee una librería para leer y escribir en archivos, por lo que claramente es muy fácil implementar el uso de archivos con este lenguaje. Asimismo, se selecciona utilizar estructuras como matrices y arreglos para poder representar y organizar claramente los datos recibidos por el usuario. Todo esto con el fin de mantener un orden y que el sistema se ejecute de manera eficaz. En segundo lugar, se consideró aspectos de validación y verificación, esto con el fin de evitar que el usuario introduzca datos erróneos y que perjudiquen la lógica en el código. Finalmente, el manejo de objetos va a ser importante para lograr tener varios perfiles a la vez y que se puedan modificar los datos de diferentes usuarios al mismo tiempo.

**Pantallas**

**HOME**

****

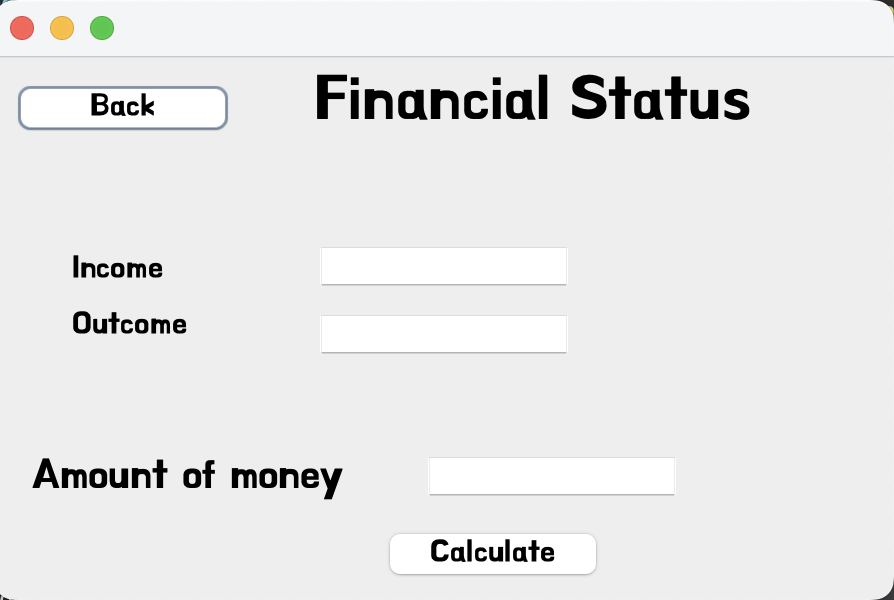
**SET UP**

****

**Financial Status**

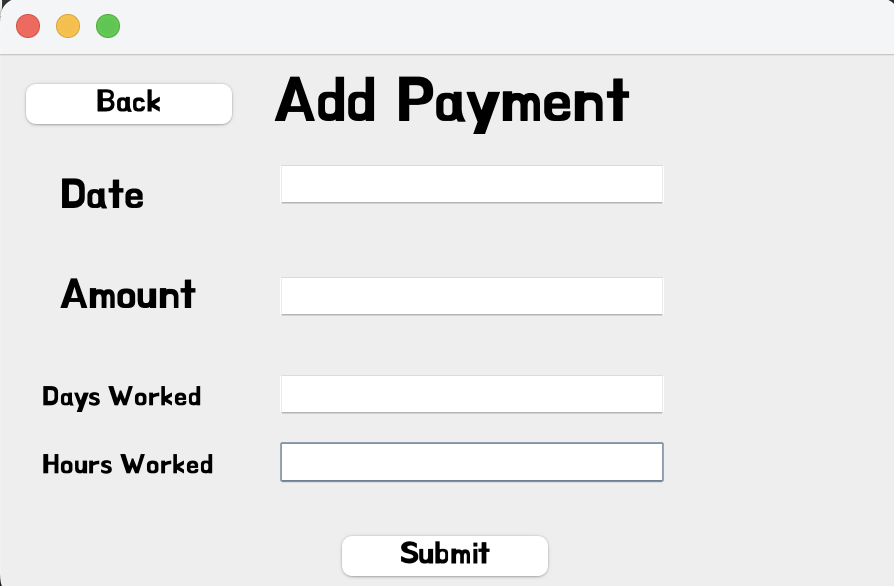
En esta pantalla será la principal, la que aparecerá cuando se abra la aplicación. Asimismo, se dará bienvenida y se le dará las diferentes opciones para acceder a las diferentes pantallas.

Esta pantalla tiene el propósito de generar un nuevo perfil con los datos que se introduzcan así tomando en cuenta los datos básicos de un usuario para poder identificarlo de manera fácil.

****

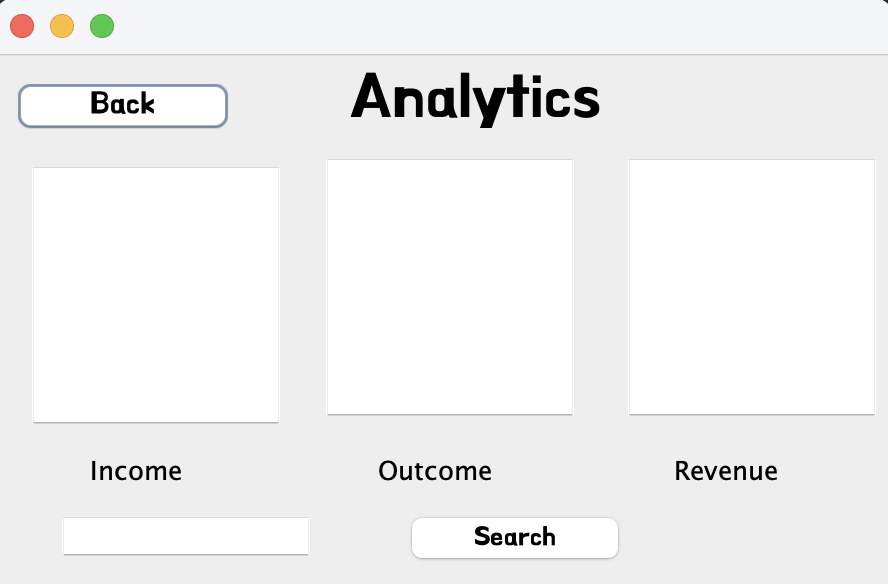
Esta pantalla muestra la suma de todos los pagos que ha recibido una persona, todos los egresos que pudo haber tenido si mantiene constante sus egresos y la cantidad de dinero que ganó durante todo este periodo que es la diferencia entre lo que ganó y lo que perdió.

**Add Payment**

****

**Analytics**

Esta pantalla se encarga de agregar cualquier ingreso que haya tendio el usuario así como identificar y procesar la cantidad de dinero que ganó por tiempo.

****

En esta pantalla se busca mostrar todos los datos que el programa leyó, así dando la posibilidad de comprobar si el registro es correcto. De igual manera, se da la posibilidad a que el usuario pueda introducir específicamente una cadena de string y buscarlo en todo el archivo para mostrar los otros datos de ese mismo.

**Plan de pruebas**

* El programa será capaz de agregar algún registro que se haya realizado.

|  | Aquí se puede ver el archivo con los registros actuales |
| --- | --- |
|  | Para agregar un nuevo registro basta con que se llenen cada uno de los siguientes campos mostrados en la ventana:  Una vez que se presiona  al darle clic al botón, |
|  | Aquí se muestra el archivo con el nuevo registro |

* El programa será capaz de analizar la información y mostrar el resultado de este análisis.

|  | Aquí se realiza el cálculo de todos los datos registrados. Este cálculo se realiza, primero se calcula el ingreso total durante todos los registros, así sumando todas las cantidades que se registraron como ingresos. Después, por medio de la cantidad de registros se multiplica por la cantidad de egresos promedio, así llegando a reducir los costos fijos que tiene el usuario. Finalmente se realiza una diferencia entre estos dos, para obtener la cantidad de dinero después de los gastos fijos del usuario. |
| --- | --- |

* El programa será capaz de buscar algún registro que el usuario quisiera ver entre todo el archivo donde se guardaron sus registros, de igual manera mostrarlos y dar la posibilidad de cambiarlos.

|  | Aquí se muestra toda la información que está en el archivo de registros del usuario. Mostrando todos los ingresos que este ha tenido. |
| --- | --- |
|  | Aquí, se introduce el dato por el que se quisiera analizar y buscar el registro. Con esto se pude encontrar fácilmente toda la información, con solo sabe alguno de los tres datos del registro |
|  | Finalmente, aquí se encuentra el registro y se le muestra al usuario los datos que se obtuvieron, así mostrando el registro que el usuario estaba buscando y toda la información de este mismo |

* El programa será capaz de leer archivos y sobreescribirlos

|  | Aquí se puede ver el archivo con los registros actuales. Este archivo es leído por el programa |
| --- | --- |
|  | Aquí se observa como ya se leyó todos los datos y los acomodó de manera llamativa, para que el usuario pueda ver todos los registros que ha hecho en el programa. |
|  | Con esta pantalla es como se sobreescribe el archivo actual. Introduciendo los datos de fecha, cantidad, horas trabajadas y días trabajados es como se puede agregar esta información al registro. |
|  | Aquí se muestra el archivo con el nuevo registro, con esto se demuestra que el archivo se sobrescribe y fue alterado, lo que demuestra que sí se pueden agregar registros a la aplicación. |