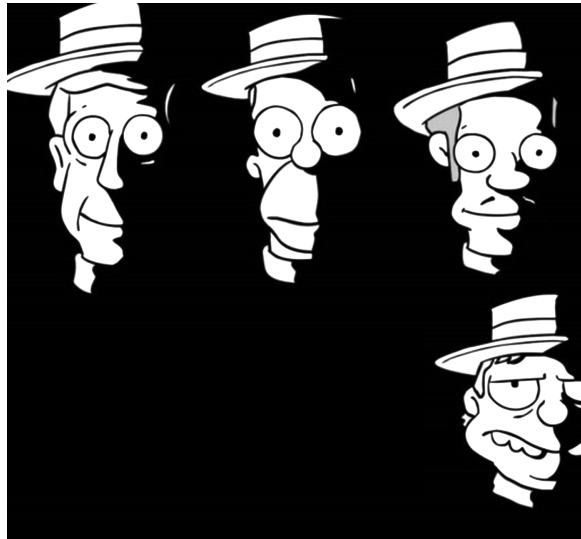


Los Borbotones

Llavero Digital

Documento de Arquitectura



Autor(es):

Lorenzo, Facundo

Ojeda, Gastón

Riba, Franco

Roig, Patricio

Versión actual del documento: 1.0

Fecha:23/04/2022

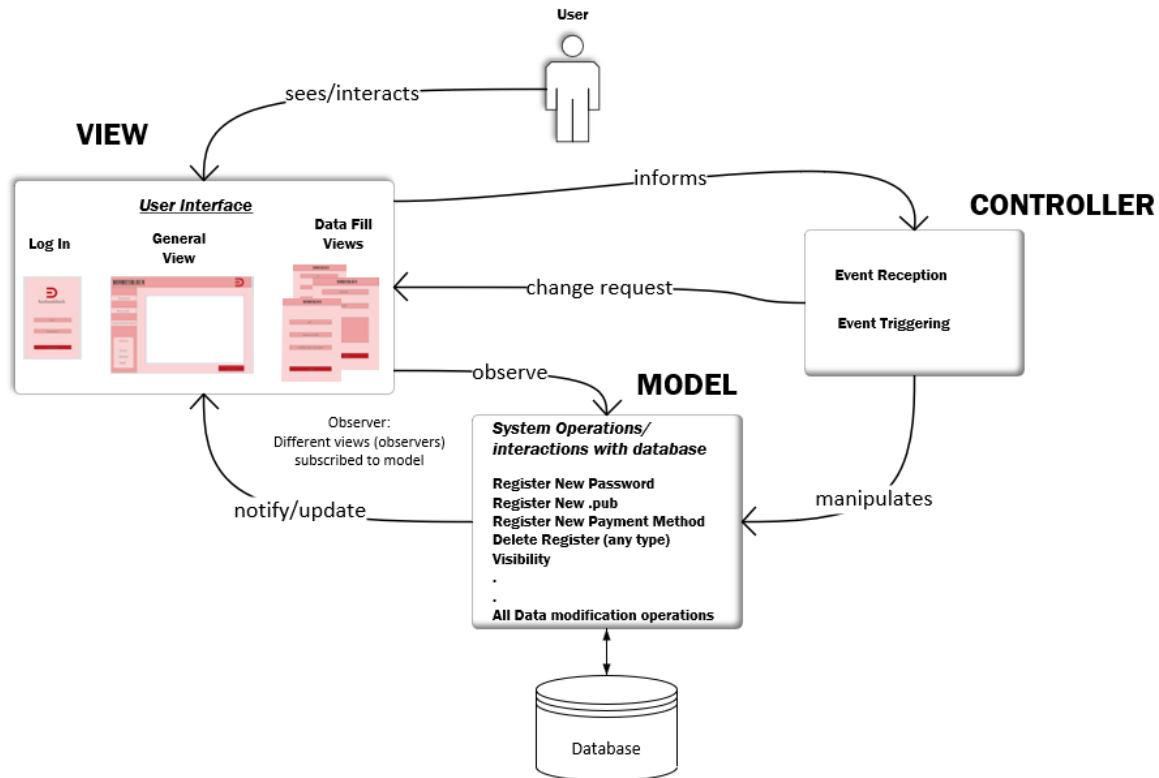
Índice

| | |
|---|----------|
| Historial de cambios | 1 |
| Diagrama de Arquitectura General | 2 |
| Justificación del Patrón de Arquitectura Utilizado | 2 |
| Diagrama de componentes | 4 |

Historial de cambios

| Versión | Fecha | Cambios |
|---------|------------|------------------------|
| 0.0 | 10/06/2022 | Documentación inicial. |

1. Diagrama de Arquitectura General



2. Justificación del Patrón de Arquitectura Utilizado

El proyecto implementa el patrón de arquitectura MVC (Model View Controller). En el diagrama anterior, observamos los módulos que componen este patrón, el modelo, la vista, el controlador y las relaciones entre ellos.

Este patrón, separa la lógica de la aplicación y los datos, de la visualización de los mismos. Esta separación en módulos, nos permite un mayor desacople entre los componentes del sistema, y por consiguiente, una mayor mantenibilidad del mismo, permitiendo realizar cambios en cada uno de los módulos, sin necesidad de modificar los otros. Nos da la posibilidad de mostrar los mismos datos de diferentes maneras de forma sincronizada.

En primera instancia, necesitamos una forma de comunicarnos con el usuario de manera efectiva para que al momento de iniciar la aplicación se haga de manera sencilla, para cumplir con los requerimientos RFU 3.0.0 -

RFU 3.0.1 - RFU 3.0.2, por lo que la utilización de una interfaz gráfica (Vista) nos permite realizarlo.

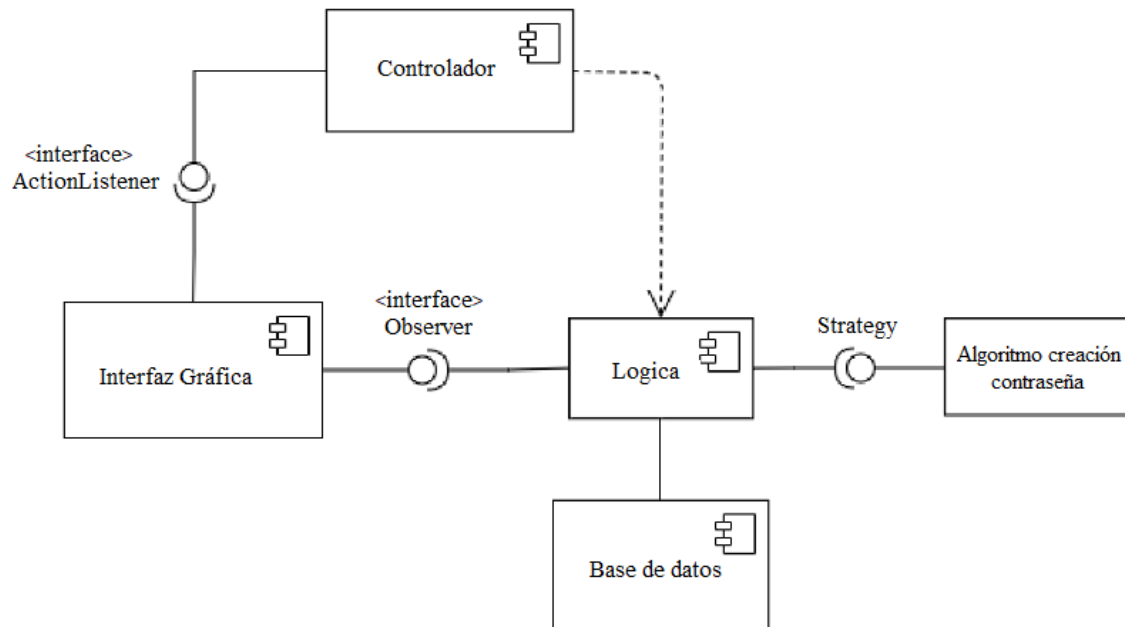
Luego necesitamos un módulo en el cual se desarrolle la lógica de la aplicación y contenga la información de la misma en todo momento, por lo que el módulo Modelo, nos brinda las características necesarias para llevarlo a cabo de manera correcta. En este módulo se realizan todas las acciones necesarias para dar comienzo a la aplicación por lo que unificarlas, produce que estas se hagan de manera rápida, en pos de cumplir el requerimiento RNFS 4.1.3. Finalmente, necesitamos un modelo para conectar Vista y Modelo, el cual será el Controlador. Este recibe información de la Vista, y se la comunica al Modelo, por lo que unifica estas tareas en un único módulo logrando un bajo tiempo de respuesta a las acciones realizadas por el usuario, lo que permite cumplir con el requerimiento RNFS 6.1.2 . La aplicación de este patrón a nuestro proyecto en particular, se da de la siguiente manera:

El usuario realiza una interacción con el sistema, la cual es recibida por el controlador. Esta interacción se realiza a través de clicks para determinar qué modo de uso quiere. A partir de esta interacción recibida por el controlador, éste se comunica con el modelo, el cual controla que se ejecute la lógica de la aplicación, determinando que es lo que quiere hacer el usuario, El modelo actualiza la vista, para dar información al usuario del estado actual en el que está.

En complemento de este patrón de arquitectura, se harán uso de los patrones de diseño Observer y Strategy. Cuya aplicación se explicará con mayor detalle en el Documento de Diseño.

Diagrama de componentes

A continuación se muestra el diagrama de componentes:



En este diagrama podemos observar los diferentes componentes que integran nuestro proyecto. Podemos observar los tres componentes básicos del patrón de arquitectura MVC, en donde la Lógica es el Modelo, la interfaz gráfica es la Vista y el Controlador es el Controlador. El componente Lógica, hace uso de la base de datos. En la Lógica debe establecerse el de uso que se quiera, Lógica hace uso de Algoritmo de creación de Contraseña utilizando el patrón Strategy. Además, Lógica, se comunica con la Interfaz Gráfica, a través de la aplicación de interfaces del patrón Observer. La comunicación entre el Controlador y la Interfaz Gráfica, se realiza a través de la interfaz ActionListener esperando un evento de la Interface ActionEvent de Java, en la cual el Controlador espera a las acciones realizadas por el usuario en la Interfaz Gráfica y las comunica a Lógica.