

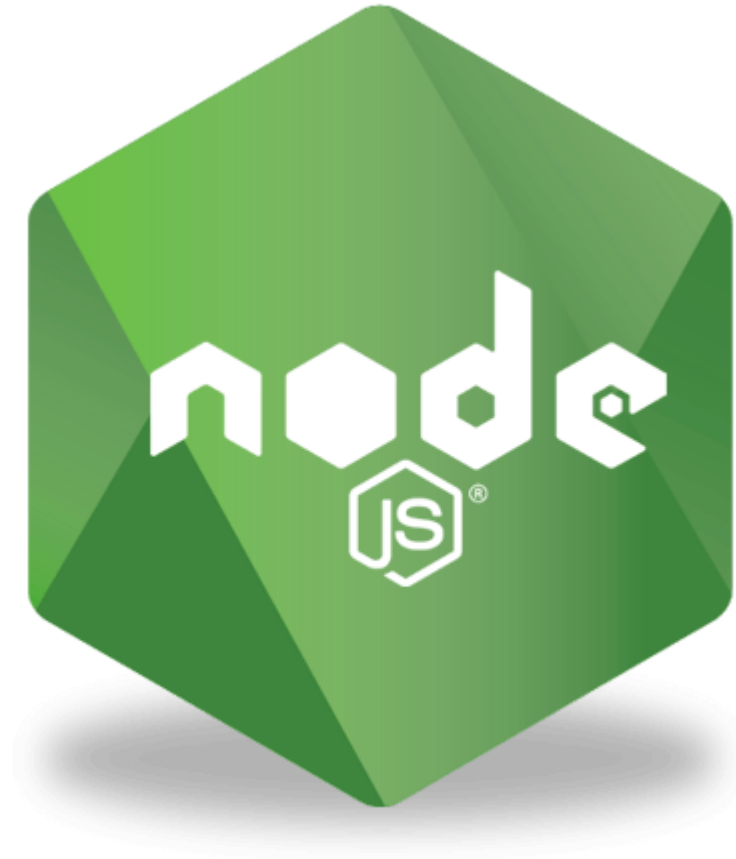
AUTENTICACIÓN DE USUARIO CON NODE.JS Y FIREBASE



INTRODUCCIÓN

¿Qué es Nodejs?

Es un entorno de ejecución de JavaScript multiplataforma. Es decir, que permite la ejecución de Nodejs fuera del navegador.



INTRODUCCIÓN



¿Qué es Firebase?

Firebase, es una plataforma para el desarrollo de aplicaciones web y móviles. Actualmente es desarrollada y mantenida por Google. Dispone de varias funcionalidades pero en el proyecto solo hace uso de Firestore.

¿Qué es Firestore?

Firestore, es una de las dos bases de datos que componen Firebase. Es la mas moderna y la que tiene una mayor disponibilidad.

INTRODUCCIÓN

¿Qué es TailwindCSS?

Es una librería css que nos provee de clases predefinidas, su principal reclamo frente a sus competidores, es la mayor personalización de las paginas. A diferencia de otros como Bootstrap.



ORIGEN DEL PROYECTO

Las decisiones de por que usar Nodejs,
Firebase o TailwindCSS

Nodejs

El uso de JavaScript en el servidor y el cliente.

La librería Express de Nodejs que permite un desarrollo mas rápido.

Unificar el servidor y el lenguaje de programación en uno

Firebase

Uso de una base de datos NoSQL

Evitar el mantenimiento de la base de Datos.

TailwindCSS

Evitar el uso de grandes ficheros CSS

Uso de una librería CSS con clases predefinidas.

El poco tiempo disponible para dedicar a los estilos

IMPLEMENTACIÓN

El manual de instalación se encuentra en el Readme del GitHub del proyecto

Tanto para desarrollo como para explotación, solo es necesario Nodejs y NPM, además de la clave privada de Firebase. Aunque se recomiendan otras aplicaciones a mayores



FUNCIONAMIENTO APLICACIÓN

Diagrama de casos de uso

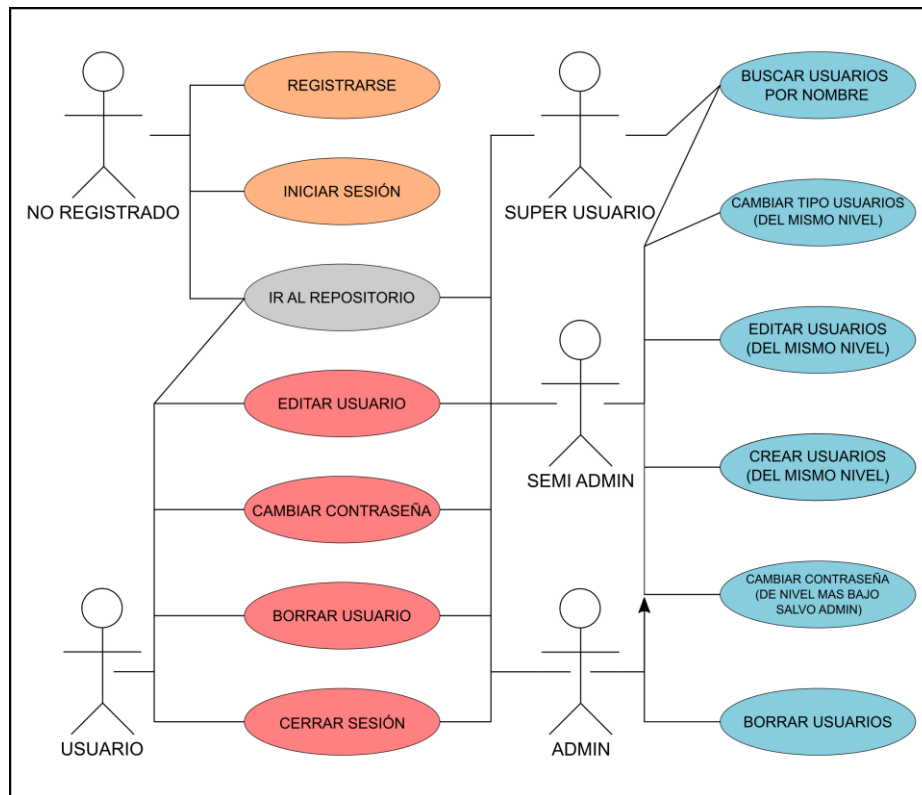
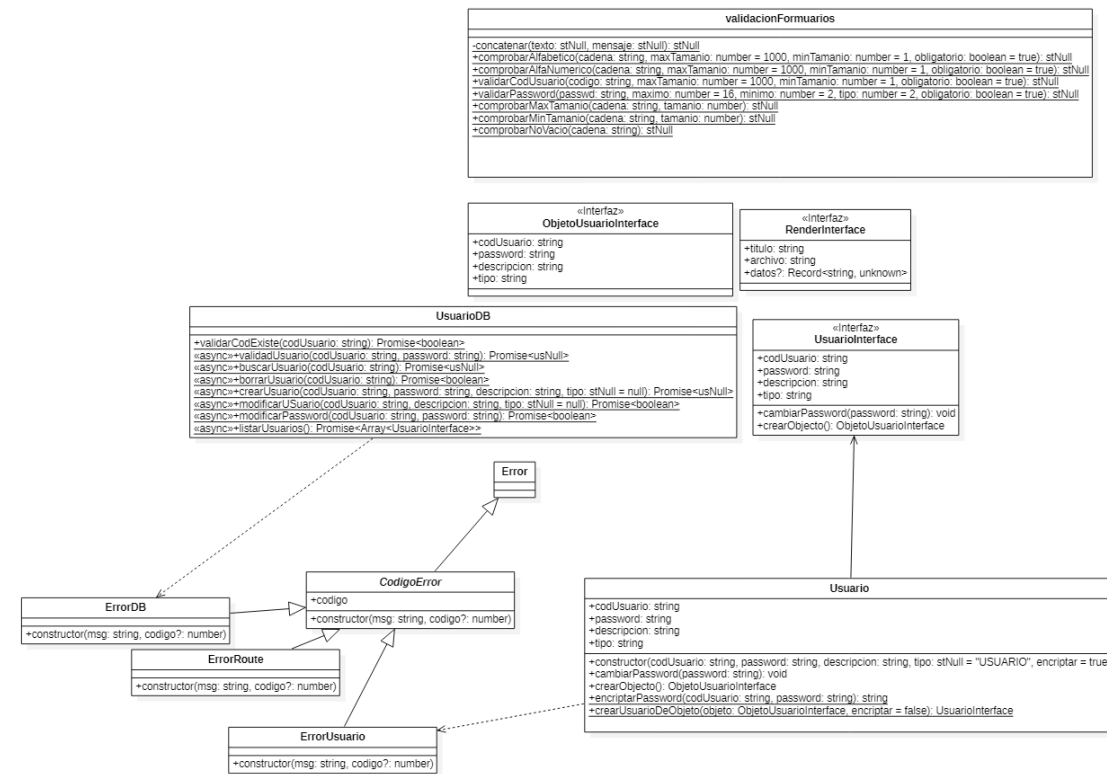


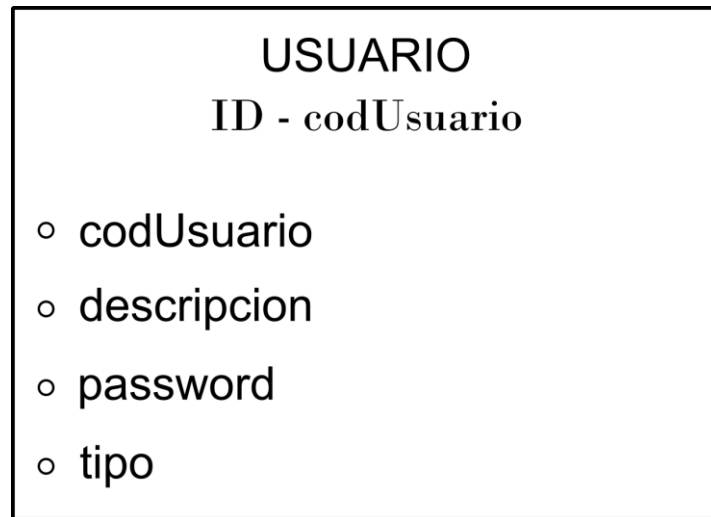
Diagrama de clases



Ciertas funcionalidades de los administradores no, han sido terminadas. Al ser objetivos opcionales

FUNCIONAMIENTO APLICACIÓN

Modelo físico de datos



Los datos están almacenados en documentos. Estos documentos necesitan un identificador único. Ese identificador es el mismo que el código del usuario

Estructura de almacenamiento

```
/dist
/node_modules
/docs
/src
  /config
  /controller
  /core
  /errors
  /interfaces
  /model
  /public
  /routes
  /scripts
  /views
index.ts
```


EXPRESS Y TYPESCRIPT

¿Por qué Express?

Aunque Nodejs por defecto, proporciona un modulo para la creación de un servidor, este puede llegar a ser complicado al inicio. Express es una librería considerada casi por defecto para Node, su amplio uso ha permitido el desarrollo de múltiples funciones, tanto por sus creadores como por otros creadores.

¿Por qué TypeScript?

Con el auge de JavaScript, surgió el problema de que no era tipado, ni orientado a objetos (las clases de ECMAScript 6 son el llamado Azúcar sintáctico). TypeScript, introdujo clases, tipos, interfaces, clases abstractas, entre otros. Haciendo el desarrollo de un programa JavaScript mas organizado, siendo TypeScript altamente recomendado para proyectos grandes. TypeScript necesita compilar su código a JavaScript y es usado por defecto en Angular y Deno (similar a Nodejs, desarrollado por la misma persona).

CONCLUSIÓN

El uso Nodejs y Express me han permitido desarrollar una arquitectura distinta, aunque intentado imitar la que usábamos en clase, además de entender de forma mas clara como funcionan las peticiones a un servidor y como son procesadas por este.

Firestore, me ha servido como contraparte a los sistemas gestores de datos SQL. Y además de haberlo usado en un proyecto mayor.