ShuttleGo
Trabajo de Fin de Grado

MEMORIA DE TRABAJO DE FIN DE GRADO

Carlos Castellanos Mateo
Víctor Chamizo Rodríguez

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

fecha ??

Documento maquetado con TEXIS v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

ShuttleGo
Trabajo de Fin de Grado

Memoria destinada a comprender el Trabajo de Fin de Grado

Carlos Castellanos Mateo
Víctor Chamizo Rodríguez

Dirigida por el Doctor

Antonio Sarasa Cabezuelo

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

fecha ??

Resumen

...

...

Aqui va el resumen

Índice

Índice de figuras

Índice de Tablas

Capítulo 1

Introducción

Resumen:

1.1. Introducción

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 2

Motivación

Resumen:

2.1. Introducción

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 3

Estado de arte

Resumen:

3.1. Introducción

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 4

Arquitectura

4.1. Introducción

En este capítulo se explicará la arquitectura de toda la aplicación en detalle.
En la primera sección se explicará la arquitectura general de toda la aplicación: que papeles básicos juegan el cliente y el servidor dentro de nuestra aplicación, mientras que en los siguientes apartados se describirá de forma más detallada la arquitectura de cada uno.

4.2. Arquitectura general de la aplicación

La arquitectura de este proyecto sigue el patrón típico de cliente-servidor, donde el cliente se encarga de la interacción del sistema con el usuario, mientras que la función del servidor consiste en asegurarse de que se cumplan todas las reglas de negocio y de interactuar con la base de datos. En muchos casos, el lado del cliente no permite introducir datos erróneos con el objetivo de mejorar la experiencia de usuario, aun así el servidor vuelve a comprobar la validez de estos datos. Además el cliente es también el encargado de hacer de intermediario con la API que gestiona los mapas, esto es debido a que queremos evitar sobrecargar el servidor con demasiadas peticiones y aunque hubiéramos querido, el plan gratuito de Firebase no permite conectar el servidor con servicios externos a Google. El diálogo entre cliente y servidor se realiza mediante peticiones https. Con todo esto, la aplicación tendría la estructura básica de la figura 4.1.

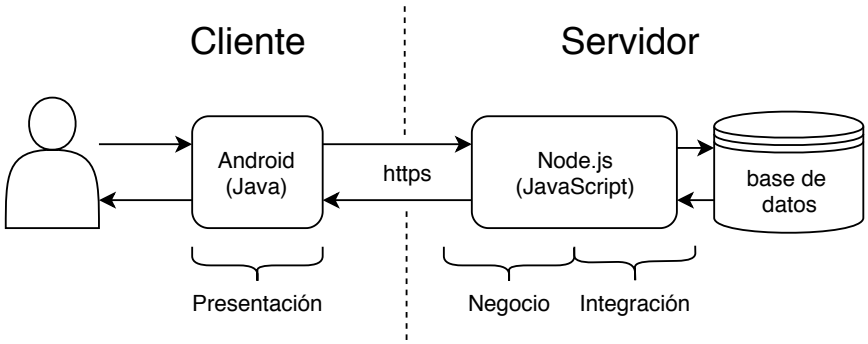

Figura 4.1: Arquitectura general de la aplicación.

4.3. Arquitectura del cliente

4.4. Arquitectura del servidor

El servidor tiene una arquitectura multicapa formada por tres partes bien delimitadas:

- Controlador frontal: Representado por el archivo ïndex.js". Es el encargado de recibir las peticiones. Además también realiza pequeñas validaciones de datos que están estrechamente ligados al formato de las peticiones y que podrían suponer un error de ejecución (ej: comprobar que no lleguen los datos vacíos) y validaciones de autorización (ej: comprobar que el usuario que ha mandado la petición de eliminar un origen es un administrador);

- Capa de negocio: Formada por todos los servicios de aplicación ubicados en la carpeta "business". El objetivo de esta capa es asegurarse de que se cumplen las diferentes reglas de negocio (ej: No se pueden añadir más pasajeros a un viaje si están todas las plazas ocupadas).

- Capa de integración: Formada por los "Data Access Objects"(DAOs) de la carpeta "dataAccess". Esta capa se encarga de dar acceso al sistema a la base de datos mediante operaciones simples (insertar, borrar, modificar y leer). Además esta capa contiene un archivo "database.js"que se encarga de realizar la configuración necesaria para hacer las peticiones a la base de datos.

Para el manejo de errores tenemos un archivo .^{er}rors.js", donde se encuentran todos los errores posibles que le pueden llegar al cliente junto a una descripción de cual puede ser la causa. Para solucionar el problema de la asincronía a la hora de interactuar con la base de datos, utilizamos promesas, por lo

que todas las funciones devuelven siempre una promesa.
Una vez que el cliente ha enviado una petición, el servidor se realiza los clientes pasos:

1. Se ejecuta la función del controlador asociada a la petición, donde primeramente se realizan comprobaciones de autorización y validación de datos, si no ha ocurrido ningún error, se llamará a una función de algún servicio de aplicación.

2. El servicio de aplicación lleva a cabo las reglas de negocio llamando a las funciones de los DAOs, asegurándose de que todos los datos sean coherentes y de que se cumplan todas las restricciones.

3. Los DAOs interactúan directamente con la base de datos obteniendo, insertando, modificando y borrando información.

Esta separación en capas permite que la lógica de negocio no dependa de la plataforma en la que se esta desarrollando: Si queremos cambiar el servicio que aloja el código del servidor solo habría que hacer cambios en el controlador, mientras que si cambiamos la forma en la que los datos son almacenados, solo habría que cambiar los DAOs. En el siguiente capítulo veremos qué servicios hemos usado para desarrollar la aplicación y cómo.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 5

Tecnología empleada

Resumen:

5.1. Introducción

En este capítulo se explicarán las diferentes tecnologías usadas para llevar a cabo este proyecto. Mostraremos qué servicios externos hemos necesitado y cómo hemos usado sus APIs para integrarlos en nuestra aplicación.

5.2. Cliente

5.3. Servidor

El servidor se encuentra alojado en Firebase, el cual permite el desarrollo del proyecto en Node.js mediante varios paquetes. Para desplegar el código del servidor en Firebase utilizamos el paquete firebase-functions, mientras que para el almacenamiento persistente de datos tenemos el paquete firebase-admin. Mediante este paquete, Firebase nos ofrece un servicio, Cloud Firestore, el cual nos permite salvar los datos en una base de datos NoSQL.

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Capítulo 6

Casos de Uso

Resumen:

6.1. introducción

...

Notas bibliográficas

Citamos algo para que aparezca en la bibliografía... (Bautista et al., 1998)

Y también ponemos el acrónimo CVS para que no cruja.

Ten en cuenta que si no quieres acrónimos (o no quieres que te falle la compilación en "release" mientras no tengas ninguno) basta con que no definas la constante `\acronimosEnRelease` (en `config.tex`).

En el próximo capítulo...

...

Parte I

Apéndices

Apéndice A

Así se hizo...

...

...

Resumen: ...

A.1. Introducción

...

Bibliografía

Y así, del mucho leer y del poco dormir,
se le secó el celebro de manera que vino
a perder el juicio.

Miguel de Cervantes Saavedra

Bautista, T., Oetiker, T., Partl, H., Hyna, I. y Schlegl, E. *Una Descripción de LaTeX 2ε*. Versión electrónica, 1998.

–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes

–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.

Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes