

Aplicación móvil para reserva de tours nacionales

Manual Técnico

Versión: 1.0

Elaborado por:

- Mario Andrés Guzmán

ÍNDICE DE CONTENIDO

MANUAL TÉCNICO	3
Propósito	3
Conocimientos necesarios.....	3
Requerimientos técnicos	3
PLATAFORMA WEB Y MÓVIL	4
Desarrollo.....	4
Estructura del proyecto	4
Variables de conexión	5
Inicio	6
Requisitos.....	6
Descargar.....	6
Ir a la carpeta del proyecto utilizando la consola de Node JS o CMD de Windows.....	6
Instalación	7
Database	7
Ejemplo	7
Ejecutar Servidor Web	7
Ejecutar Proyecto Web Angular	7
Ejecutar Proyecto móvil IonicFramework	8
Ejecutar Server DialogFlow	8
GLOSARIO DE TÉRMINOS	9

ÍNDICE DE FIGURAS

Figura 1: Estructura del proyecto	4
Figura 2: Directorio principal del proyecto.....	5

MANUAL TÉCNICO

Propósito

El presente manual tiene como finalidad ser una guía básica de operación y configuración del sistema; permitiendo al lector del mismo adquirir las destrezas y conocimientos indispensables para una operación adecuada del sistema, y ser una herramienta de consulta de primera mano a la cual puede recurrir el desarrollador en cualquier momento.

Conocimientos necesarios

- ✓ Conocer previamente el funcionamiento de un computador.
- ✓ La configuración de la red, para saber determinar el desempeño del sistema con otros sistemas existente.
- ✓ Los principios básicos de administración y manejo de una reserva turística.
- ✓ Otras habilidades inherentes a su destreza y agilidad para sacar mayor ventaja de la información que se encuentra en la base de datos que maneja el sistema.

Requerimientos técnicos

Requerimientos mínimos de hardware

- ✓ Procesador: Core Dúo
- ✓ Memoria RAM: Mínimo: 2 Gigabytes (GB)
- ✓ Disco Duro: 120Gb.

Requerimientos mínimos de software

- ✓ Privilegios de administrador
- ✓ Sistema Operativo: Windows o Linux

PLATAFORMA WEB Y MÓVIL

Desarrollo

La plataforma web se encuentra desarrollada con el lenguaje de programación TypeScript y como framework principal Angular CLI, el servidor se encuentra desarrollado con NodeJS y el gestor de base de datos utilizado es MySQL Server.

Estructura del proyecto

La estructura del directorio de la plataforma web es la siguiente:

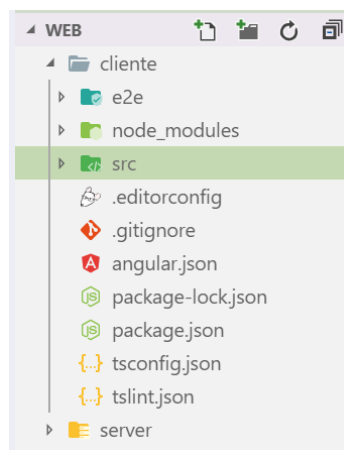


Figura 1: Estructura del proyecto

- ✓ **e2e**: Esta carpeta, cuyo nombre es "end to end", contiene una serie de ficheros que se encargaran de realizar test automáticos, como si un usuario real interactuara con nuestra app.
- ✓ **node_modules**: Es la carpeta que contiene todas nuestras dependencias de nuestro proyecto.
- ✓ **.editorconfig**: Configuración de nuestro editor de código.
- ✓ **.gitignore**: Los archivos o carpetas que debe ignorar git a la hora de añadirlos al repositorio.
- ✓ **angular.json**: Archivo json que contiene la configuración de angular. Este archivo incluye rutas, nombres, versiones, etc.
- ✓ **package.json**: Configuración de nuestra aplicación. Este archivo contiene el

nombre de nuestra app, la versión, todas las dependencias necesarias para su correcta ejecución, etc.

- ✓ **tsconfig.json**: Este archivo contiene la configuración de TypeScript.
- ✓ **tslint.json**: Se usa para mantenibilidad y sostenibilidad de nuestro código.

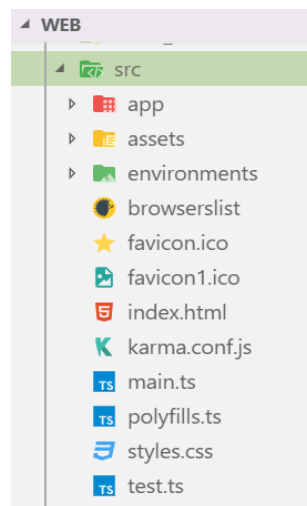


Figura 2: Directorio principal del proyecto

En este directorio se encuentra la carpeta **app**, que contiene el código de nuestra aplicación. También observamos la carpeta **assets**, que contiene los archivos estáticos, **environments**, contiene las variables de entorno, y muchos otros archivos que forman parte de nuestra aplicación.

Si accedemos a la carpeta **app**, podemos encontrar nuestro **componente principal** (*app.component.ts*), junto a su archivo de estilos css y su template html con el mismo nombre. También podemos encontrar el **módulo principal**, que envuelve nuestro componente.

Variables de conexión

La API de Google Maps se encuentra en esta ruta:

\web\cliente\src\app\app.module.ts

La configuración acerca de la autenticación de usuarios se encuentra en esta ruta:

\easyTour\src\app\firebase.info.ts

La configuración de PayPal se encuentra en esta ruta, en la variable paypalConfig se cambia la llave de desarrollo por la llave de producción.

\easyTour\src\pages\checkout-trip\checkout-trip.ts

La configuración del chatbot está en el server dialogflow.js en la siguiente ruta:

\easyTour\dialogflow.js

Inicio

Estas instrucciones le permitirán obtener una copia del proyecto en funcionamiento en su máquina local para fines de desarrollo y pruebas. Sistema operativo utilizado: Windows 10 Home x64 bits.

Requisitos

Descargar

- [Node JS](#) - Entorno de ejecución.
- [MySQL Community Server versión 8.0.15 o superior](#) - Base de datos
- [MySQL Workbench versión 8.0.15 o superior](#) - Gestor de Base de datos
- [Visual Studio Code](#) - Editor de código

¿Qué cosas necesita para instalar el software y cómo instalarlo?

Ir a la carpeta del proyecto utilizando la consola de Node JS o CMD de Windows

```
* Node JS      versión 10.13.0
* Angular Cli  npm install -g @angular/cli
* Ionic Framework npm install -g ionic
* Express JS   npm install express --save
* TypeScript   npm install -g typescript --save
* MySQL
```

NOTA: De preferencia utiliza el editor de código Visual Studio Code, para abrir el proyecto y ejecutar todos los pasos con su terminal integrada.

Instalación

Pasos que le indican cómo instalar y ejecutar el entorno de desarrollo:

Database

La configuración de la base de datos se encuentra en:

```
$ cd /tesis/web/server/src/keys.ts
```

Ejemplo

```
// code!!
export default {
  database: {
    host: 'localhost',
    user: 'root',
    password: 'root',
    database: 'reserva_db'
  }
}
```

Una vez realizado cualquier cambio en el Server para que guarde los cambios debera ejecutar:

```
$ cd/tesis/web/server
$ npm run build
$ npm run dev
```

El backup de la base de datos se llama reserva_db se encuentra en la carpeta /Database-Backup y se debe importar utilizando MySQL Workbench

Ejecutar Servidor Web

```
# Ir a la carpeta del proyecto Server Web utilizando la consola de Node JS o CMD de Windows
$ cd /tesis/web/server
```

```
# Instalar dependencias
```

```
$ npm install
$ tsc
$ node build/index.js
$ npm run build
$ npm run dev
```

Ejecutar Proyecto Web Angular

```
# Ir a la carpeta del proyecto Web Angular Cli
$ cd /tesis/web/client
# Instalar dependencias
$ npm install
$ ng serve -o
En el navegador de internet escriba http://localhost:4200
```


Ejecutar Proyecto móvil IonicFramework

```
# Ir a la carpeta del proyecto IonicFramework
$ cd tesis/easyTour
$ npm install
$ ionic serve --lab
En el navegador de internet escriba http://localhost:8200
**Deberá registrarse una cuenta para poder ingresar al sistema!
```

Ejecutar Server DialogFlow

```
# Ir a la carpeta del proyecto IonicFramework para ejecutar para ejecutar servidor de DialogFlow
$ cd tesis/easyTour
$ node server.js
```

GLOSARIO DE TÉRMINOS

- ✓ **TypeScript:** TypeScript es un lenguaje de programación de código abierto con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes.
- ✓ **Framework:** Es marco de trabajo, es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado.
- ✓ **Angular CLI:** Angular CLI es la herramienta adecuada para generar aplicaciones Angular. Juntos son imbatibles en cuanto a velocidad en desarrollo y a potencia en ejecución.
- ✓ **NodeJS:** Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript
- ✓ **Server:** Proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negociación y de acceso a los datos de las aplicaciones.
- ✓ **e2e:** Esta carpeta, cuyo nombre es "end to end", contiene una serie de ficheros que se encargaran de realizar test automáticos, como si un usuario real interactuara con nuestra app.
- ✓ **node_modules:** Es la carpeta que contiene todas nuestras dependencias de nuestro proyecto.
- ✓ **.editorconfig:** Configuración de nuestro editor de código.
- ✓ **.gitignore:** Los archivos o carpetas que debe ignorar git a la hora de añadirlos al repositorio.
- ✓ **angular.json:** Archivo json que contiene la configuración de angular. Este archivo incluye rutas, nombres, versiones, etc.
- ✓ **package.json:** Configuración de nuestra aplicación. Este archivo contiene el nombre de nuestra app, la versión, todas las dependencias necesarias para su correcta ejecución, etc.
- ✓ **tsconfig.json:** Este archivo contiene la configuración de TypeScript.

- ✓ **tslint.json:** Es el linter de TypeScript, se usa para mantenibilidad y sostenibilidad de nuestro código.
- ✓ **assets:** Contiene los archivos estáticos, imágenes, audio, etc.
- ✓ **environments:** Contiene las variables de entorno, y muchos otros archivos que forman parte de nuestra aplicación.
- ✓ **Sistema operativo:** Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.
- ✓ **Database:** Base de datos; información que puede ser almacenada y procesada mediante un computador.
- ✓ **npm:** Es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript,
- ✓ **build:** Es la versión operativa de un producto software que incorpora un subconjunto de las funciones que se incluirán en el producto final. Generalmente se identifica por un número de compilación, en lugar de por un número de versión.
- ✓ **Ionic Framework:** Ionic es la plataforma que facilita la creación de aplicaciones móviles con tecnologías web para los programadores web. La plantilla de Ionic permite a los desarrolladores crear aplicaciones móviles en diferentes plataformas que pueden instalarse en teléfonos con Android e iOS.
- ✓ **DialogFlow:** DialogFlow es la herramienta de creación de chatbots capaz de entender el lenguaje natural.