

# Trabajo Final Sistemas Distribuidos

Luis Miguel Panadero Cerdá - Grupo ARA

Para este trabajo he decidido programar los servidores tanto de Vuelos, Hoteles y Coches de Alquiler, como el sistema de acreditación de usuarios en Node.js, usando express y accediendo a base de datos en MongoDBAtlas, basado en MongoDB.

Cada uno de estos tipos deberá correr en un puerto distinto de la máquina y con el 'host' como 0.0.0.0 si queremos acceder a sus funcionalidades desde cualquier dispositivo donde tengamos el cliente de la Agencia corriendo.

El cliente lo he desarrollado en Angular.js por su sencillez y simplicidad a la hora del diseño de la interfaz de usuario.

He de decir que este proyecto no he conseguido acabarlo, ya que las funcionalidades más importantes exigidas para este no están acabadas por falta de tiempo.

Procederé a explicar qué he conseguido hacer y cómo está estructurado el proyecto.

En primer lugar, cada funcionalidad está dispuesta en una carpeta representativa, teniendo tanto la agencia como todos los servidores en una distinta. Para comprobar la funcionalidad del proyecto, comencé creando un servidor que almacenara y accediese a la base de datos de Vuelos. Una vez tuve una APIRest completamente funcional para mi propósito, comencé con el cliente de la Agencia para obtener esa información y que se mostrara en pantalla correctamente.

Una vez tuve un cliente y un servidor, comencé a implementar un sistema de autenticación básico con usuarios y contraseñas, usando el servidor de Usuarios que me permitía acceder a la base de datos de Usuarios y registrar o acceder a ellos.

Si conseguía hacer que un usuario pudiera iniciar sesión y registrarse correctamente, lo cual no pude por temas de definición de variables en Typescript que no supe arreglar, podría haber añadido el resto de las funcionalidades del proyecto.

Aquí definiré lo que habría podido hacer:

- Añadir servidores de Hoteles
- Añadir servidores de Coches de Alquiler.
- Hashear contraseñas para que sean imposibles de robar.
- Implementar reconocimiento de qué servidor ofrece qué información.
- Un servidor diferente al resto para la Entidad Bancaria usando NodeJS y SocketIO, completamente securizado.
- Securizar el resto de los servidores para usar Https.
- Crear un “Carrito de Compra” donde el usuario pondría las ofertas que quiere contratar.
- Pedir a los servidores las ofertas deseadas por el usuario y marcarlas como reservadas en la base de datos.
- Interfaces visuales en los servidores para crear, editar y eliminar información de cada uno de ellos.

Mi proyecto necesita estas funcionalidades para considerarse funcional, y siento mucho no haberlo hecho, ya que estoy seguro de que con más tiempo habría podido tenerlo correctamente.

Para mantener funcionando los servidores se necesita acceder a la carpeta Root en un terminal y correr “**node index.js**”

Para acceder al cliente de la agencia se necesita acceder a la carpeta root en un terminal y correr “**npm start**”