

The logo for UNIR, featuring the letters 'UNIR' in white, bold, sans-serif font inside a blue arrow-shaped graphic pointing to the right. This graphic is part of a larger dark blue vertical bar on the left side of the page.

UNIR

Manual Técnico

Aplicación web de gestión de
Usuarios y Autos. (Softlutions)

An abstract graphic in the bottom left corner consisting of several thin, curved lines in dark blue and light grey, resembling stylized grass or reeds.

Mario Alberto Cortés Muñoz (2do semestre)
MASTER EN DIRECCIÓN E INGENIERÍA DE SITIOS WEB

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Contenido

Introducción	2
Objetivo	2
Construcción de la aplicación	2
Configuración del entorno de desarrollo	3
Desarrollo de la aplicación	9
Uso del código de la app	11
Conclusión.....	12
Referencias	12

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Introducción

Con un manual técnico podemos conocer más a fondo la forma como se desarrolló un sistema, las tecnologías que lo conforman, las librerías, utilidades y demás componentes que fueron necesarios para poder lograr que el sistema funcione como se esperaba.

Objetivo

La elaboración de este documento tiene como principal objetivo, presentar un informe lo más detallado posible del trabajo realizado, se explicarán las decisiones de diseño e implementación que se tomaron conforme se fue realizando la construcción de la aplicación.

Construcción de la aplicación

Como primer paso para iniciar con el desarrollo de la aplicación, se procedió con la descarga e instalación de las tecnologías que serían utilizadas para la construcción del sistema.

El primero fue el lenguaje en el que se escribiría el código, en este caso NodeJs.



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

#BlackLivesMatter

Node.js Certifications and Trainings are on sale, now until Dec. 8

Descargar para Windows (x64)

14.15.1 LTS

Recomendado para la mayoría

15.3.0 Actual

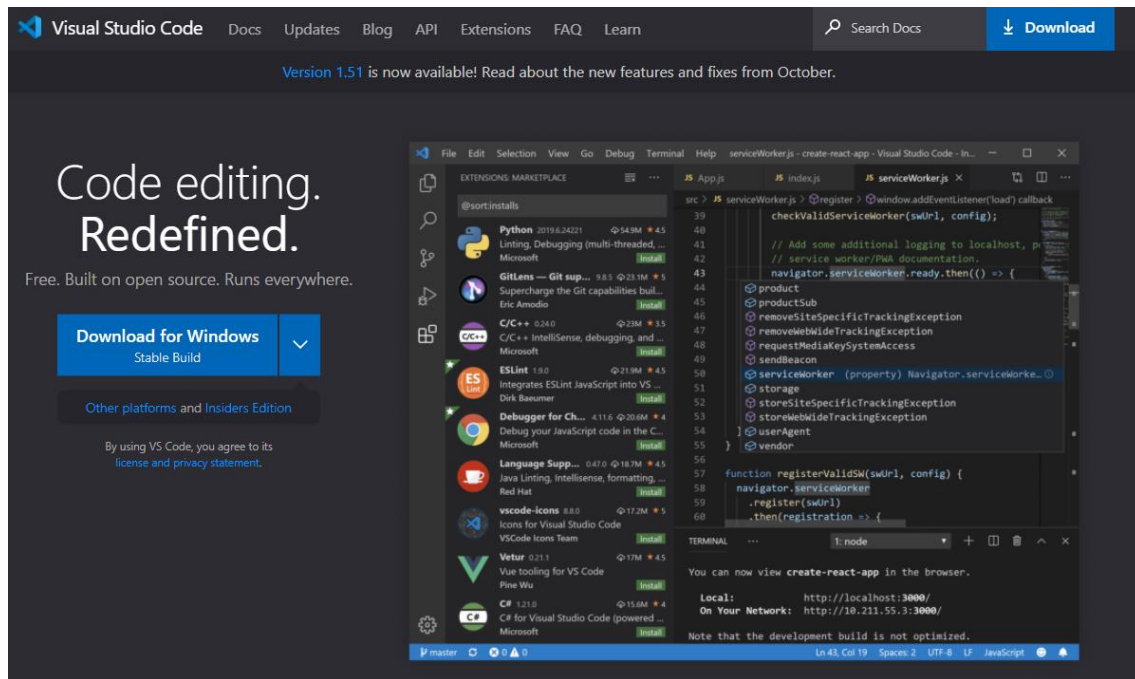
Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

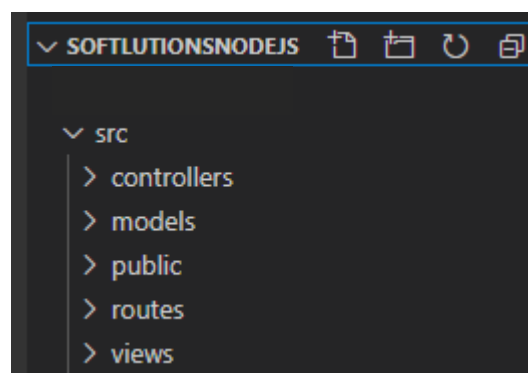
La segunda descarga fue el IDE Visual Studio Code que nos serviría como entorno de desarrollo.



Configuración del entorno de desarrollo

Con estas dos herramientas ya instaladas, dimos inicio al proyecto que sería un sistema para la creación, consulta, modificación y eliminación (CRUD) de dos entidades diferentes. (Usuarios y Autos). La aplicación se denominó como Softlutions.

Creamos la estructura del árbol de carpetas que debe tener el proyecto:



Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

- » **Src.** – Ésta carpeta es la carpeta raíz de la aplicación, en ella se encuentran contenidas las demás carpetas y archivos.
- » **Controllers.** – En esta carpeta están definidos los controladores de la aplicación.
- » **Models.** – En esta carpeta están definidos los modelos que tienen las tablas de las entidades guardadas en la base de datos.
- » **Public.** – En esta carpeta se encuentran todos los archivos que tienen acceso público en la aplicación como, por ejemplo, los archivos css de estilos, documentos, etc.
- » **Routes.** – En esta carpeta están definidas las rutas de acceso a las diferentes páginas de la aplicación web.
- » **Views.** – En esta carpeta están las vistas de la aplicación, lo que al final ve el usuario en su navegador.

Siguiente paso:

Desde la terminal del Visual Studio Code, usamos la instrucción `npm install <nombre del módulo> --save` para instalar los módulos necesarios para que la aplicación pueda funcionar.

```
// Express
const express = require('express');

// Módulo para el uso de Plantillas
const exphbs = require('express-handlebars');

// Habilita el uso del path.join para concatenación
const path = require('path');

// Para ver en la terminal las peticiones que llegan al servidor
const morgan = require('morgan');

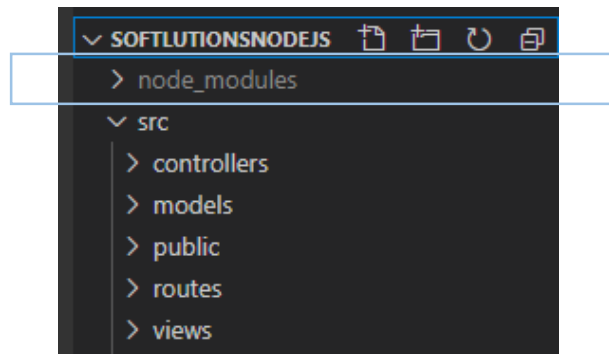
// Sobreescribe el verbo POST con los verbos DELETE y/o PUT para usarlos en los formularios
const methodOverride = require('method-override');

// Módulos para enviar mensajes entre las páginas del sitio
const flash = require('connect-flash');
const session = require('express-session');

// Módulo Mongoose como manejador de base de datos
const mongoose = require('mongoose');
```

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

La instalación de estos módulos crea en el árbol de carpetas una más que se llama `node_modules` que contiene todos los módulos anteriormente instalados.



Adicional se instalaron unas dependencias que sirvieron como apoyo durante el desarrollo de la aplicación.

```
"devDependencies": {
  "dotenv": "^8.2.0",
  "nodemon": "^2.0.6",
  "npm-check-updates": "^10.2.2"
}
```

- » **dotenv** se usa para leer un archivo con extensión “.env” en el que se pueden definir variables de entorno en vez de usar valores fijos.
- » **nodemon** está pendiente de los cambios realizados al código y los aplica sin necesidad de estar apagando e iniciando el servidor en cada cambio.
- » **npm-check-updates** está pendiente de las posibles actualizaciones que pudieran tener los módulos.

En el archivo package.json, que es donde se enlistan las dependencias y módulos instalados, se agregó al json lo siguiente:

```
"scripts": {
  "start": "node src/index.js",
  "dev": "nodemon src/index.js"
}
```

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

- » **start** indica la ruta que se ejecuta cuando la aplicación ya se encuentra en ambiente de producción.
- » **dev** indica la ruta que se ejecuta cuando la aplicación se encuentra en ambiente de desarrollo.

Para probarlas se ejecuta desde la terminal los siguientes comandos respectivamente:

```
npm start
npm run dev
```

Después de instalar todos los módulos y dependencias necesarias, probamos la conexión al servidor. En el archivo server.js tenemos el código que se encarga de escuchar las peticiones que le llegan al servidor:

```
// Servidor que escucha las peticiones
app.listen(app.get('port'), () => {
  console.log("Server on port ", app.get('port'));
});
```

Y para probarlo en el ambiente de desarrollo ejecutamos el siguiente comando:

```
npm run dev
```

La respuesta:

```
PS C:\Users\Mario Cortés\Desktop\softlutionsNodejs> npm run dev
```

```
> notes-app@1.0.0 dev C:\Users\Mario Cortés\Desktop\softlutionsNodejs
> nodemon src/index.js
```

```
[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
```

```
Server on port 4000
```

<- Aquí nos indica que el servidor ya está iniciado en el puerto 4000

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Siguiente paso:

Para la base de datos se utilizó el servicio en la nube de MongoDB Atlas, por lo que se procedió a crear una cuenta gratuita. Se optó por este servicio porque es mucho más visual y más entendible que la opción nativa de MongoDB.

Después de seguir una serie de pasos para la configuración del cluster*, como resultado se obtiene una cadena que se coloca dentro del código de conexión de Mongoose en el proyecto.

Connect to Cluster0

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

```
// Módulo Mongoose
const mongoose = require('mongoose');

// URI para la conexión a la base de datos
const MONGODB_URI = 'aquí va la url proporcionada por MongoDB Atlas';

// Conexión a la base de datos
mongoose.connect(MONGODB_URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
  useCreateIndex: true
})
.then(db => console.log("base de datos conectada a : ", db.connection.host))
.catch(err => console.error(err));
```

Una vez hecho esto, detenemos la ejecución del servidor (si está en ejecución) con el comando `ctrl + c` (dos veces) y probamos la conexión ejecutando nuevamente el comando desde la terminal de Visual Studio:

```
npm run dev
```

La respuesta:

```
PS C:\Users\Mario Cortés\Desktop\softlutionsNodejs> npm run dev

> notes-app@1.0.0 dev C:\Users\Mario Cortés\Desktop\softlutionsNodejs
> nodemon src/index.js

[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Server on port 4000

base de datos conectada a : cluster0-shard-00-01.bx0cv.mongodb.net
```

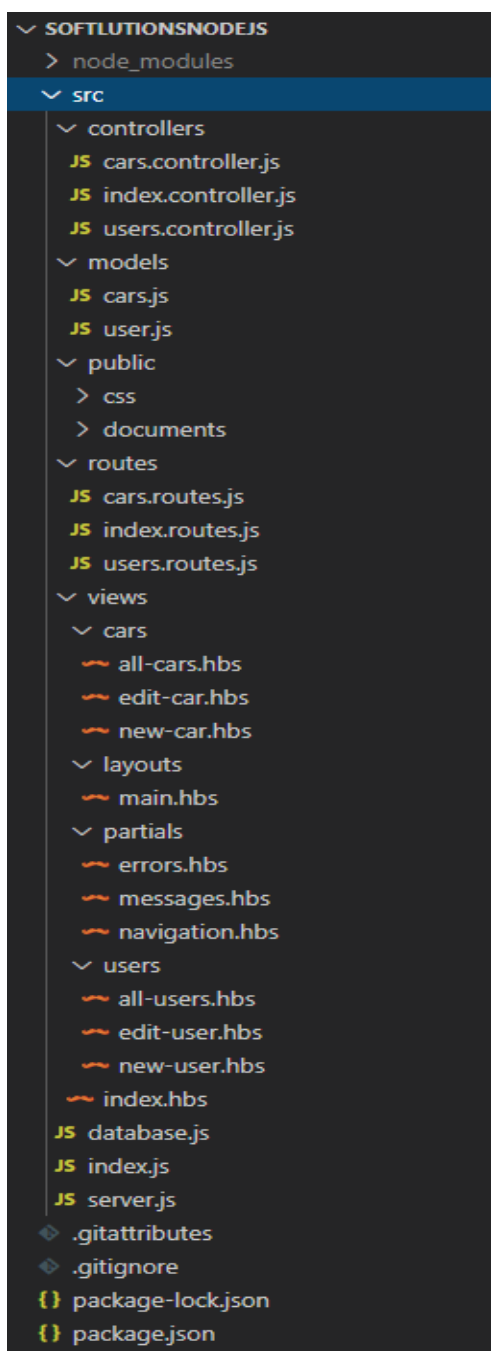
En ésta última línea nos indica que la base de datos se conectó correctamente al servicio en la nube del MongoDB Atlas. Ahora ya tenemos listo el servidor y la conexión a la base de datos.

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Desarrollo de la aplicación

La aplicación se desarrolló desde cero, iniciando con la descarga de las tecnologías, y posteriormente con la preparación y configuración del ambiente de desarrollo en general.

Al final, la estructura del proyecto y de las carpetas quedó de la siguiente manera:



Controllers:

Están los controladores que contienen las funciones que acceden a la base de datos, y que realizan las acciones del crud, tanto para el módulo de autos como para el de usuarios.

Models:

Aquí están definidas las estructuras que tienen las tablas de usuarios y de autos en la base de datos del MongoDB Atlas.

Public:

Archivos públicos

Routes:

Aquí están definidas las diferentes rutas que siguen las acciones de ambos módulos (usuarios y autos)

Views:

Contiene las vistas agrupadas.

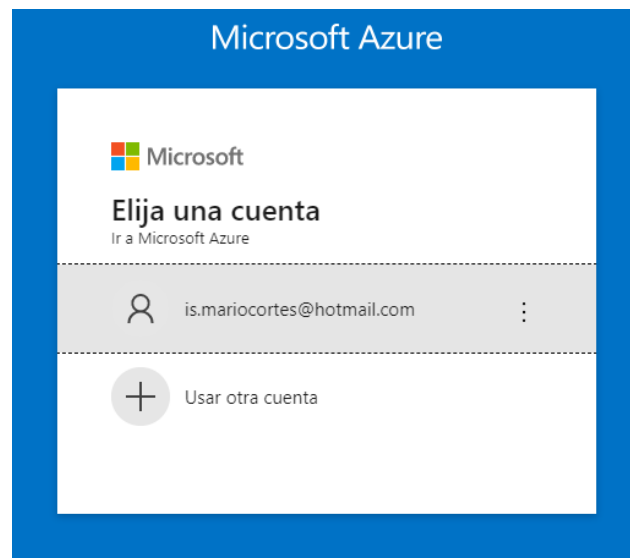
- » Cars: son las vistas para el módulo de autos.
- » Layouts: es la plantilla principal del sitio.
- » Partials: son secciones que se pueden incluir en la plantilla principal del sitio.
- » Users: son las vistas para el módulo de usuarios.

Fuera de éstas carpetas, pero siempre dentro de la carpeta src, están los archivos para la conexión a la bd y para el inicio del servidor.

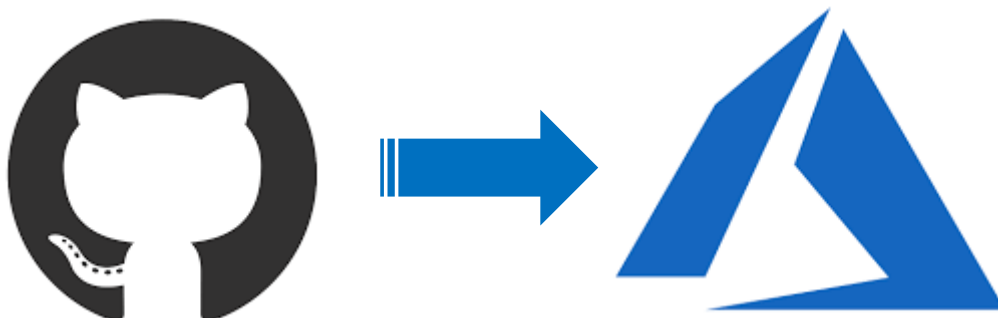
Por último está el archivo package.json que contiene la lista de módulos y dependencias instaladas en el proyecto.

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Ya concluido el desarrollo de la aplicación, necesitamos un espacio para desplegarlo. Para ello utilizamos un servicio en la nube de Microsoft Azure que nos provee de dicho servicio. En este caso no fue necesario crear una cuenta nueva ya que solicita la misma que se utiliza para acceder a todos los servicios de Microsoft como la del correo Outlook/Hotmail por ejemplo.



Habiendo accedido, y después de seguir una guía (video tutorial**) para la configuración del espacio en donde estaría la aplicación, obtuvimos como resultado un enlace que nos sirvió para configurar la importación del código desde github hacia Azure.



Finalmente obtenemos el enlace en donde se encuentra publicada la aplicación.
<https://softlutions.azurewebsites.net/>

Nota: La primera vez que accedes al sitio tarda un poco. (procesamiento de Azure)

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	



Uso del código de la app

Para usar/instalar el código de la aplicación, se puede clonar desde Github a la máquina local como sigue:

El comando para clonarlo desde el cmd de Windows es:

git clone https://github.com/ismariocortes/softlutionsNodejs

Después de clonarlo, abrirlo con el Visual Studio Code (recomendado), y desde la terminal del editor se ejecuta el comando:

npm i

Con este comando se instalan todos los módulos y dependencias que necesita el proyecto.

Cuando termine de instalar, ejecutar el comando:

npm run dev

¡Y eso es todo!

Asignatura	Datos del alumno	Fecha
Ingeniería y desarrollo en la web	Apellidos: Cortés Muñoz	6 de diciembre de 2020
	Nombre: Mario Alberto	

Conclusión

El desarrollo de ésta aplicación fue un verdadero reto ya que las tecnologías que se usaron son nuevas (para mí) y es la primera vez que las uso para desarrollar un proyecto.

Al principio fue complicado entender cómo funciona el código, las conexiones entre las tecnologías, pero con la ayuda de guías y tutoriales*** cada vez fue quedando más claro.

La parte que me costó más trabajo fue la configuración con Azure, realicé muchos intentos y no funcionaba, hasta que por fin encontré lo que estaba fallando y logré desplegar la aplicación.

En lo personal me gustó mucho esta forma de desarrollo, éstas tecnologías, a pesar de que me llevó mucho tiempo terminarla, al final todo funcionó como se esperaba y ahora puedo empezar a hacer más proyectos en NodeJs.

Referencias

NodeJs

<https://nodejs.org/es/>

Visual Studio Code

<https://code.visualstudio.com/>

MongoDb Atlas

<https://www.mongodb.com/cloud/atlas/signup>

*Tutorial para la configuración del cluster en MongoDB Atlas

<https://zellwk.com/blog/crud-express-mongodb/>

Microsoft Azure

<https://azure.microsoft.com/es-mx/>

**Tutorial para la subida de la aplicación a un servidor en producción

https://www.youtube.com/watch?v=8vPi6AeCLKQ&t=660s&ab_channel=FaztCode

Github

<https://github.com/>

***Tutoriales para el desarrollo de la aplicación

<https://www.youtube.com/playlist?list=PLo51Ae9kQrwqUEXK7oQbZv63KsdODzuAy>