**PROYECTO FINAL DAW**

Autoescuela Fast



Hugo Díaz Casado

20 de Febrero de 2024

1. Definición del proyecto
   1. Tipos de Empresa. Sectores Productivos
   2. Descripción del problema.
   3. Objetivos. Especificación de Requisitos

La finalidad de nuestra aplicación es dar un soporte a los usuarios que estén intentando sacarse el Permiso B de Conducir.

En base a esto, podemos definir los siguientes requisitos funcionales:

* ***Registro de usuarios*** 🡪 Los usuarios podrán registrarse en un apartado de “Regístrate” en el que se les pedirá una serie de datos, además en el registro los usuarios deberán de llenar todos los campos correctamente para poder acceder a la app. Si consigue registrarse se le generará un token de usuario durante su sesión.
* ***Inicio de Sesión y Autorización***🡪 Si el usuario tiene ya una cuenta para poder acceder a los test teóricos o su perfil, deberá “logearse” si lo hace correctamente se le mostrará un mensaje. Además cada vez que se “logee” se le generará un token de usuario durante su sesión.
  1. Herramientas de diseño

***Visual Studio Code*** 🡪 Utilizamos este editor de código ya que es una herramienta que nos facilita el desarrollo del código en cuanto a tiempo se refiere. Además nos trae innumerables extensiones gratuitas que descargaremos, como “intelisense”, detectores de errores, “debuggers”…

***NPM*** 🡪 Es un instalador de paquetes de JavaScript que nos permitirá instalar todas las dependencias necesarias para nuestro proyecto tanto en el lado del cliente como el del servidor.

***Node.js***🡪 Será nuestro lenguaje en el lado del servidor y en nuestra aplicación se encargará principalmente de hacer llamadas a nuestra API y conexiones con nuestra BBDD, para ello usaremos librerías de encriptación de datos, de generación de tokens, de conexiones a la BBBD…

Node nos facilita mucho hacer un MVC, con Node podremos realizar tanto los controladores como los modelos.

Emplearemos la versión de Node 18.16.1.

***Express***🡪Es uno de los frameworks más conocidos de Node.js que nos facilitará la creación de nuestra API, a la hora de generar nuestras rutas (endpoints), además de muchas otras cosas más. Emplearemos la versión 4.18.2.

***MongoDB***🡪 Node.js y Express para realizar nuestras API’s y conectarse a la BBDD empleará MongoDB, que un Sistema Gestor de Base de Datos no relacional, lo cuál nos permitirá una mayor flexibilidad.

Aun así, crearemos nuestra propias relaciones, la razón por la que hemos decidido usar MongoDB es la siguiente:

* Conecta muy bien tanto con Node.js como Express.
* Su rápidez ya que trabaja con datos BSON (JSON en binario) y esto permite evitarle a Node hacer conversiones a la hora de tratar con los datos.
* Para usar MongoDB usaremos la dependencia de Node.js llamada mongoose que nos permitirá hacer modelos y consultas.

MongoDB nos permite un servidor en la nube gratuito, llamado MongoDB Atlas, cada vez que iniciemos el servidor nos conectaremos y de forma asíncrona haremos las peticiones y llamadas a la BBDD cuando el usuario lo solicite desde el cliente.

***React.js***🡪 Esta librería nos permitirá generar interfaces muy escalables y reactivas en lado del cliente, generamos nuestros propios componentes (que podrán repetirse en varias rutas de la aplicación) y plantillas que se irán renderizando cada vez que el cliente interactúe con la página.

React nos permitirá hacer generar las vistas del MVC, para conectarlas a los controladores y por consiguiente a los modelos, usaremos la librería de js Axios. Además usaremos algunas de las librerías más conocidas en el ecosistema de React como: React Router (Para generar rutas en la App), React Hook Form (Manejo de Formularios), Zod (Validación de Esquemas)…

***Material UI*** 🡪 Esta librería de React nos proporcionará componentes ya creados, con lo cuál nos facilitará tanto la elegancia como la rapidez del desarrollo.

***ReCharts*** 🡪 Esta librería de React nos proporcionará diagramas que usaremos en el perfil del usuario, para que pueda ver su seguimiento y progresión en la aplicación.

***Vite*** 🡪 Esta herramienta de compilación nos facilitará el desarrollo en el entorno del cliente, además de la empaquetación de todos nuestros archivos a la hora de subir a producción nuestra aplicación.

***Tailwind CSS***🡪 Este framework de CSS nos permitirá estilar nuestras interfaces generadas con React de una manera flexible y ágil, usará por debajo también Post CSS para guardar los cambios cada vez que nombremos una clase.

***Git*** 🡪 Es un Sistema de Control de Versiones distribuidor que nos ayudará a la hora de manejar errores y al desarrollo de la aplicación ya que podremos ir guardando cada uno de nuestros cambios de la app. Además todos estos cambios los podremos ver reflejados en el entorno de GitHub, en nuestro correspondiente repositorio y en nuestros commits (con diferentes títulos descriptivos con cada uno de los cambios para ir viendo la evolución de nuestro desarrollo).

***Vercel y Render***🡪 Serán dos lugares dónde podremos subir a producción nuestra aplicación hemos escogido Vercel para “deployar” el entorno cliente y Render para “deployar” el servidor, la ventaja que nos dan estos dos hosting, son las siguientes:

* Nos proporcionan un servicio gratuito
* Cada vez que realizamos un commit, tanto nuestro servidor como cliente se “autodeployará” con los cambios realizados.