

# RESUMEN INICIAL PROYECTO

## FINALIDAD DEL PROYECTO

El proyecto que quiero presentar consiste en una autoescuela virtual llamada "AutoescuelaFast" que tiene como fin ayudar a los usuarios con numerosos test a sacar su permiso de conducir teórico. Además de poder realizar estas pruebas, los usuarios podrán comprobar sus estadísticas como por ejemplo sus últimos 10 resultados, sus resultados por test, porcentaje de test aptos, etc. Además en caso de obtener buenos resultados tendrán acceso a test extra.

Los clientes también dispondrán de un lugar de contacto para que la propia empresa pueda responder dudas por correo. Asimismo, la aplicación tendrá la opción de poder comprobar las preguntas y respuesta más demandadas por los usuarios, ya respondidas por nuestros profesionales.

Además de estas funciones, buscamos nuevos empleados que ayuden y generen un mayor soporte a los usuarios que usen nuestra web, con lo que dispondremos de un apartado donde los usuarios puedan adjuntar sus currículums y que en un futuro puedan ser seleccionados para poder adherirse a nuestra plantilla.

## TECNOLOGÍAS

Las tecnologías que hemos utilizado para realizar este proyecto son las siguientes:

- **VSCODE:** Utilizaremos este IDE para acelerar el desarrollo de nuestra aplicaciones, gracias a los pluggings que podemos instalarnos.

- **Excalidraw:** Usaré esta aplicación para poder generar los mockups y wireframes de la aplicación, ya que dispone de herramientas para acelerar bastante este proceso.

- **Dia:** He decidido utilizar Dia para generar los diagramas tanto de UML, como E-R, como casos de uso.

- **MongoDB:** He querido utilizar una base de datos no relacional (aún así crearé mis propias relaciones en determinadas situaciones), ya que al trabajar con tantos usuarios quiero que se envíe al cliente con rapidez y el formato BSON que usa MongoDB para su tratamiento de datos ayuda bastante a ello.

- **NodeJS**: Como lenguaje de servidor he querido utilizar NodeJS, ya que casa muy bien con MongoDB además que generar una API y MVC resulta bastante sencillo y eficaz, usaremos este lenguaje para validar datos, modelos, almacenamiento de archivos en la nube, gestión de la BBDD, etc.

- **ExpressJS**: He pensado que sería muy eficaz usar un framework de NodeJS para acelerar el desarrollo de la aplicación, este framework nos facilita mucho el enrutamiento de nuestra API.

- **ReactJS**: Para el lado del cliente he decidido utilizar React ya que es una librería de JS muy versátil y muy flexible ya que en base a las necesidades del proyecto podemos ir instalando diferentes librerías que tiene creadas el lenguaje (para rutas relativas, para redirecciones, para ofrecer plantillas de componentes...), además dispone de una gran ventaja y es que podemos realizar páginas SPA (Simple Pages Application), las cuales nos permiten que la página se maneje mediante eventos y no necesite un recargado constante al por ejemplo modificar los datos de una BBDD.

- **Tailwind CSS**: Para facilitar y acelerar el diseño he querido usar Tailwind porque creo que hace muy buena pareja con React para crear interfaces muy elegantes y algo más personalizables que otros frameworks de css, aunque el usar Tailwind no quita que algunas partes del código estén diseñadas con CSS Nativo como por ejemplo animaciones o colores gradientes.

- **Git**: Usaremos un control de versiones para poder tener el proyecto en la nube y sea de código abierto, además Git nos facilita la subida a producción de la aplicación y el tratamiento de errores, visualizaremos nuestro repositorio en GitHub.

- **Vercel / Render**: He decidido que sería buena idea alojar la API (página dinámica) de nuestra aplicación en un hosting diferente al de nuestra interfaz (página estática), para ello usaré estos dos hosting gratuitos y bastante fiables que son muy recomendables para un stack MERN.