

# ThunderJobs

Jonathan Tanasa

**Este proyecto es un portal de empleo diseñado para conectar a empresas con candidatos calificados y facilitar la búsqueda y aplicación de empleos.**

## Índice

- Funcionalidades principales
- Tecnologías utilizadas
- Instalación y despliegue
- Rutas del servidor
- Estado del proyecto
- Contacto

## Funcionalidades principales

1. Registro de Usuarios y Empresas: Los usuarios y las empresas pueden registrarse en la plataforma para acceder a sus respectivas funciones y servicios.
2. Búsqueda Avanzada de Vacantes: Los usuarios pueden buscar vacantes utilizando diversos filtros, incluyendo ubicación, palabra clave, empresa y categoría, lo que les permite encontrar oportunidades laborales que se ajusten a sus preferencias y habilidades.
3. Solicitud de Vacantes: Los usuarios pueden solicitar vacantes de empleo, adjuntando su curriculum vitae, que se carga y almacena en el servidor. Esto facilita a los usuarios la aplicación a las oportunidades laborales de manera conveniente y eficiente.
4. Seguimiento de Solicitudes: Los usuarios tienen la capacidad de realizar un seguimiento de las solicitudes enviadas, lo que les permite mantenerse informados sobre el estado de sus aplicaciones y cancelarlas si así lo desean.
5. Gestión de Vacantes por Parte de las Empresas: Las empresas pueden crear y modificar vacantes de empleo, lo que les permite mantener actualizada la información sobre las oportunidades laborales que ofrecen.
6. Visualización y Gestión de Solicitudes por Parte de las Empresas: Las empresas pueden ver las solicitudes recibidas para sus vacantes y descargar los currículums adjuntos a dichas solicitudes. Además, tienen la capacidad de aceptar o rechazar las solicitudes según sus criterios de selección.

## Tecnologías utilizadas

### Frontend:

- HTML: Lenguaje de marcado utilizado para estructurar el contenido de las páginas web.
- CSS: Lenguaje de estilos utilizado para diseñar y dar formato al contenido HTML.
- JavaScript: Lenguaje de programación utilizado para agregar interactividad y funcionalidades dinámicas a las páginas web.

### Backend (Server):

- Node.js: Entorno de ejecución de JavaScript utilizado para ejecutar el servidor.
- Express: Framework de Node.js utilizado para crear aplicaciones web y APIs de manera sencilla y eficiente.
- Axios: Cliente HTTP basado en promesas para el navegador y Node.js.
- Bcrypt: Librería utilizada para el cifrado seguro de contraseñas.
- Cors: Middleware de Express para habilitar el acceso a recursos de diferentes orígenes.
- Diacritics: Utilizado para eliminar marcas diacríticas de los caracteres.
- JSON Web Token (jsonwebtoken): Implementación de JSON Web Tokens para la autenticación de usuarios.
- Mongoose: Biblioteca de modelado de objetos MongoDB para Node.js.
- Mongoose Paginate V2: Plugin de paginación para Mongoose.
- Multer: Middleware de Node.js para manejar datos de formularios multi-partes, como la carga de archivos.
- Path: Módulo de Node.js para manejar y transformar rutas de archivo y directorio.
- UUID: Librería para la generación de identificadores únicos universales (UUID).

### Base de datos:

- MongoDB Atlas

Para resetear la base de datos ejecuta el script createDB.js

```
$ node createDB.js
```

El script recoge la información de resources/dataDB.js

## DevDependencies:

- Vercel: Herramienta utilizada para implementar y administrar proyectos web en la nube.

## Instalación y despliegue

1. Descarga e instala Node.js en tu pc
2. Importa y configura el proyecto en tu pc

```
# Clona el repositorio en tu pc (si no lo tienes aún)
$ git clone https://github.com/Jonitp7/thunderjobs.git
# Instala todas las dependencias del package.json
$ npm i
# Arranca el servidor en local
$ node index.js
```

3. Despliegue del proyecto en Vercel con tu cuenta GitHub (opcional)

Primero debes crear un repositorio en tu cuenta de GitHub

```
# Debes configurar todas las peticiones fetch de los scripts para que hagan las peticiones
# a la URL de tu dominio.
# Sube el código actualizado a tu repositorio remoto
$ git init
$ git add .
$ git commit -m "(mensaje del commit)"
$ git remote add origin <url-tu-repo>
$ git push origin main
# Loggeate en Vercel usando tu cuenta de Github
$ vercel login
# Realiza el despliegue y sigue las instrucciones
$ vercel deploy
```

## Rutas del servidor

Importante: Muchas rutas usan la función `verificarToken`, por lo que para probarlas será necesario incluir el JWT en la cabeza de la petición

Empresas:

- GET (URL)/api/empresas
- POST (URL)/api/empresas/login
- GET (URL)/api/empresas/perfil
- POST (URL)/api/empresas

Usuarios:

- POST (URL)/api/usuarios/login
- GET (URL)/api/usuarios/perfil

- POST (URL)/api/usuarios

Vacantes:

- GET (URL)/api/vacantes
- GET (URL)/api/vacantes/:id
- POST (URL)/api/vacantes
- GET (URL)/api/vacantes/empresa/:cif
- PATCH (URL)/api/vacantes/:vacante\_id
- PATCH (URL)/api/vacantes/modificar/:vacante\_id

Solicitudes:

- GET (URL)/api/solicitudes/usuario/:usuario\_id
- GET (URL)/api/solicitudes/empresa/:cif
- POST (URL)/api/solicitudes
- GET (URL)/api/solicitudes/descargar/:solicitud\_id
- PATCH (URL)/api/solicitudes/estado/:solicitud\_id

## Estado del Proyecto

Este proyecto se considera completado y no se espera realizar cambios significativos en él en el futuro cercano.

## Contacto

Para cualquier pregunta, comentario o problema relacionado con el proyecto, no dudes en ponerte en contacto conmigo:

- Correo Electrónico: [jonathan.tanasa7@gmail.com]
- GitHub: [jonitp7]