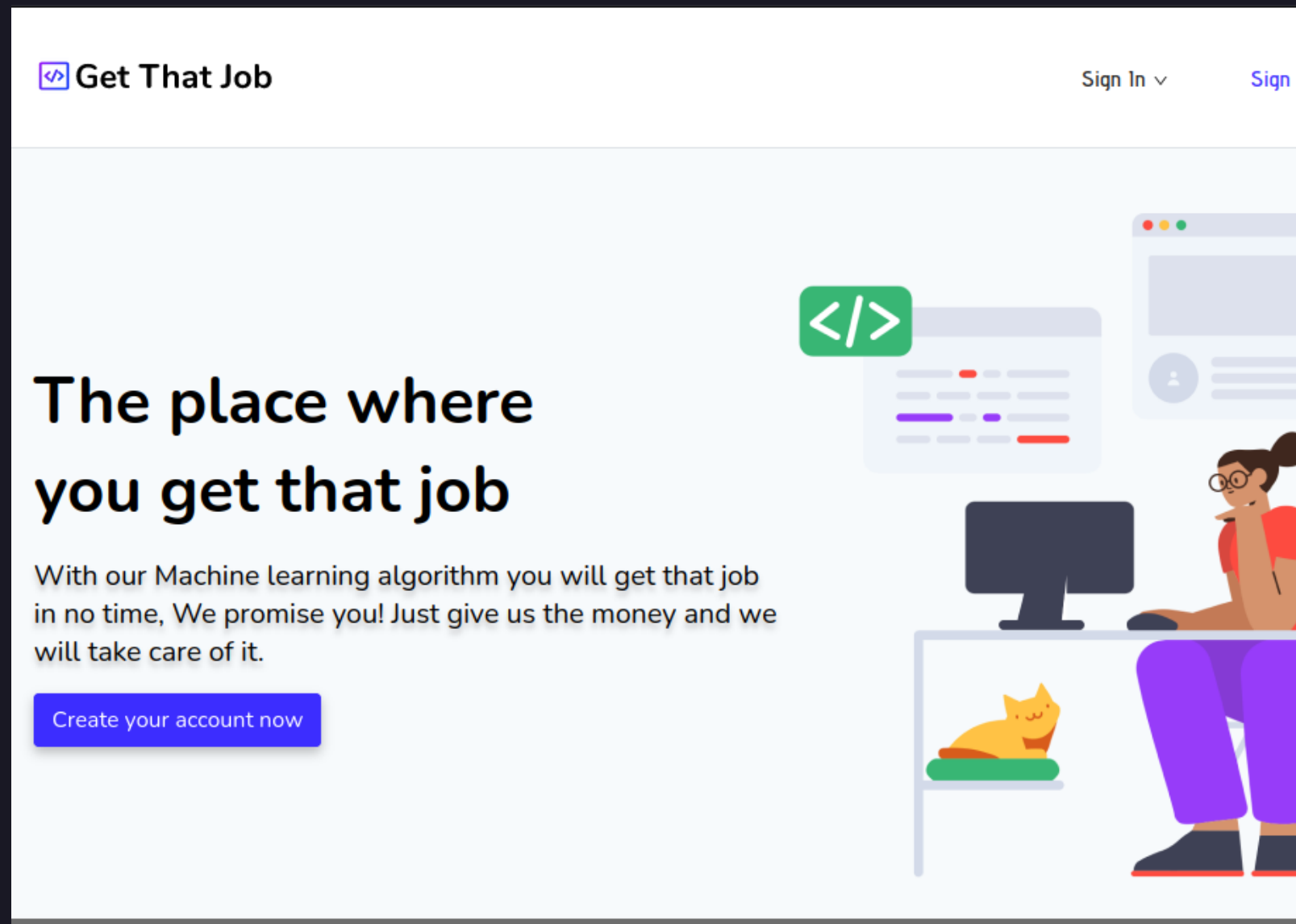


A person's hands are shown typing on a silver laptop keyboard. The laptop screen displays a webpage with text and a search bar. In the background, a potted plant is visible on a desk. The entire scene is dimly lit, with a dark blue overlay on the right side of the image.

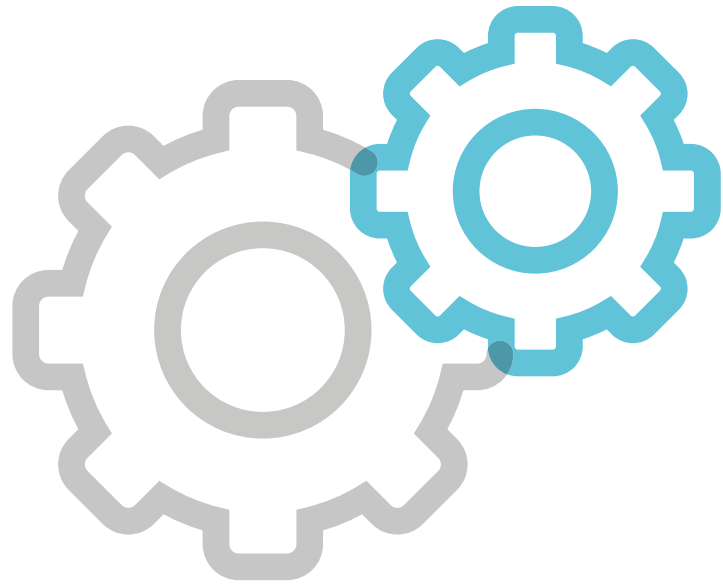
Get That Job

DESCRIPCIÓN

01



Get That Job es un sistema web que permite publicar y difundir vacantes de empleo. Mediante reclutadores, empresas que están interesadas en integrar nuevos miembros a su organización. De la misma forma permite a los profesionales del mundo tec ser parte de dichas corporaciones.



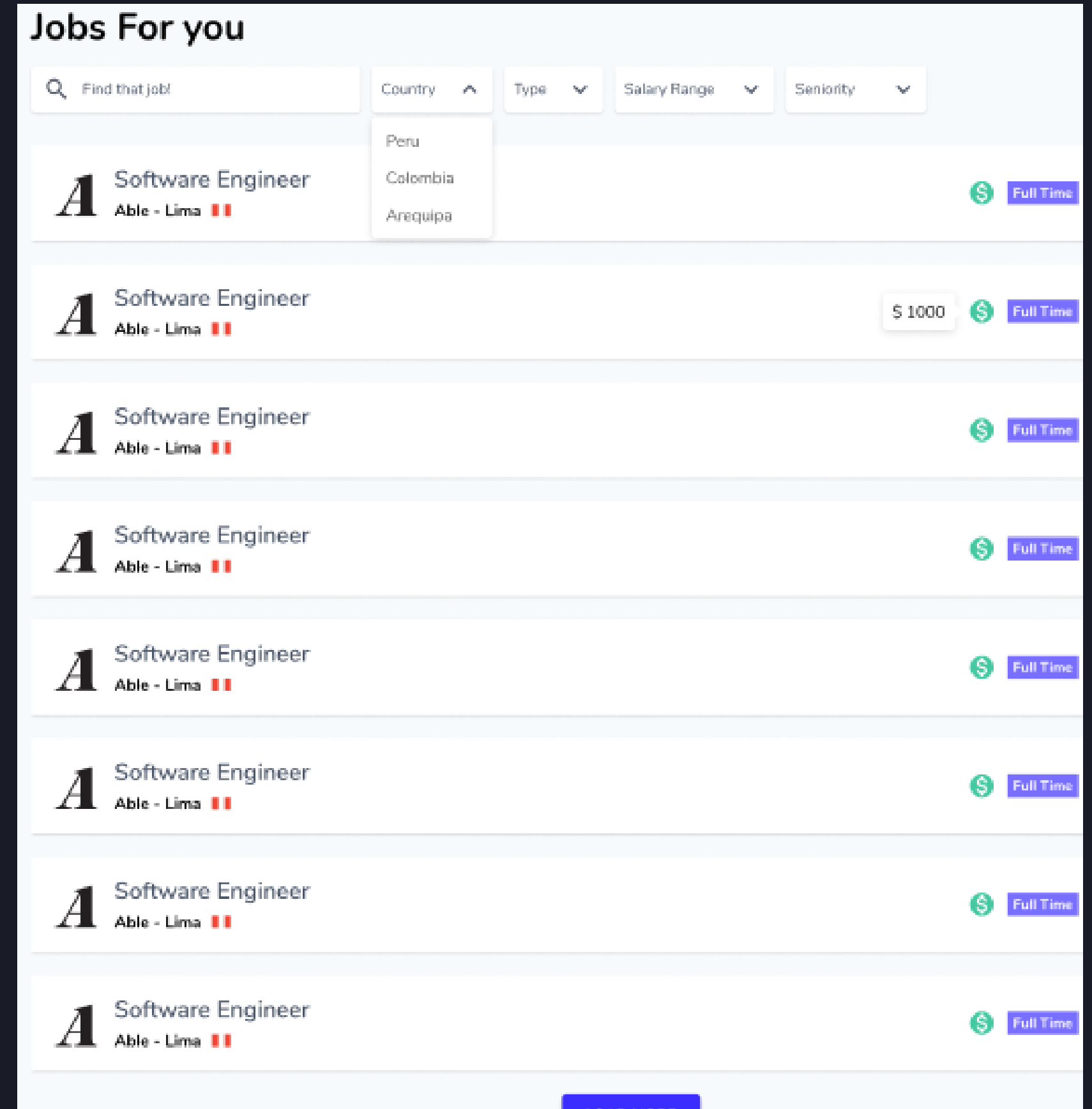
PROBLEMA

El objetivo de la empresa Get That Job es funcionar como intermediaria entre empresas interesadas en ampliar los recursos humanos en su organización y profesionales que desean postularse a dichas vacantes, es un problema poder llevar la administración y el control adecuado de este proceso de reclutamiento.

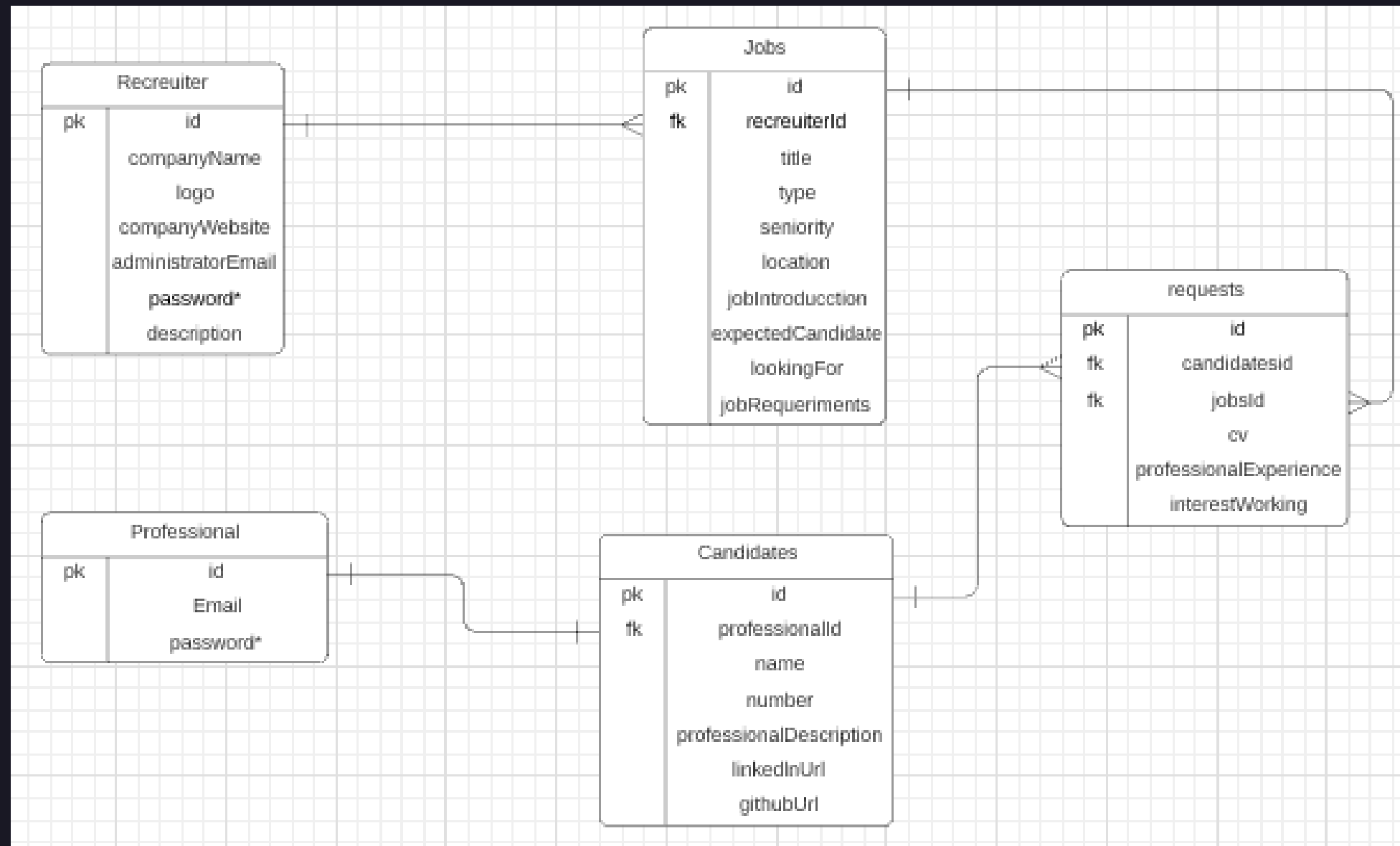
Solución

03

- Se diseñó e implementó un sistema capaz de administrar cada una de las publicaciones de empleo
- los profesionales pueden crear una cuenta, pueden buscar las vacantes a nuevos empleos
- Se lleva el historial y control de cada una de las vacantes, mediante la implementación adecuada de una base de datos



Evidencias



● modelo de datos

● Probando rutas

The screenshot shows a REST client interface with a sidebar on the left containing a file explorer with folders 'recruiters' and 'auth'. The main panel displays a POST request to 'localhost:3000/api/auth/login' with a JSON body:

```
{  "email": "test1_front@gmail.com",  "password": "test1"}
```

. The response is a 200 OK status with a response time of 2.78 s and a body size of 193 B. The response body is shown in the 'Preview' tab as a JSON object:

```
{  "message": "Authenticated sucessfully",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTYzMTMyMzYyMSwiZXhwIjoxNjMxMzU5NjIxfQ.jFh8f8fMoJEk02XjZqw7Zqw-X07hN9ccCMHcfW32xyU"}
```

Dashboard / Get That Job

to Environment Cookies

POST localhost:3000/api/auth/login Send 200 OK 2.78 s 193 B A Minute Ago

Filter

recruiters

GET recruiters

PUT recruiters

auth

POST recruiters/signup

POST recruiters/login

POST professionals/login

JSON Auth Query Header 2 Docs

```
1 {
2   "email": "test1_front@gmail.com",
3   "password": "test1"
4 }
```

Preview Header 8 Cookie Timeline

```
1 {
2   "message": "Authenticated sucessfully",
3   "token":
4     "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiJEsImhhdCI6MTYzMTMyMzYyMSwiZXhwIjoxNjMxMzU5NjIxfQ.jFh8f8fMoJEk02XjZqw7Zqw-X07hN9ccCMHcfW32xyU"
5 }
```

06

● Probando rutas

The screenshot displays a REST client interface with the following components:

- Request Bar:** Method `GET`, URL `localhost:3000/api/users/`, and a `Send` button.
- Response Summary:** Status `200 OK`, Time `669 ms`, Size `225 B`, and a timestamp `Just Now`.
- Response Body:** A JSON array containing one user object.

```
1 {  
2   "data": [  
3     {  
4       "id": 1,  
5       "type": "professional",  
6       "email": "test1_front@gmail.com",  
7       "password":  
8         "$2b$10$0NfUvJ5BijVG.PdzD9xSIecGo7rd1CktqdE2XPQwy0xQkfg.RpCH.",  
9       "createdAt": "2021-09-01T01:00:34.000Z",  
10      "updatedAt": "2021-09-01T01:00:34.000Z"  
11    }  
12  ]  
}
```
- Header Section:** A table with one header row.

Authorization	AJYHeyuKNnrLkZjvdMGc_NkOQ	✓	🗑️
---------------	---------------------------	---	----
- Form Section:** A table for adding new headers.

New header	New value
------------	-----------


● FrontEnd

Login

As Professional

Email

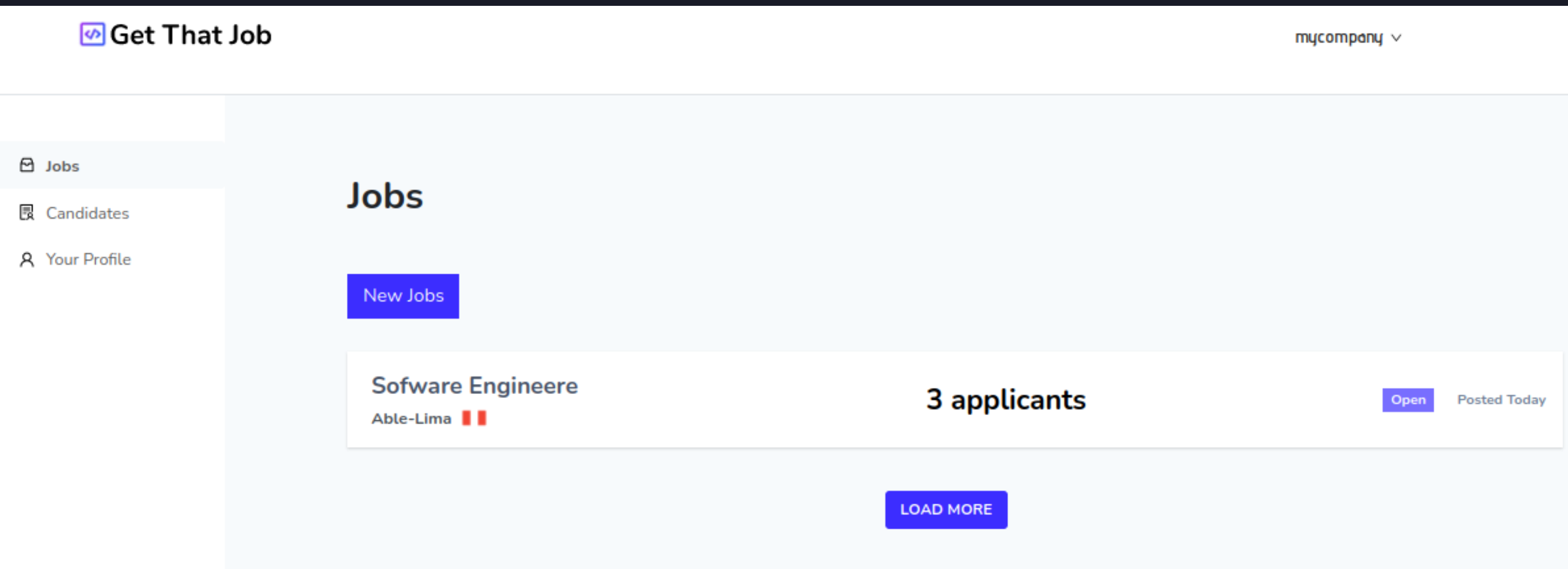
Password



Login

08

• FrontEnd



Oportunidad de mejora

- Depurar el código
- Buscar inconsistencias con los endpoints y el frontEnd

Conclusiones

El desarrollo del proyecto como full stack ha sido exitoso, en esta primera versión se logró concluir de forma exitosa la parte del Back-End y se pretende concluir la integración con el Front-End en futuras versiones.