



DIGITALHOUSE

Desarrollo Web Full Stack Node

Sprint - Práctica Profesional

Trabajo Integrador - Sprint Práctica Profesional

Introducción

¡Prepárate para entregar un proyecto de alto valor y dar un impulso a tu perfil profesional! Al agregarle <millas> de experiencia, estarás un paso más cerca de alcanzar tus sueños laborales.

El objetivo de este desafío de codificación es brindarte la oportunidad de mostrar tu conocimiento y mostrarnos cómo trabajas en un proyecto de la vida real.

Este challenge te pide desarrollar un frontend que consumirá datos de un API que también deberás desarrollar.

El frontend deberá mostrar la información obtenida respetando las pautas que se describirán más abajo. Asimismo, la API también debe ser desarrollada siguiendo requisitos explícitos.

¡Aprovecha esta oportunidad para acercarte cada vez más a tu trabajo ideal!



Contexto

Somos una empresa multinacional de Recruiting de Recursos Humanos, para insertar a los aspirantes en puestos de trabajos altamente demandados en la industria.

Queremos ofrecer a nuestro público una forma de ver nuestro catálogo de aspirantes.

Para ello, queremos desarrollar una aplicación web que permita a los usuarios ver los aspirantes disponibles. Ten en cuenta que:

- No sabemos si el framework que utilicemos ahora será el definitivo, pero queremos reutilizar el máximo de código posible.
- La aplicación debe ser fácil de usar y agradable a la vista.
- Tenemos un 80% de usuarios que vienen de navegadores de escritorio, sin embargo, la aplicación se debe visualizar perfectamente también en dispositivos móviles y en tablets.

Objetivo

El challenge, está dividido en 3 partes:

1. Desarrollar la Base de datos “RecruitingRH”, valiéndose del motor de base de datos **MySQL**.
2. Construir una **API-REST**, utilizando para ellos **Node.JS + Express**, la misma debe mostrar los **aspirantes** registrados en la base de datos.
3. Diseño y desarrollo de un Frontend cliente usando la librería **React + Bootstrap**, si desea también puede utilizar **Tailwind**.

Requisitos:

- **Documento provisto:** antes de iniciar revisemos en detalle el [material provisto](#).
- **Cuenta de GitHub:** Todas las entregas se harán, sin excepción, en repositorios públicos de GitHub. Todo el grupo deberá contribuir, sin excepción, al mismo repositorio.

Consignas

FrontEnd

1. Debe ser desarrollado en **React JS**.
2. Los componentes deben ser desarrollados en ECMAScript 6 como base del JS.
3. Creación de Componentes Basados en **Componentes Funcionales** y React Hooks (useEffect, useState,)
4. Para el consumo de servicios **Api Rest** debe realizarse por medio de **Fetch o Axios** haciendo mucho uso de programación asíncrona con Promesas y si desea puede utilizar async - await.
5. La navegación entre los componentes debe realizarse por medio de React **Router Dom**.
6. El tipo de proyecto frontend debe ser un **SPA**.
7. El proyecto de frontend debe ser creado con la plantilla **vite**.
8. Todas las dependencias del frontend deben ser instaladas y manejadas con **NPM package Manager**.
9. Puede utilizar el framework **Bootstrap o Tailwind** para la maquetación de los componentes.
10. Determine usted la cantidad de componentes que considere necesarios crear.

Entregable: Dashboard completamente funcional.

BackEnd - Base de Datos

11. Debe crear una base de datos **"recruitingRH"**, para ello debe utilizar el motor de base de datos **MySQL**.
12. Crear una tabla denominada: **"aspirantes"**
13. La tabla de **"aspirantes"**, debe poseer los siguientes campos, genere su propia impronta y determine los tipos de datos, longitud de los mismos, las respectivas restricciones y sobre todo considere si debe existir otra tabla, en base a los campos dados:
 - Dni
 - Nombre
 - Apellido
 - Email
 - Teléfono
 - Perfil URL de LinkedIn
 - Fecha nacimiento
 - Sexo
 - Imagen
 - Profesión

Entregable: Diagrama Entidad Relación o Modelo Entidad Relación de la Base de Datos, así como el respaldo de la estructura de la Base de datos (.sql).

BackEnd - Api - Rest

Nuestra API de aspirantes va a proveernos de dos endpoints muy importantes. El primero nos entregará la lista completa de aspirantes y el segundo nos permitirá consultar el listado de las profesiones que pueden poseer los aspirantes.

Vamos con esas consignas:

14. API de Aspirantes y de Profesiones

Nuestra API de aspirantes y de profesiones, serán muy similares. Sus endpoints entregarán la lista completa de aspirantes y profesiones, dicha API debe ser creada valiéndose de Node.JS y el Framework Express.

Para la construcción de la API, se debe utilizar el Patrón de Diseño: Modelo Vista Controlador.

Entregable: URL funcionales devolviendo datos de aspirantes y profesiones en formato JSON.

