

Alcance

Nuestro Developer Hiring Test, apunta a brindar un escenario posible de trabajo, para que puedas desarrollar tus habilidades como Developer, desenvolverte, y poner en práctica tu experiencia.

¡Empecemos!

Imagínate que dentro de tu día a día de trabajo habitual, el equipo de producto necesita desarrollar una nueva funcionalidad y por esto, define las bases bajo el flujo de trabajo definido, en donde construye una Historia de Usuario, (HdU) con la estructura acordada (objetivo a nivel usuario, assumptions, y criterios de aceptación), y lleva dicho requerimiento a refinar junto al equipo de desarrollo.

Este requerimiento es aprobado y se lleva a cabo dentro de un sprint. Al finalizar este desarrollo, se debe notificar que la funcionalidad está lista para las pruebas funcionales y técnicas, y tu tarea consiste en llevar adelante este desarrollo.

Consignas

Te encontrarás con el análisis funcional del requerimiento junto a la HdU que se te entrega como desarrollador y a partir de ella debes:

- Construir la funcionalidad solicitada utilizando Javascript como lenguaje de programación (Preferiblemente Nest Js como framework) y Prisma como ORM.
- Documentar y/o comentar el proceso desarrollado.
- Crear las migraciones de base de datos según corresponda.
- Asegurar el correcto funcionamiento del desarrollo entregado.
- Será un plus el uso de Vue Js para el frontend de la aplicación.

El formato de entrega debe ser un repositorio en Github, Gitlab o Bitbucket.

Nota: Desplegar este proyecto en un servidor, entregará un puntaje adicional.

Funcionalidad: Importación de datos

Historia de usuario

Yo como usuario de Cargofive, requiero una API que me permita usando autenticación JWT, scrapear los artículos de un blog por categorías y luego poder visualizar los resultados filtrados por autor o categoría de forma ordenada en una estructura JSON definida.

Assumptions

1. Desarrollar una API REST con un sistema de usuarios, de modo que:
 - Se requiere de un endpoint para crear un usuario ("Name", "Email", "Password")
 - Se requiere de un endpoint para actualizar el nombre, el correo electrónico y la contraseña
 - Se requiere de un endpoint para ver el listado de usuarios
 - Se debe poder iniciar sesión con el estándar JWT
2. Se debe poder scrapear desde <https://cargofive.com/blog/>:
 - 3 primeros artículos de categoría **Innovation**
 - 3 primeros artículos de categoría **International Logistics**
 - Guardar cada artículo en DB con los siguientes campos ("Title", "Published At", "Author", "Source Link", "Category", "Body Description");
3. Una vez finalizado el proceso de scrapeo, crear un endpoint /api/articles para consultar los artículos desde la base de datos con opción de filtrado:
 - **Por Categoría**
 - **Por Autor**
4. Para el retorno de los artículos se requiere que el formato JSON de la respuesta sea:

```
[{
  title: "",
  description: "",
  author: "",
  published_at: "",
  category: [],
  source: ""
}]
```

Criterios de aceptación:

- ✓ Al momento de enviar la petición se debe retornar un mensaje de éxito o error según el resultado del proceso usando la arquitectura REST.
- ✓ Los campos del formulario deben estar correctamente validados.
- ✓ Los usuarios deben relacionarse con los posts como autor, un usuario puede tener muchos posts.