



<?php echo "Start the challenge"; ?> [Backend]

Desafío Backend

Para continuar en el proceso de selección de personal para la posición de Backend de MailUp se deberán realizar dos desafíos para así validar los conocimientos manifestados en la entrevista.

Los desafíos mencionados a continuación no tienen ningún fin comercial para la empresa a la que está aplicando, solo es para fines evaluativos y así poder continuar en el proceso de selección de personal.

Ejercicios a resolver

1 - Construir un API REST

2 - Utilizar un API REST

Contexto

Estos desafíos están pensados para simular un día normal de trabajo en donde el programador debe poder desarrollar cualquiera de las dos tareas dependiendo cual se le asigne, **son escenarios simplificados de una tarea completa.**

Notas importantes

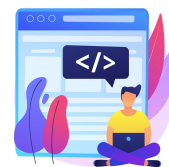
- * El lenguaje de programación a utilizar deberá ser PHP + framework Laravel.
- * Se tendrá en cuenta los tiempos de desarrollo una vez entregado el desafío.
- * Se tendrá en cuenta todo el valor agregado que ofrecen.

¿Consultas? ¿Dudas?

Le pedimos por favor que si tienen alguna consulta o duda sobre el desafío le escriba a su evaluador y lo ayudaremos a despejar su inquietud para concluir el desafío y continuar en el proceso de selección.

¿Que hago una vez que termino los desafíos?

Una vez que se termine el desafío debe responder el correo que el evaluador le envió con este desafío con los links al repositorio público de cada ejercicio resuelto.



Ejercicio 1: Construir un API REST

Al momento de desarrollar nuevos features o refactorizar nuestro sistema nos encontramos ante la necesidad de saber ¿Cómo? y ¿Dónde? agregar código y como así también de ser ágil a la hora de buscar código para analizar y refactorizar.

Dicho lo anterior el programador deberá crear un API REST, el concepto que se aplicará para esta evaluación es la de lograr administrar información de productos de un E-commerce por medio de esta API.

Este sistema deberá tener GIT como control de versión y se debe subir a GITHUB/GITLAB al finalizar para compartirlo con su evaluador de MailUp.

Nuestros sistemas tienen un gran enfoque en los E-commerces y por eso se seleccionó dicho escenario.

El API REST debe persistir los datos en una base de datos MySQL.

El API REST debe tener los siguientes endpoints (CRUD Completo)

- * Crear un producto
- * Obtener todos los productos (Implementar una lógica de paginación y de búsqueda por nombre)
- * Obtener información de un producto
- * Modificar información de un producto
- * Eliminar un producto

Datos básicos del producto

name, description, image, brand, price, price_sale, category, stock

Ejemplo de datos

```
{
  name: "MacBook Pro 13.3" Retina [MYD82] M1 Chip 256 GB - Space Gray"
  description: ""
  image: "apple.com/v/macbook-pro/ac/images/overview/hero_13__d1tfa5zby7e6_large_2x.jpg"
  brand: "Apple"
  price: 2000
  price_sale: 1950
  category: "Macbook Pro"
  stock: 5
}
```

Conocimientos a validar

PHP

LARAVEL (Eloquent con MYSQL)

CONCEPTO Y CREACIÓN DE API REST

GIT

GITHUB o GITLAB

Recomendaciones a tener en cuenta

Validaciones, código de status de respuestas, nomenclatura de los endpoints, verbos de los endpoints (GET, POST, PUT, etc), documentación de uso del API, migrations y seeders de Laravel.

Ejercicio 2: Utilizar un API REST



La creación de nuevas plataformas SaaS es constante y los clientes las adquieren para mejorar y automatizar sus procesos, lo que hace que los clientes necesitan sincronizar sus datos entre estas plataformas y la nuestra, Luego de identificar estratégicamente que plataformas nos beneficia más integrarnos llega el día de comenzar el desarrollo y así debemos integrarnos a plataformas de terceros como Facebook Ads, Whatsapp, Vtex, Magento y existen miles de opciones más.

Hoy en día contamos con más de 30 integraciones que están intercambiando información con nuestro sistema y para eso es importante saber cómo utilizar una API no construida por nosotros.

Dicho lo anterior el programador deberá lograr conectarse a un API externo y descargar todos los items de Photos del API <https://jsonplaceholder.typicode.com/photos> y guardarlos en una base de datos.

Este sistema deberá tener GIT como control de versión y se debe subir a GITHUB/GITLAB al finalizar para compartirlo con su evaluador de MailUp.

La APP debe persistir los datos en una base de datos MySQL.

La APP no debe descargar la imagen, solo guardar el string del API.

Conocimientos a validar

PHP

LARAVEL (Eloquent con MYSQL)

CONCEPTO Y UTILIZACIÓN DE API REST

GIT

GITHUB o GITLAB

Recomendaciones a tener en cuenta

Validaciones de conexión (reintentos) y control de que el API externo esté online, validación de status de respuestas del API, control de datos al recorrer los resultados, métodos helpers de Laravel.

Creditos

imagen: Freepik.com