

# Prueba técnica – Web developer Jr.

## Objetivos:

Evaluar conocimientos básicos de PHP, Laravel, APIs REST y bases de datos.

## Preguntas teóricas (20 min)

- a) PHP
  - a. ¿Qué diferencia hay entre los operadores == y === en PHP?
  - b. Explicar que son los namespaces y porque son útiles.
- b) Laravel
  - a. Menciona al menos 3 métodos de rutas en Laravel y su propósito.
  - b. ¿Para qué sirven los migrations y seeder en Laravel?

## Pregunta practica (30 min)

Escriba una función en PHP llamada **findSecondLargest** que reciba un arreglo de números como argumento y que retorne el segundo número más grande.

```
$input = [3, 5, 1, 7, 2];  
function findSecondLargest($array) { /* ... */ }  
// Resultado esperado: 5
```

## Criterios de evaluación.

- Correcto funcionamiento
- Legibilidad del código
- Manejo de Edge cases (ej. Arreglo vacio)

## Ejercicio práctico (120 – 180 min)

**Contexto:** Necesitamos un módulo para gestionar “**tareas pendientes**” (To - Do List) con Laravel.

### Requisitos:

- a. Base de datos
  - a. Crear una migración para la tabla tasks con los campos.
    - i. id
    - ii. title (string)
    - iii. description (text)
    - iv. due\_date (datetime)
    - v. is\_completed (boolean)
  - b. Generar un Seeder con 5 tareas de ejemplo.
- b. Backend Laravel
  - a. Crear un modelo Task con su controlador
  - b. Implementar 3 rutas API RESTful básicas:
    - i. GET /tasks - Listar todas las tareas
    - ii. POST /tasks - Crear una nueva tarea
    - iii. PUT /tasks/{id} - Marcar una tarea como completada
- c. Frontend (Opcional bonus)
  - a. Crear frontend con Blade o livewire para mostrar las tareas en una lista simple y un formulario de creación

### Entregables:

- Código cargado a un repositorio github.
- Instrucciones claras para ejecutar el proyecto (ej. Comandos artisan, dependencias)