

## Prueba Técnica – Desarrollo

Hola,

Gracias por tu interés en formar parte del equipo.

A continuación, te compartimos los detalles de la **prueba técnica para el rol de Desarrollador**. Esta prueba tiene como objetivo evaluar tus habilidades tanto en backend como en frontend o fullstack, según sea el caso, enfocada en el consumo de servicios externos, estructuración de datos, buenas prácticas de desarrollo, diseño de APIs REST y creación de interfaces usables y amigables.

### Objetivo general

Deberás desarrollar una aplicación que:

- Consuma información desde la [PokeAPI](#) (no requiere autenticación).
- Exponga esa información a través de una API REST desarrollada en Java. *(Solo para Backend y Fullstack)*
- Consuma esa API desde un frontend en Angular. *(Solo para Frontend y Fullstack)*
- Presente de forma amigable los datos consultados del Pokémon.

### Entrega

- Sube tu solución a un repositorio Git (GitHub, GitLab, Bitbucket, etc.).
- Incluye un archivo `README.md` con:
  - Instrucciones para ejecutar el backend y frontend localmente.
  - Dependencias necesarias (JDK, Node, etc.)
  - Tiempo estimado que te tomó completar la prueba.
  - Puedes desplegar la app en algún servicio gratuito como GitHub Pages, Netlify o Vercel y agregar el enlace en el `README.md`. *(Opcional Solo para Frontend y Fullstack)*

## Resultados esperados

### Backend (Java / Spring Boot) *(Solo para Backend y Fullstack)*

- Crear una API REST que reciba como parámetro el nombre de un Pokémon (/pokemon/{nombre}).
- Obtener los datos del Pokémon desde la API pública.
- Procesar la información para devolverla estructurada, clara y sin campos innecesarios.
- Datos esperados en la respuesta:
  - Nombre / especie
  - Tipo(s)
  - Habilidades
  - Ataques
  - Estadísticas
  - Imagen en base 64

### Frontend (Angular) *(Solo para Frontend y Fullstack)*

- Crear una interfaz que permita al usuario:
  - Ingresar el nombre de un Pokémon.
  - Visualizar los datos retornados por el backend.
  - Mostrar mensajes de error si el Pokémon no existe o hay un fallo de red.
- La interfaz debe ser:
  - Responsiva
  - Intuitiva
  - Visualmente ordenada (puedes usar Bootstrap, Angular Material, Tailwind o similar)

## Requisitos adicionales (deseables, pero no obligatorios)

### Backend

- Registrar cada búsqueda en una base de datos local (H2 o Postgres).
- Documentar el API con Swagger/OpenAPI.
- Manejo de excepciones con respuestas HTTP adecuadas.
- Contener pruebas unitarias o de integración (JUnit, Mockito, etc.).
- Despliegue en Docker.

### Frontend

- Aplicar principios básicos de seguridad en el cliente: escape de HTML, validación de entradas.
- Uso de cache local o sessionStorage para búsquedas recientes.
- Separación clara de componentes, servicios y modelos.
- Puedes desplegar la app en algún servicio gratuito como GitHub Pages, Netlify o Vercel y agregar el enlace en el README.md.

## Tecnologías sugeridas

### Backend:

- Java 8 o superior
- Spring Boot
- Maven
- Swagger (opcional)
- Base de datos local o embebida (H2 o Postgres opcional)
- Docker (opcional)

### Frontend:

- Angular 14+
- TypeScript
- Angular Material / Bootstrap
- HTML5, SCSS/CSS3

Evaluaciones:

#### Perfil Backend

- Integración con APIs externas (servicios HTTP).
- Diseño de APIs REST limpias y bien estructuradas.
- Código limpio, mantenible y testeable.
- Manejo de errores y excepciones de forma clara y controlada.
- Registro de operaciones en base de datos (opcional).
- Despliegue en Docker (opcional)

#### Perfil Frontend

- Consumo correcto de APIs REST.
- Interfaz clara, responsiva y visualmente ordenada.
- Buenas prácticas de desarrollo frontend (componentes reutilizables, separación de responsabilidades).
- Manejo de errores y estados de carga en la UI.
- Seguridad básica en el cliente (validación y sanitización).

#### Perfil Fullstack

- Cumplimiento de todos los objetivos anteriores.
- Integración fluida entre frontend y backend.
- Organización y cohesión del código en ambos lados.
- Complejidad y completitud de la solución presentada.

Se estima entre **2 y 4 horas**. Si necesitas más tiempo, no hay problema, solo indícanos cuánto te tomó finalmente (inclúyelo en el `README.md`).

Si tienes cualquier pregunta técnica o necesitas aclaración sobre algún punto, no dudes en escribirnos. Estamos aquí para ayudarte.

¡Mucho éxito y gracias por participar!