



## Kata Practicing - Challenge 1 | Plataforma Básica de Selección de Practicantes

**Duración estimada Desarrollo:** 4 horas aproximadamente.

 **Reto: Plataforma Básica de Selección de Practicantes** [🔗](#)

 **Duración estimada: 4 horas** [🔗](#)

---

 **Descripción del Reto:** [🔗](#)

El objetivo es crear una **herramienta web sencilla** que permita a los practicantes registrar su información en un formulario y que los **analistas de selección** puedan revisarla. La aplicación debe tener una parte para que los practicantes carguen sus datos y otra donde los analistas puedan ver y decidir si el candidato es "viable" o "no viable".

Puedes usar **cualquier lenguaje o framework** que conozcas y herramientas gratuitas o con licencia académica. No es necesario que la aplicación esté desplegada en la nube, puede ser ejecutada de manera local.

---

 **Requisitos Técnicos:** [🔗](#)

♦ **Front-End:** [🔗](#)

- **Formulario de Registro para Practicantes:**

- Campos básicos:
  - Nombre completo
  - Correo electrónico
  - Carrera universitaria
  - Semestre actual
- Opción para subir archivos:
  - Hoja de vida (PDF)
- Verificación de campos obligatorios (por ejemplo, que el nombre no esté vacío).

- **Panel de Analista:**

- Ver la lista de practicantes registrados.
- Visualizar la información de cada practicante.
- Botón para marcar como "**Viable**" o "**No viable**".

♦ **Back-End:** [🔗](#)

- API sencilla que permita:
  - Guardar la información del practicante.

- Consultar la lista de practicantes.
  - Actualizar el estado del practicante (viable/no viable).
  - Base de datos local (puede ser **SQLite** o similar).
  - Almacenar archivos en una carpeta del servidor.
- 

### **Desafíos Adicionales (Opcional):** [🔗](#)

1. **Autenticación Básica:** Que solo los analistas puedan ver el panel (usuario y contraseña simples).
  2. **Descarga de Archivos:** Permitir que el analista descargue la hoja de vida desde el panel.
- 

### **Pasos de Implementación:** [🔗](#)

#### 1. **Diseño del Front-End:**

- Crear el formulario de registro con HTML y CSS.
- Agregar validación simple para campos obligatorios con JavaScript.
- Crear una página de listado para los analistas con botones de "Viable" y "No viable".

#### 2. **Desarrollo del Back-End:**

- Crear una API sencilla con Node.js (Express), Python (Flask), o Java (Spring Boot).
- Guardar los datos en una base de datos local (ej. SQLite).
- Almacenar los archivos en una carpeta del servidor.

#### 3. **Integración:**

- Conectar el formulario al backend para guardar la información.
- Mostrar la lista de practicantes en el panel del analista.

#### 4. **Documentación:**

- Escribir un archivo README con instrucciones para instalar y ejecutar el proyecto.
- Incluye capturas de pantalla si es posible.

### **El acompañamiento**

- Democratización del conocimiento
- Líder de equipo
- Crearemos un grupo en teams para resolver preguntas