




Kata Junior - Fullstack/Cloud | Portal de Capacitaciones Interactivo










Reto Junior – Portal de Capacitaciones Interactivo

Contexto

El **CoE de Desarrollo** quiere un portal web sencillo  donde los colaboradores puedan **consultar capacitaciones** y **guardar su progreso**. El sistema debe ser lo suficientemente simple para que un equipo junior pueda construirlo, pero funcional y organizado.

Consideraciones de Seguridad en la parte baja del reto.



Requerimientos Funcionales

1.  **Autenticación básica** (usuario/contraseña o login simulado).
 2.  **Consulta de capacitaciones** organizadas en módulos:
 -  **Fullstack**
 -  **APIs e Integraciones**
 -  **Cloud**
 -  **Data Engineer**
 3.  **Seguimiento básico de cursos**: marcar “iniciado” o “completado”.
 4.  **Insignia simple**: al terminar un curso.
 5.  **Administración mínima**: agregar/editar cursos.
-

Requerimientos Técnicos

- **Backend**: Node.js o Java (servicios REST básicos).
 - **Frontend**: Angular / React / HTML+JSP (interfaz simple).
 - **Base de datos**: SQLite / PostgreSQL / MongoDB.
 - **Despliegue**: local con Docker o nube gratuita (Heroku, AWS Free Tier, Render).
 - **Extras Free**: almacenamiento de insignias como imágenes estáticas.
-

Entregables

- Portal web navegable con login.
- Listado de capacitaciones por módulos .
- Marcar cursos como completados .

- Pantalla de perfil con insignias 🏆 e histórico.
- Documentación corta de instalación y uso.

Diagrama Visual Propuesto

```
1 flowchart TD
2   A[👤 Usuario] -->|Login| B[🌐 Portal Capacitaciones]
3   B --> C[📖 Ver Cursos por Módulo]
4   C --> D[✅ Marcar Curso Completado]
5   D --> E[🏆 Insignia Generada]
6   E --> F[📊 Perfil con Histórico]
7   B --> G[👤 Admin Agrega Cursos]
8
```

✅ Criterios de Evaluación

♦ Frontend (Interfaz de Usuario)

- Claridad y simplicidad en la navegación.
- Uso correcto de componentes (Angular/React/HTML).
- Validación básica de formularios (ej: login, marcar curso).
- Diseño responsivo y amigable (no importa que sea simple).

♦ Backend (Lógica de Negocio)

- Exposición de endpoints REST básicos (listar cursos, marcar completado, login simulado).
- Buen manejo de rutas y controladores.
- Manejo simple de errores (ej: curso no encontrado).

♦ Base de Datos

- Correcta definición de tablas/colecciones (usuarios, cursos, progreso).
- Uso de claves primarias/IDs para cada curso.
- Operaciones CRUD básicas implementadas.

♦ Buenas Prácticas

- Código organizado y comentado.
- Uso de Git para versionamiento.
- Estructura clara de carpetas (frontend / backend / BD).

♦ Extras (No obligatorios, suman puntos 🔥)

- Despliegue en un servidor gratuito (ej: AWS Free Tier, Render, Heroku).
- Notificaciones por correo (Mailhog, SendGrid Free).
- Docker Compose para levantar frontend + backend + BD.

Consideraciones de Seguridad

- No subir credenciales reales ni secretos a repositorios públicos.
- Usar datos simulados en todos los ejemplos.
- Versionar el código en repositorios personales.

- Se recomienda el uso de `.gitignore` para excluir archivos sensibles.

🔒 ¡Atención, equipo! Importante mensaje sobre seguridad en repositorios 🚨

¡Importante!

📢 ¡Atención, equipo! 🚀. General

Queremos recordarles algo **MUY IMPORTANTE** para mantener la seguridad de nuestra información y la del banco. Recientemente, tuvimos un incidente porque en una **Kata de desarrollo** se usó un repositorio real de la organización y accidentalmente se expuso un **Secret Key** en GitHub. 🙄

Esto representa un **riesgo grave** para la seguridad, por lo que necesitamos su ayuda para evitarlo en el futuro.

📌 **Recomendaciones clave:**

- ♦️ **🚫 No usen repositorios reales del banco en Katas.**
 - Las Katas son para práctica, ¡usen repositorios personales, no usar sandbox del banco pueden usar sandbox personales!
- ♦️ **🔒 Nunca suban credenciales, keys o información sensible.**
 - Si necesitan datos de prueba, usen valores ficticios (ej: `"token_ejemplo_123"`).
- ♦️ **✅ Verifiquen antes de hacer commit/push.**
 - Usen herramientas como `git-secrets` o `.gitignore` para evitar subir archivos críticos.
- ♦️ **🛡️ Si ven algo raro, repórtenlo de inmediato.**

💡 **Alternativas seguras para Katas:**

- **GitHub Personal:** Crear un repo en sus cuentas personales (no vinculado al banco).
- **Entornos de prueba:** Usar bases de datos o APIs mock (como Mockaroo, JSON-Server).
- **Datos falsos:** Siempre que sea posible, generen datos simulados.

☀️ ¡Juntos protegemos nuestra seguridad! ☀️

Este tipo de errores pueden tener consecuencias serias, pero con buenas prácticas los evitamos. ¡Gracias por su colaboración!

¡Éxitos con el Reto! 🚀

El acompañamiento

- Democratización del conocimiento