

Prueba de Conocimiento para Desarrollador Full Stack

Instrucciones Generales:

1. Responde a las preguntas y ejercicios con claridad y precisión.
2. Escribe código limpio, eficiente y bien documentado. Explica brevemente tus decisiones técnicas.
3. Asegúrate de que tu solución sea funcional y optimizada para manejar grandes volúmenes de datos.
4. Tiempo máximo de entrega: **Miércoles 16 de abril de 2025 a las 6:00 p.m.**

Criterios de Evaluación:

1. Claridad y estructura del código (20%) : El código debe ser fácil de leer, bien organizado y correctamente documentado.
2. Eficiencia y optimización (20%): Las soluciones deben ser eficientes y evitar redundancias innecesarias.
3. Funcionalidad (30%): El código debe cumplir con los requisitos y funcionar correctamente.
4. Conocimiento técnico (20%): Las respuestas deben demostrar un sólido entendimiento de los conceptos y tecnologías involucradas.
5. Solución de errores (10%): Capacidad para identificar, reproducir y corregir errores en código de otros desarrolladores.

Entrega:

Envía los archivos con tus soluciones en un repositorio de **GitHub** o como un archivo comprimido (.zip) al correo **luz.cardona@escueladidactica.com**. Incluye un archivo **README.md** con una breve descripción de tu enfoque y cualquier consideración adicional.

Parte 1: PHP (Backend) - API REST

Construye una API en PHP que permita gestionar tareas y usuarios. La API debe:

1. Conectarse a una base de datos MySQL.
2. Soportar las siguientes operaciones:
 - a. GET /users → Devuelve todos los usuarios.
 - b. POST /users → Agrega un nuevo usuario.
 - c. GET /tasks → Devuelve todas las tareas.
 - d. POST /tasks → Agrega una nueva tarea.
 - e. PUT /tasks/{id} → Marca una tarea como completada.
 - f. DELETE /tasks/{id} → Elimina una tarea.
3. Retornar las respuestas en formato JSON.
4. Implementa manejo básico de errores (por ejemplo, si un ID no existe).

Estructura de la Base de Datos (MySQL):

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE  
);  
  
CREATE TABLE tasks (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    title VARCHAR(255) NOT NULL,  
    completed BOOLEAN DEFAULT 0,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES users(id)  
);
```

Pregunta Adicional:

Explica cómo podrías mejorar el rendimiento de esta API utilizando caché o índices en la base de datos.

Parte 2: React (Frontend) - Aplicación de Lista de Tareas

Desarrolla una aplicación en React que interactúe con la API creada anteriormente. La aplicación debe:

1. Mostrar una lista de usuarios obtenida desde GET /users.
2. Permitir seleccionar un usuario y mostrar sus tareas (GET /tasks filtrando por user_id).
3. Permitir agregar nuevas tareas (POST /tasks).
4. Marcar tareas como completadas (PUT /tasks/{id}).
5. Eliminar tareas (DELETE /tasks/{id}).

Requisitos de la Interfaz:

- Lista de Usuarios: Muestra los nombres de los usuarios en un menú desplegable.
- Lista de Tareas: Muestra las tareas del usuario seleccionado con casillas de verificación para marcarlas como completadas.
- Formulario para Agregar Tareas: Un campo de entrada y un botón para añadir tareas.
- Botón de Eliminar: Cada tarea debe tener un botón para eliminarla.

Extras Opcionales:

- Mejorar la interfaz con estilos CSS modernos.
- Agregar filtros (Todas, Completadas, Pendientes).
- Manejar errores mostrando mensajes cuando la API falla.

Parte 3: SQL (Base de Datos) - Optimización y Consultas

Escribe una consulta SQL optimizada que obtenga todas las tareas pendientes de un usuario específico, ordenadas por fecha de creación descendente.

Explica qué es un índice compuesto en SQL y cómo podría beneficiar el rendimiento de la consulta anterior.

Parte 4: JavaScript (Lógica Frontend)

Escribe una función en JavaScript llamada **filterTasks** que reciba un array de tareas y un filtro ("all", "completed", "pending") y devuelva un array con las tareas filtradas según el criterio.

Ejemplo de uso:

```
const tasks = [
  { id: 1, title: "Task 1", completed: true },
  { id: 2, title: "Task 2", completed: false }
];
console.log(filterTasks(tasks, "pending"));
// Salida: [{ id: 2, title: "Task 2", completed: false }]
```

Parte 5: HTML y CSS (Interfaz de Usuario)

Crea una página HTML simple con un formulario que contenga los siguientes campos:

- Nombre (texto)
- Correo electrónico (email)
- Botón de envío

Aplica estilos CSS para que el formulario tenga un diseño limpio y moderno. Asegúrate de que sea responsivo.

Parte 6: Motores de Inteligencia Artificial (Innovación)

Investiga y describe brevemente qué es TensorFlow.js y menciona un caso de uso potencial en una aplicación web.

Propón una idea innovadora para integrar inteligencia artificial en un proyecto web. Explica cómo beneficiaría a los usuarios.

Parte 7: Moodle y Error en Plugin Estado de Finalización

Soluciona el error en el plugin estado de finalización siguiendo los siguientes pasos.

1. Instala Moodle 4.5:

Descarga e instala Moodle 4.5

<https://download.moodle.org/download.php/direct/stable405/moodle-latest-405.zip>

en un servidor local (puedes usar **XAMPP**, **WAMP** o **Docker**). Configura la base de datos **MySQL** necesaria para Moodle, documenta los pasos seguidos para la instalación en el archivo **README.md**.

2. Instala el plugin completion_progress:

Descarga el plugin tipo bloque **completion_progress** que se encuentra en la siguiente carpeta

<https://drive.google.com/drive/folders/12GWaBHv130pXR4MXQZ6w9FZA0YGrTkoC>

e instala en tu instancia de Moodle 4.5 local como una extensión adicional.

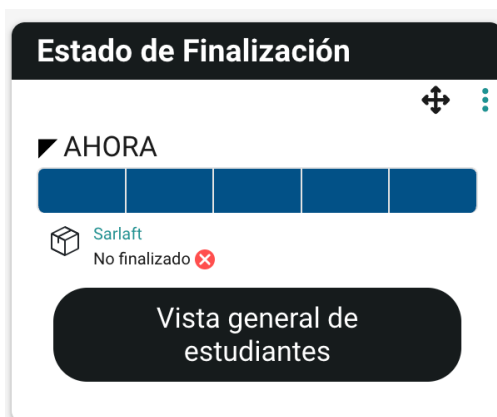
3. Restaura el curso demo

Restaura el curso demo en tu instancia de Moodle local usando el archivo **curso-demo.mbz** que se encuentra en la carpeta:

<https://drive.google.com/drive/folders/12GWaBHv130pXR4MXQZ6w9FZA0YGrTkoC>

4. Soluciona el error

Ingresa al curso **Sarlaft** y ubica el bloque **Estado de finalización** y has clic en el botón **vista general de estudiantes**



el cual mostrará **Error al leer de la base de datos**, encuentra y corrige el error,

Sarlaft

Curso Configuración Participantes Calificaciones Informes Más ▾

Error al leer de la base de datos

Más información sobre este error

Información de depuración: Unknown column 'r.capabilite' in 'SELECT'
SELECT DISTINCT r.id, r.name, r.shortname, r.archetype, r.sortorder, r.capabilite
FROM mdl_role r, mdl_role_assignments ra
WHERE ra.contextid = ?
AND r.id = ra.roleid
ORDER BY r.sortorder
[array (
0 => 2183,
)]
Error code: dmlreadexception

Documenta el error encontrado, la causa raíz y la solución implementada. Incluye capturas de pantalla que muestren el progreso los estudiantes después de aplicar la solución.

**Buena suerte, y esperamos ver tus habilidades en acción.
¡Nos vemos en la entrevista técnica!**