



Es hora de que pongas en práctica todo lo aprendido. 🧐

Este apartado tiene el objetivo de ayudarte a reforzar y aplicar los conocimientos adquiridos durante el curso en Apex y Lightning Web Components (LWC), que son fundamentales para el desarrollo en la plataforma Salesforce. Este ejercicio está diseñado para ayudarte a consolidar tu comprensión de cómo el código backend y el frontend pueden trabajar juntos para crear aplicaciones interactivas y dinámicas en Salesforce.

Más adelante conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 😊

¡Manos a la obra!

1. Desafío 🎯

El desafío consistirá en desarrollar una aplicación en Salesforce que permita a los usuarios gestionar una lista de tareas. La aplicación debe contener un componente LWC que deberá interactuar con Apex para realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en una lista de tareas almacenadas.

Backend con Apex:

- ➡ Modelo de Datos: Define una clase Tarea en Apex que incluya propiedades como *id*, *titulo*, *descripcion*, y *completado*.
- ➡ Manipulación de colecciones: Utiliza listas y mapas en Apex para gestionar las tareas. Crea métodos para *añadir*, *eliminar*, y *actualizar* tareas en la colección.
- ➡ Métodos de Apex: Desarrolla métodos expuestos a LWC para obtener todas las tareas, añadir una nueva tarea, marcar una tarea como completada y eliminar una tarea.

Frontend con LWC:

- ➡ Estructura del Componente: Crea un componente LWC que muestre la lista de tareas. Incluye campos para ingresar el título y la descripción de nuevas tareas.

- ↳ **Interacción Usuario:** Permite al usuario añadir nuevas tareas mediante un formulario y botones para marcar las tareas como completadas o eliminarlas.
- ↳ **Data Binding y Reactividad:** Asegúrate de que el componente reacciona adecuadamente a los cambios en el estado de las tareas, actualizando la interfaz de usuario en tiempo real cuando se agregan, modifican o eliminan tareas.

Estructura del Proyecto

→ Directorios y Archivos Principales

- **force-app/main/default/classes/**
 - **TareaController.cls:** Clase Apex que maneja la lógica de negocio para las operaciones CRUD de las tareas.
 - **TareaControllerTest.cls:** Clase de pruebas para TareaController, asegurando que todos los métodos funcionan correctamente.
- **force-app/main/default/lwc/**
 - ↳ **gestorTareas/:** Directorio para el componente LWC que gestiona tareas.
 - ↳ **gestorTareas.html:** Archivo HTML para la estructura del componente.
 - ↳ **gestorTareas.js:** Archivo JavaScript que maneja la lógica del componente.
 - ↳ **gestorTareas.css:** Hoja de estilos CSS para el diseño del componente.

→ **Desarrollo de la Clase Apex (TareaController.cls):**

- ↳ Métodos para obtener la lista de tareas, añadir una nueva tarea, actualizar y eliminar tareas.
- ↳ Métodos anotados con `@AuraEnabled` para permitir su llamada desde LWC.

→ **Desarrollo del Componente LWC (gestorTareas):**

- ↳ **gestorTareas.html:** Estructura del componente, incluyendo

formularios para añadir tareas y una lista que muestra las tareas existentes.

↪ **gestorTareas.js:** Controladores de eventos y métodos para interactuar con la API de Apex, actualizar la UI basada en el estado de las tareas.

↪ **gestorTareas.css:** Estilos específicos para el componente.

2. ¿Dónde se lleva a cabo? 🧑💻

→ Puedes crear una org de Salesforce usando este link: <https://developer.salesforce.com/signup>

→ Utiliza IDEs como Salesforce Developer Console o Visual Studio Code con extensiones Salesforce para mejorar tu eficiencia en la codificación.

3. Tiempo de dedicación ⌚

3 horas

4. Recursos 🛠️

1. Google

Búsqueda de Soluciones y Códigos de Ejemplo: Google es excelente para buscar soluciones específicas a errores, ejemplos de código y mejores prácticas. Usa palabras clave precisas relacionadas con tu problema o la funcionalidad que estás implementando.

Foros y Comunidades: Utiliza Google para encontrar y participar en foros y comunidades de desarrolladores, como Stack Overflow, Developer Forums de Salesforce, y otros. Estos lugares son valiosos para obtener consejos de otros desarrolladores que pueden haber enfrentado problemas similares.

2. ChatGPT

Explicaciones y Aprendizaje: ChatGPT puede explicar conceptos complejos, ofrecer tutoriales paso a paso y responder preguntas específicas sobre programación y mejores prácticas de desarrollo.

Generación de Código y Revisión: Puede ayudarte a generar fragmentos de código para problemas comunes o revisar el código que has escrito para sugerir mejoras o corregir errores.

3. Documentación Oficial de Salesforce:

Accede siempre a la documentación oficial de Salesforce para obtener la información más precisa y actualizada sobre el desarrollo en la plataforma.

<https://developer.salesforce.com/docs>

5. ⚠ Condición

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante puedes compartir tus resultados con el resto de los bootcampers y construir conocimiento en conjunto.