



# Proyecto 1er Parcial: Desarrollo de Aplicaciones Web con Angular

## Temas según número de grupo:

- 1. Catálogo de Servicios para Empresas de Eventos**  
Una aplicación web que permite a pequeñas empresas de eventos registrar y mostrar sus servicios, como salones, mantelería, música, animación, decoración, etc. Los usuarios pueden explorar el catálogo, seleccionar los servicios que desean para su evento, obtener un presupuesto referencial y realizar un pedido directamente a la empresa.
- 2. Plataforma de Servicios de Oficios Locales**  
Una aplicación web que permite a profesionales de oficios como gasfiteros, electricistas, albañiles y servicios de limpieza registrar sus servicios y crear perfiles. Los usuarios pueden buscar y filtrar por tipo de oficio, ubicación, etc. Además, los clientes pueden contactar directamente a los profesionales para solicitar servicios, calificar su trabajo y dejar reseñas basadas en su experiencia. La plataforma facilita la conexión entre quienes necesitan servicios de oficios y los profesionales disponibles
- 3. Catálogo de Servicios Turísticos**  
Una aplicación web que permite a pequeñas empresas del sector turístico registrar y mostrar sus servicios, como excursiones, guías turísticos, transporte, alojamiento y actividades recreativas. Los usuarios pueden explorar el catálogo, seleccionar los servicios que desean para su viaje, obtener un presupuesto referencial y realizar una reserva directamente con la empresa. La plataforma también puede incluir reseñas y calificaciones de otros usuarios, ayudando a los viajeros a tomar decisiones informadas sobre sus opciones turísticas.
- 4. Marketplace de Productos para Fiestas Infantiles**  
Una aplicación web que permite a los usuarios comprar una variedad de productos para fiestas infantiles, como decoraciones, invitaciones, disfraces, juegos y suministros para actividades. Los usuarios pueden explorar categorías, filtrar por temática (cumpleaños, bautizos, fiestas temáticas) y realizar un pedido.
- 5. Aplicación de Intercambio de Libros Usados**  
Un sitio web para facilitar el intercambio de libros usados entre personas. Los usuarios pueden buscar libros por área, categoría, establecer contacto directo y arreglar la logística de intercambio. Incluye secciones de recomendaciones de libros y comentarios de los usuarios.
- 6. Sistema de Gestión de Recursos Comunitarios**  
Una aplicación web que permita a los vecinos de una comunidad o barrio gestionar recursos compartidos, como instalaciones (salones comunales, áreas deportivas) y herramientas (equipos de jardinería, mesas de eventos). Los usuarios pueden consultar la disponibilidad de recursos, reservarlos y visualizar el historial de uso, optimizando la administración de bienes comunes.
- 7. Plataforma de Coordinación de Reciclaje Domiciliario**  
Una aplicación para que los hogares puedan registrarse en un sistema de recolección de reciclaje y gestionar el tipo de materiales que generan (plásticos, metales, papel, vidrio). Las empresas de reciclaje pueden consultar la cantidad y tipo de materiales disponibles en la comunidad y planificar rutas de recolección.
- 8. Plataforma de Registro de Iniciativas de Preservación Ambiental**  
Una aplicación donde los usuarios puedan registrar y unirse a iniciativas de preservación del medio ambiente, como limpieza de playas, reforestación o campañas de reducción de plástico. La plataforma permite organizar actividades, compartir avances de cada iniciativa mediante reportes y fotos. Los usuarios pueden interactuar y visualizar el progreso general.
- 9. Portal de Servicios Postventa y Soporte al Cliente**  
Una aplicación donde los clientes pueden registrar solicitudes de soporte, garantías o devoluciones, y el negocio puede responder y hacer seguimiento a cada caso. Permite a las empresas gestionar el servicio postventa y brindar soporte



## 10. Plataforma de Donaciones y Campañas Solidarias

Una aplicación web donde los usuarios pueden crear y unirse a campañas de donación para apoyar a personas o comunidades en necesidad. Cada campaña incluye detalles como la ubicación de recolección, los tipos de donaciones aceptadas (ropa, alimentos, juguetes, etc.) y la fecha límite para contribuir. Las personas pueden donar. Los usuarios pueden explorar campañas activas, informarse sobre los objetivos y beneficiarios de cada una, y decidir en cuáles desean participar.

### Instrucciones Generales

Este proyecto tiene como objetivo que cada integrante del equipo desarrolle un componente CRUD (Crear, Leer, Actualizar, Eliminar) como parte del modelo de negocio asignado. Además, se emplearán los frameworks **Angular** y **Angular Material**, con la opción de utilizar **Bootstrap** para mejorar el diseño y la presentación visual. Cada componente CRUD deberá implementar un servicio y un modelo que interactúe con un archivo JSON local que contenga datos de prueba. Esto permitirá simular una interacción con una fuente de datos sin necesidad de una base de datos real.

### 1. Requisitos del Proyecto

#### 1. Frameworks y Herramientas:

- Angular (versión más reciente).
- Angular Material para la interfaz de usuario (UI).
- Opcional: Bootstrap o Tailwind para diseño responsivo y estilos adicionales.

#### 2. Inicialización del Proyecto: (evaluación grupal)

- Crear el proyecto Angular standalone (componentes no módulos) e instalar Angular Material.

#### 3. Creación de Estructura del Proyecto: (evaluación grupal)

- Crear una estructura de carpetas que permita organizar los componentes, servicios y otros archivos.
- Utilice la siguiente estructura de carpetas:

```
src/
├── app/
│   ├── models/           #carpeta con los modelos a usar en cada CRUD
│   ├── services/         # Servicios y clases compartidas
│   ├── components/       # componentes
│   │   ├── shared/       # Componentes reutilizables
│   │   └── <nombre_componente>/ # carpeta para cada componente CRUD
│   ├── ...
│   ├── app.routes.ts
│   └── app.component.ts
```

#### 4. Componentes CRUD (evaluación individual):

- **Cada integrante** debe crear un componente CRUD relacionado con una entidad específica del modelo de negocio asignado.
- Cada componente debe incluir la funcionalidad de **crear, buscar, actualizar y eliminar** elementos.
- Siga la estructura:
  - **Formulario** para crear y editar registros, el formulario debe tener al menos 4 campos de al menos 3 tipos diferentes (caja de texto, combo, checkbox, etc), adicionales a los botones.
  - **Vista de tabla** para listar y buscar los registros existentes.
  - **Funciones de edición y eliminación.**
- Implementar un **servicio y un modelo** para cada componente CRUD. El servicio se encargará de la interacción con un archivo JSON local que contendrá datos de prueba y gestionará las operaciones CRUD simuladas. Los archivos .json se ubicarán en la carpeta **public/json**
- Si la funcionalidad del componente permite visualizar datos sin la posibilidad de modificarlos, entonces deberá crearse un componente adicional solo de visualización para dicha entidad.



5. **Ruteo: (evaluación grupal):**

- Configurar el **ruteo** en la aplicación para permitir una navegación fluida entre las diferentes secciones y componentes CRUD.
- Modificar el archivo de ruteo principal para que permita agrupar todas las rutas y facilite la navegación entre las vistas.

6. **Componentes Reutilizables: (evaluación grupal):**

- Utilizar al menos **dos componentes reutilizables** en el proyecto, que puedan ser empleados en varias partes de la aplicación. Asegúrese de cada componente reutilizable sea utilizado al menos dos veces dentro del proyecto. Ejemplos incluyen:
  - **Ventana de diálogo:** Para confirmar acciones como eliminación o mostrar información importante.
  - **Tabla reutilizable:** Para mostrar listas de datos en cada componente CRUD, con opciones para ordenar y paginar.
  - **Notificaciones o alertas:** Para dar feedback al usuario sobre acciones realizadas (éxito, error, etc.).
  - **Formulario de búsqueda:** Para facilitar la búsqueda de registros en cada componente CRUD.
  - **CARDS para visualización de datos:** Para mostrar listas de datos atractiva para el usuario final.

7. **Estilos y Consistencia: (evaluación individual y grupal):**

- Utilizar Angular Material y, opcionalmente, Bootstrap, Tailwind para asegurar un diseño consistente y atractivo.
- Todos los componentes deben respetar un diseño homogéneo, usando los mismos colores, fuentes y estilos de botones definidos en la configuración de Angular Material y Bootstrap, si se usa.

8. **Validaciones y Manejo de Errores: (evaluación individual)**

- Incluir validaciones de formularios en cada componente CRUD para asegurar que los datos ingresados son válidos antes de guardarlos o actualizarlos.
- Manejar los errores en el frontend, mostrando mensajes claros y amigables para el usuario en caso de errores en las operaciones CRUD.

## 2. Entrega y Documentación

- Enlace a carpeta que contenga:
  - Un documento pdf con: tabla de nombres de integrantes que realizaron el proyecto conjuntamente con los nombres de los componentes realizados por cada autor, enlace para descarga de la carpeta del proyecto.

Ejemplo de tabla de integrantes:

Integrante	Componente CRUD	Componente de solo Visualización de datos	Componente Reutilizable que utiliza
Juan Pérez	crud-productos	lista-productos	ventana-dialogo, card-general
Marlón Solís	crud-usuarios		ventana-dialogo
Julio López	No participo del proyecto		

### Evaluación:

Nota de proyecto 2,3 puntos por el cumplimiento del 100% de lo indicado en el documento.