Universidad Rafael Landívar
Análisis y Diseño de Sistemas
Ing. José Andrés De la Cruz Franco
Laboratorio

Proyecto Final:
Gestión de Reservas para Clases y Talleres de Cocina "MasterCook Academy"

---

1. Narrativa del Proyecto

MasterCook Academy es una prestigiosa escuela de cocina que ofrece talleres presenciales y virtuales en diversas especialidades: cocina internacional, repostería, cocina saludable y técnicas avanzadas.

Actualmente, la gestión de inscripciones se realiza de forma manual, lo que genera ineficiencias, sobrecupo y mala experiencia para estudiantes y administradores.

La dirección ha decidido modernizar sus procesos mediante el desarrollo de una plataforma web que permita a los estudiantes gestionar sus reservas y a la administración controlar talleres, cupos e instructores de manera eficiente.

El desafío no solo es funcional, sino también arquitectónico: se requiere una solución moderna, escalable, basada en microservicios, contenedores Docker y desplegada en la nube.

2. Objetivo General

Diseñar, documentar e implementar una plataforma web para la gestión de reservas de talleres de cocina, aplicando principios de:

- Análisis y diseño de sistemas.
- Arquitectura basada en microservicios.
- Buenas prácticas de desarrollo y despliegue en la nube.

3. Tecnologías Definidas

Para garantizar consistencia tecnológica en el desarrollo:

- **Backend:** Python + Flask
- **Frontend:** React.js
- **Base de Datos:** MySQL
- **Contenedores:** Docker

4. Definición e Implementación de la Arquitectura

4.1 Definición de la Arquitectura

Cada equipo deberá diseñar un **Diagrama de Arquitectura de Microservicios** que refleje claramente la estructura del sistema.

El diagrama debe contemplar:

- Separación clara de responsabilidades: frontend, backend y base de datos.
- Uso explícito de contenedores Docker.
- Flujo de comunicación entre servicios.
- Inclusión de servicios externos (opcional).
- Justificación de todas las decisiones arquitectónicas (simplicidad, escalabilidad, mantenibilidad).

4.2 Implementación de la Arquitectura

Una vez definida:

- Se desarrollará el sistema siguiendo estrictamente la arquitectura propuesta.
- Se contenerizarán los servicios usando Docker.
- Se desplegará el sistema en un servicio de nube (gratuito como inicio de sesión con Gmail).
- Cualquier desviación de la arquitectura original deberá documentarse y justificarse.

5. Historias de Usuario

A continuación, se detallan las historias principales que representan el alcance mínimo del sistema.

Historia 1: Registro de Usuario

Descripción:
Como estudiante interesado en talleres de cocina,
quiero registrarme en la plataforma
para poder reservar talleres y gestionar mis datos personales.

Criterios de Aceptación:

- Registro con nombre, correo y contraseña.
- Validación de correo único.
- Contraseña con mínimo 8 caracteres.
- Confirmación visual tras registrarse.
- Almacenamiento seguro en MySQL.

Historia 2: Autenticación (Login)

Descripción:
Como usuario registrado,
quiero iniciar sesión en la plataforma para acceder a mi perfil y reservar talleres.

Criterios de Aceptación:

- Login con correo y contraseña válidos.
- Manejo adecuado de credenciales inválidas.
- Redirección automática tras autenticación exitosa.
- Mantener la sesión activa.

Historia 3: Exploración de Talleres Disponibles

Descripción:
Como estudiante,
quiero visualizar los talleres disponibles con sus detalles
para elegir el que más me interese.

Criterios de Aceptación:

- Listado visible con nombre, categoría, fecha, precio y cupo disponible.
- Filtros por categoría de taller.
- Mensaje adecuado si no hay talleres disponibles.
- Información proveniente de la API Flask.

Historia 4: Reserva de Taller

Descripción:
Como estudiante,
quiero poder reservar un taller
para asegurar mi lugar en la clase deseada.

Criterios de Aceptación:

- Posibilidad de reservar un taller.
- Validación de disponibilidad de cupo.
- Registro correcto en base de datos.
- Confirmación visual de la reserva.
- Bloqueo de reservas duplicadas.

Historia 5: Simulación de Pago

Descripción:
Como estudiante,
quiero simular el pago de mi taller
para completar el proceso de reserva.

Criterios de Aceptación:

- Pantalla de simulación de pago tras reservar.
- Botón para confirmar el pago.
- Cambio de estado de la reserva a “Pagado”.
- Mostrar resumen actualizado de la operación.
- Implementar como un servicio Mock

Historia 6: My Booking (Mis Reservas)

Descripción:
Como estudiante registrado,
quiero acceder a una sección llamada "My Booking" donde pueda ver todas mis reservas actuales y pasadas,
para poder gestionar mis talleres de forma organizada y verificar detalles importantes de mis clases.

Criterios de Aceptación:

- El usuario debe poder visualizar un listado de todas sus reservas activas y pasadas.
- Cada reserva debe mostrar:
    - Nombre del taller.
    - Fecha y hora.
    - Estado del pago (Pagado o Pendiente).
    - Estado de la reserva (Confirmada, Cancelada, Completada).
- El usuario debe poder filtrar o buscar entre sus reservas.
- El usuario debe poder ver detalles completos de cada reserva haciendo clic o tocando una tarjeta o fila.
- El diseño debe ser claro, con estado visual (colores o etiquetas) para el pago y el estatus del taller.

6. Criterios de Evaluación

La evaluación del proyecto final se realizará en dos partes:

6.1 Cumplimiento Funcional de las Historias de Usuario (50%)

- **Descripción:** El cumplimiento completo de todas las historias de usuario representará el **50%** de la nota total.
- **Condición:**
  - Si el 100% de las historias no está desarrollado y funcional, se pierde automáticamente este 50%.
  - Esto simula la sanción de un cliente por incumplimiento de alcance y tiempos definidos.
- **Verificación:** Se evaluará en vivo el correcto funcionamiento de cada historia.

6.2 Defensa Oral del Proyecto (50%)

- **Descripción:** El otro **50%** será evaluado mediante una **defensa oral individual y grupal** (simulación de un examen privado técnico).
- **Dinámica:**
  - Cada integrante deberá defender diferentes aspectos: arquitectura, decisiones de diseño, implementación.
  - Se evaluarán: dominio técnico, claridad, profundidad de respuestas y manejo de imprevistos.

7. Observaciones Finales

- El proyecto debe seguir la estructura y tecnologías aquí establecidas.
- Se permite agregar mejoras o nuevas funcionalidades **siempre y cuando respeten** la arquitectura y buenas prácticas.
- Se valorará el esfuerzo adicional que demuestre innovación, buenas prácticas de programación, y experiencia de usuario mejorada.

8. Paleta de Colores para MasterCook Academy

Propósito	Color	Hexadecimal	Descripción	Ejemplo
Primario	Salsa Tomato	#D94F4F	Rojo suave, cálido, amigable	
Secundario	Creamy Vanilla	#FFF3E2	Crema neutro, balance de fondo	
Acento	Olive Green	#6B8E23	Verde oliva natural, asociable a cocina	
Texto Principal	Charcoal Gray	#333333	Gris oscuro, alta legibilidad	
Texto Secundario	Ash Gray	#666666	Gris medio para descripciones	
Fondo Claro	Snow White	#FAFAFA	Blanco cálido para evitar pure white harshness	

Anexos

Logo:

MASTERCOOK
ACADEMY

Prototipos

Referencias:

Tema	Link de referencia
Buenas prácticas de React	https://react.dev/learn
Cómo consumir una API REST en React	https://www.freecodecamp.org/espanol/news/como-consumir-rest-apis-en-react-guia-para-principiantes/
Introducción a Axios para llamadas HTTP en React	https://axios-http.com/docs/intro
Cómo crear un servicio mock en frontend (JSON Server)	https://github.com/typicode/json-server
Qué es y cómo usar un Mock API rápido	https://mockapi.io/
Introducción sencilla a servicios RESTful	https://restfulapi.net/