**Universidad de San Carlos de Guatemala**

**Facultad de Ingeniería**

**Laboratorio Software Avanzado**

**Escuela Vacaciones diciembre 2024**

****

**PRACTICA 1**

**Ana Denisse Figueroa Marroquín**

**Carnet: 201801143**

**Requerimientos funcionales:**

1. Gestión de Reservas:

Permitir al usuario reservar clases grupales disponibles según el horario ofrecido.

Mostrar un calendario interactivo con horarios y disponibilidad de clases.

Enviar confirmaciones de reserva por correo electrónico o notificaciones en la plataforma.

1. Gestión de Pagos:

Procesar pagos mediante múltiples formas.

Generar comprobantes de pago electrónicos.

Implementar reembolsos en caso de cancelaciones según políticas del negocio.

1. Administración de Horarios:

Permitir a los maestros gestionar y actualizar sus horarios disponibles.

Sincronizar los horarios de los maestros con las reservas para evitar conflictos.

Notificar a los maestros sobre nuevas reservas o cancelaciones.

**Requerimientos no funcionales:**

1. Rendimiento**:**  
   La plataforma debe manejar al menos 500 usuarios simultáneos sin degradación notable en el desempeño.
2. Escalabilidad**:**  
   Diseñar la plataforma para que pueda ampliarse a medida que crezca el número de usuarios y clases ofrecidas.
3. Seguridad**:**  
   Proteger los datos de usuarios mediante cifrado en tránsito y en reposo.
4. Usabilidad**:**  
   La interfaz debe ser intuitiva y fácil de usar para usuarios con distintos niveles de experiencia tecnológica.

**Descripción de la solución**

El negocio es una plataforma de reservas para clases del idioma aleman.

Esta plataforma permite gestionar al usuario:

* Reservas de clases grupales.
* Gestión de Pago.
* Administración de horarios de maestros.
* Reportes de actividad.

**Arquitectura Monolítica actual en el negocio incluye:**

* Interfaz de Usuario:

Página web con formularios para registro de usuarios, pagos y visualización de horarios.

* Capa de Negocio:

Gestión de usuarios.

Reservas y control de capacidad.

Procesamiento de pagos.

* Capa de Acceso a Datos:

Base de datos centralizada que almacena usuarios, horarios y pagos.

**Diagrama de la Arquitectura**

**A diagram of a computer

Description automatically generated**

**Separación de Servicios:**

* + - Cada servicio es independiente.
    - Los servicios se comunicarán a través de APIs REST.

Descripción de Funcionalidades por servicio:

Servicio de Usuarios:

* + - API para CRUD (Create, Read, Update, Delete) de usuarios.
    - Retroalimentación.

Servicio de Clases:

* + - Registro y control de clases.
    - Integración con el servicio de horarios.

Servicio de Pagos:

* + - Gestión de pagos.
    - Registro de transacciones y reportes de pagos.

Servicio de Horarios:

* + - CRUD de horarios.
    - Gestión de clases a maestros.

**Diseño de la Base de Datos**

**A screenshot of a computer

Description automatically generated**

**Contratos de Microservicios**

Ms1 Usuario: Registro

A screenshot of a computer

Description automatically generated

Ms1 Usuario: Ver usuarios

A screenshot of a computer

Description automatically generated

Ms1 Usuario: Editaar usuarios

A screenshot of a computer

Description automatically generated

Ms1 Usuario: Eliminar usuarios

A screenshot of a computer

Description automatically generated

Ms2 Clases: Registrar clases

A screenshot of a computer

Description automatically generated

Ms2 Clases: Ver clases

A screenshot of a computer

Description automatically generated

Ms2 Clases: Reservar clases

A screenshot of a computer

Description automatically generated

Ms3 Pagos: Registrar pago

A screenshot of a computer

Description automatically generated

Ms3 Pagos: Ver pagos

A screenshot of a computer

Description automatically generated

Ms4 Horarios: Añadir horario

A screenshot of a computer

Description automatically generated

Ms4 Horarios: Ver horarios

A screenshot of a computer

Description automatically generated