

**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.

### **2. Gestión de Pagos:**

Procesar pagos mediante múltiples formas.  
Generar comprobantes de pago electrónicos.  
Implementar reembolsos en caso de cancelaciones según políticas del negocio.

### **3. 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 alemán.

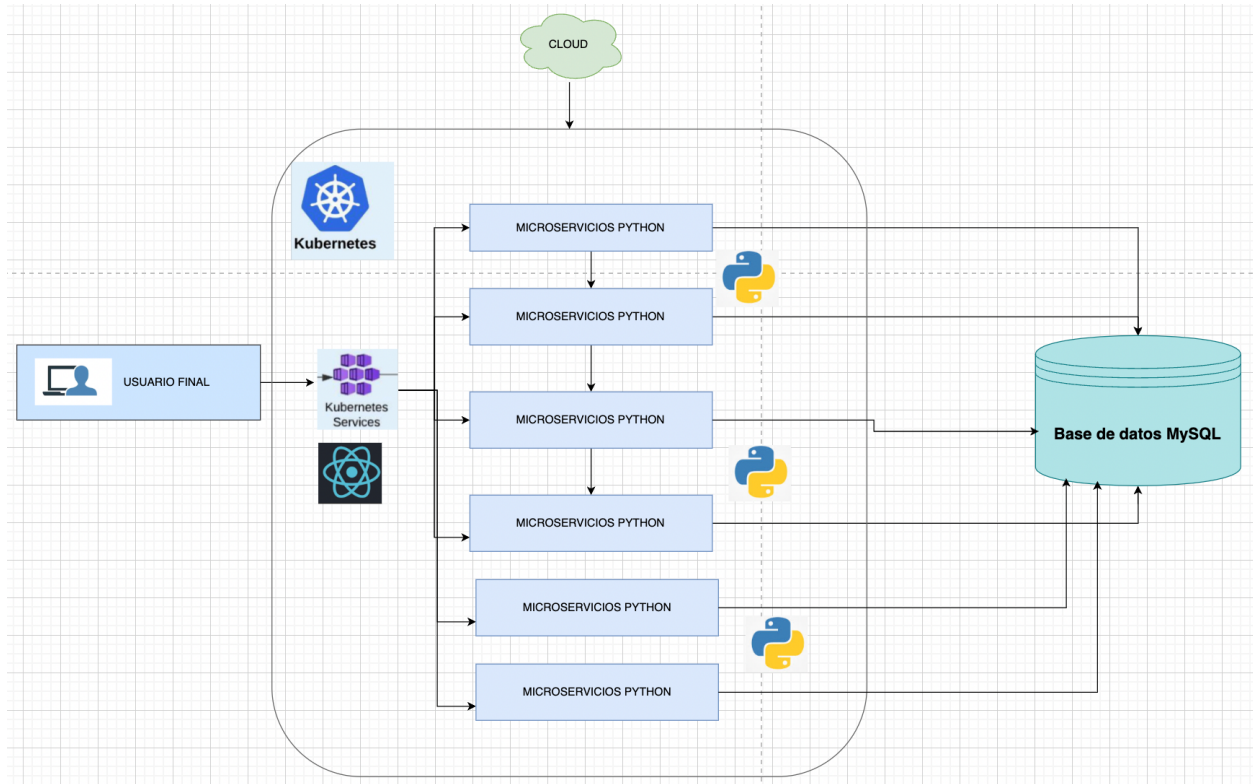
Esta plataforma permite gestionar al usuario:

- 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



## **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.

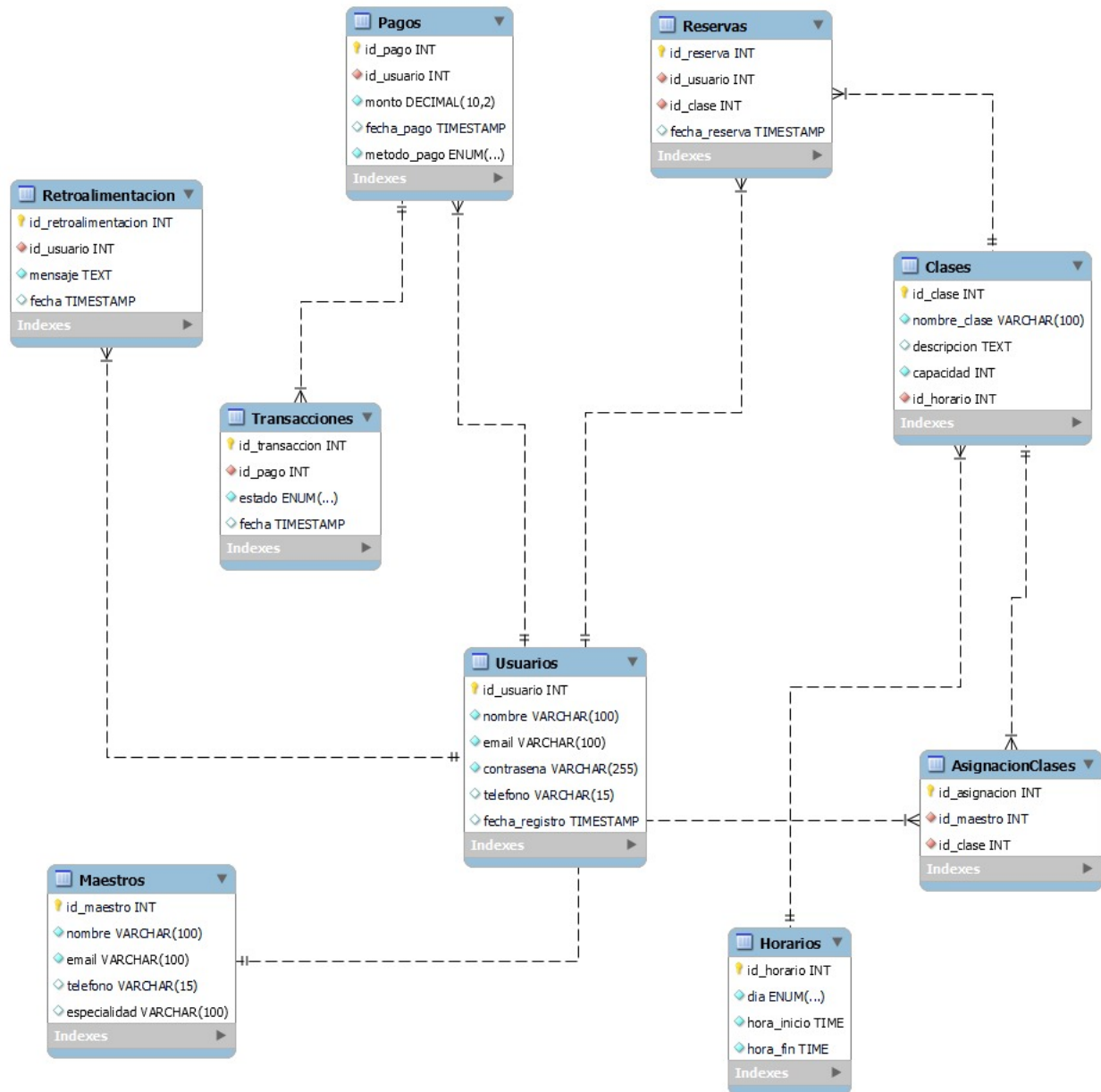
Servicio de Pagos:

- 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



# Contratos de Microservicios

## Ms1 Usuario: Registro

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nlp.io/ms1/usuarios/`
- Method:** `POST`
- Body (raw):**

```
1 {
2   "nombre": "Marco de León",
3   "email": "marco@example.com",
4   "contrasena": "123456",
5   "telefono": "33838"
6 }
7
```
- Status:** `201 CREATED` (95 ms, 184 B)
- Body (JSON):**

```
1 {
2   "mensaje": "Usuario creado exitosamente"
3 }
```

## Ms1 Usuario: Ver usuarios

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nlp.io/ms1/usuarios/`
- Method:** `GET`
- Status:** `200 OK` (87 ms, 448 B)
- Body (JSON):**

```
1 [
2   {
3     "email": "john3.doe@example.com",
4     "fecha_registro": "1234567890",
5     "id_usuario": 3,
6     "nombre": "John Doe3",
7     "telefono": "123456"
8   },
9   {
10    "email": "marco@example.com",
11    "fecha_registro": "33838",
12    "id_usuario": 4,
13    "nombre": "Marco de León",
14    "telefono": "123456"
15  }
16 ]
```

## Ms1 Usuario: Editaar usuarios

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nip.io/ms1/usuarios/3`
- Method:** `PUT`
- Body (JSON):**

```
1 {
2   "nombre": "John Updated3",
3   "email": "john3.updated@example.com",
4   "telefono": "0987654321"
5 }
6
```
- Response (JSON):**

```
1 {
2   "mensaje": "Usuario actualizado exitosamente"
3 }
```
- Status:** `200 OK`, `94 ms`, `184 B`

## Ms1 Usuario: Eliminar usuarios

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nip.io/ms1/usuarios/3`
- Method:** `DELETE`
- Response (JSON):**

```
1 {
2   "mensaje": "Usuario eliminado exitosamente"
3 }
```
- Status:** `200 OK`, `94 ms`, `182 B`



## Ms2 Clases: Registrar clases

The screenshot shows a REST client interface with the URL `http://sap2.35.192.41.172.nlp.io/ms2/clases/`. The method is **POST**. The request body is a JSON object:

```
1 {
2   "nombre_clase": "Clase Alemán Intermedio B1",
3   "descripcion": "Avanzado al idioma alemán",
4   "capacidad": 40,
5   "id_horario": 4
6 }
7
```

The response status is **201 CREATED** with a response time of 102 ms and a body size of 186 B. The response body is a JSON object:

```
1 {
2   "mensaje": "Clase registrada exitosamente"
3 }
```

## Ms2 Clases: Ver clases

The screenshot shows a REST client interface with the URL `http://sap2.35.192.41.172.nlp.io/ms2/clases/`. The method is **GET**. The response status is **200 OK** with a response time of 115 ms and a body size of 844 B. The response body is a JSON array of class objects:

```
1 [
2   {
3     "capacidad": 20,
4     "descripcion": "Introducción al idioma alemán",
5     "id_clase": 1,
6     "id_horario": 1,
7     "nombre_clase": "Clase Alemán Básico"
8   },
9   {
10    "capacidad": 10,
11    "descripcion": "Basico al idioma alemán",
12    "id_clase": 2,
13    "id_horario": 2,
14    "nombre_clase": "Clase Alemán Intermedio"
15  },
16  {
17    "capacidad": 30,
18    "descripcion": "Basico al idioma alemán",
19    "id_clase": 3,
20    "id_horario": 3,
21    "nombre_clase": "Clase Alemán Intermedio B2"
22  },
23  {
24    "capacidad": 40,
25    "descripcion": "Avanzado al idioma alemán",
26    "id_clase": 4,
27    "id_horario": 4,
28    "nombre_clase": "Clase Alemán Intermedio B1"
29  }
30 ]
```

## Ms2 Clases: Reservar clases

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nip.io/ms2/clases/reservas`
- Method:** `POST`
- Body (raw):**

```
1 {  
2   "id_usuario": 4,  
3   "id_clase": 3  
4 }  
5
```
- Response:** `200 OK` (94 ms, 183 B). The response body (JSON) is:

```
1 {  
2   "mensaje": "Reserva registrada exitosamente"  
3 }
```

## Ms3 Pagos: Registrar pago

The screenshot shows a REST client interface with the following details:

- URL:** `http://sap2.35.192.41.172.nip.io/ms3/pagos/`
- Method:** `POST`
- Body (raw):**

```
1 {  
2   "id_usuario": 4,  
3   "monto": 99.99,  
4   "metodo_pago": "Tarjeta"  
5 }
```
- Response:** `201 CREATED` (97 ms, 185 B). The response body (JSON) is:

```
1 {  
2   "mensaje": "Pago registrado exitosamente"  
3 }
```

## Ms3 Pagos: Ver pagos

REST client interface showing a GET request to `http://sap2.35.192.41.172.nip.io/ms3/pagos/`. The response is a 200 OK status with a JSON body.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body**

```
1 {
2   {
3     "fecha_pago": "Tarjeta",
4     "id_pago": 3,
5     "id_usuario": 4,
6     "metodo_pago": "Sun, 15 Dec 2024 03:10:52 GMT",
7     "monto": "60.00"
8   }
9 }
```

200 OK • 84 ms • 286 B

## Ms4 Horarios: Añadir horario

REST client interface showing a POST request to `http://sap2.35.192.41.172.nip.io/ms4/horarios/`. The request body is a JSON object, and the response is a 200 OK status with a success message.

**Body**

```
1 {
2   {
3     "dia": "Lunes",
4     "hora_inicio": "13:00:00",
5     "hora_fin": "15:00:00"
6   }
7 }
```

200 OK • 97 ms • 183 B

**Body**

```
1 {
2   {
3     "mensaje": "Horario registrado exitosamente"
4   }
5 }
```

## Ms4 Horarios: Ver horarios

http://sap2.35.192.41.172.nip.io/ms4/horarios/

GET http://sap2.35.192.41.172.nip.io/ms4/horarios/ Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (4) Test Results

200 OK · 83 ms · 889 B

Pretty Raw Preview Visualize JSON

```
1  [
2    {
3      "dia": "Lunes",
4      "hora_fin": "9:00:00",
5      "hora_inicio": "8:00:00",
6      "id_horario": 1
7    },
8    {
9      "dia": "Martes",
10     "hora_fin": "9:00:00",
11     "hora_inicio": "8:00:00",
12     "id_horario": 2
13   },
14   {
15     "dia": "Miercoles",
16     "hora_fin": "9:00:00",
17     "hora_inicio": "8:00:00",
18     "id_horario": 3
19   },
20   {
21     "dia": "Jueves",
22     "hora_fin": "9:00:00",
23     "hora_inicio": "8:00:00",
24     "id_horario": 4
25   },
26   {
27     "dia": "Viernes",
28     "hora_fin": "9:00:00",
29     "hora_inicio": "8:00:00",
30     "id_horario": 5
31   },
32 ]
```