**1. Registro de Usuarios**

**Endpoint:** POST /api/usuarios/registro  
**URL Completa:** http://localhost:8080/api/usuarios/registro

**Método:** POST  
**Body (JSON):**

json

Copiar código

{

"nombre": "Juan Perez",

"email": "juan.perez@gmail.com",

"documento": "12345678",

"password": "admin1234"

}

**Casos de Prueba:**

1. Usuario nuevo: **201 Created**.
2. Usuario duplicado (mismo email): **400 Bad Request**.
3. Campos faltantes:
   * Sin email: **400 Bad Request**, "El email es obligatorio."
   * Sin password: **400 Bad Request**, "La contraseña es obligatoria."

**2. Listar Servicios**

**Endpoint:** GET /api/servicios  
**URL Completa:** http://localhost:8080/api/servicios

**Método:** GET  
**Body:** **Ninguno**

**Casos de Prueba:**

1. Lista de servicios: **200 OK**, devuelve los servicios registrados.
2. Sin servicios registrados: **200 OK**, devuelve [].

**3. Pago de Servicio**

**Endpoint:** POST /api/pagos  
**URL Completa:** http://localhost:8080/api/pagos

**Método:** POST  
**Parámetros (Query Params):**

makefile

Copiar código

usuarioId=1

codigoServicio=ANDE001

monto=50000

referencia=REF001

**Casos de Prueba:**

1. **Pago exitoso:**
   * Saldo del usuario suficiente y servicio existe: **201 Created**.
2. **Errores:**
   * Servicio no encontrado: **400 Bad Request**, "Servicio no encontrado."
   * Usuario no encontrado: **400 Bad Request**, "Usuario no encontrado."
   * Saldo insuficiente: **400 Bad Request**, "Saldo insuficiente."
   * Monto mayor que la deuda del servicio: **400 Bad Request**, "El monto excede la deuda del servicio."
   * Campos faltantes:
     + Sin usuarioId, codigoServicio, o monto: **400 Bad Request**.
     + Monto negativo: **400 Bad Request**, "El monto debe ser mayor que cero."

**4. Consultar Pagos por Rango de Fechas**

**Endpoint:** GET /api/pagos/rango-fechas  
**URL Completa:** http://localhost:8080/api/pagos/rango-fechas

**Método:** GET  
**Parámetros (Query Params):**

makefile

Copiar código

fechaInicio=2024-01-01T00:00:00

fechaFin=2024-12-31T23:59:59

**Casos de Prueba:**

1. **Transacciones en el rango:**
   * **200 OK**, devuelve lista de transacciones.
2. **Sin transacciones en el rango:**
   * **200 OK**, devuelve [].
3. **Errores:**
   * Fechas en formato incorrecto: **400 Bad Request**, "Error al procesar las fechas."

**5. Consultar Pagos por Tipo de Servicio**

**Endpoint:** GET /api/pagos/tipo-servicio  
**URL Completa:** http://localhost:8080/api/pagos/tipo-servicio

**Método:** GET  
**Parámetros (Query Params):**

makefile

Copiar código

codigoServicio=ANDE001

**Casos de Prueba:**

1. **Transacciones encontradas:**
   * **200 OK**, devuelve lista de transacciones para el servicio.
2. **Sin transacciones:**
   * **404 Not Found**, "Datos no encontrados."
3. **Errores:**
   * Código de servicio inválido: **404 Not Found**, "Datos no encontrados."

**6. Autenticación y Seguridad**

1. **Sin autenticación:**
   * Prueba cualquier endpoint sin credenciales: **403 Forbidden**.
2. **Credenciales inválidas:**
   * Intenta con usuario/contraseña incorrectos: **401 Unauthorized**.

**Notas Adicionales:**

* **Formato de Fechas:** Usa el formato yyyy-MM-dd'T'HH:mm:ss para los parámetros fechaInicio y fechaFin.
* **Encabezados Requeridos:**
  + Para endpoints con autenticación, usa Basic Auth en Postman con:
    - **Username:** user
    - **Password:** Contraseña generada por Spring Boot.

**Lista Resumida de Endpoints**

| **Método** | **URL** | **Descripción** |
| --- | --- | --- |
| POST | /api/usuarios/registro | Registrar un usuario. |
| GET | /api/servicios | Listar servicios registrados. |
| POST | /api/pagos | Realizar el pago de un servicio. |
| GET | /api/pagos/rango-fechas | Consultar pagos por rango de fechas. |
| GET | /api/pagos/tipo-servicio | Consultar pagos por tipo de servicio. |