



Universidad de las Américas
Facultad de Ingeniería y Ciencias Aplicadas

Documentación Examen Progreso 2

Realizado por
David Andrés Guillén Balarezo.

Pruebas Documentadas del Sistema de Gestión Hotelera

Autor: David Guillén

Versión: 1.0.0

Introducción

Este documento detalla las pruebas realizadas para los endpoints principales de los microservicios que componen el Sistema de Gestión Hotelera. Cada prueba se llevó a cabo utilizando la interfaz Swagger disponible en `/api-docs`, además de herramientas como Postman y SoapUI para escenarios específicos. Se incluyen ejemplos de solicitudes y respuestas, con capturas de pantalla sugeridas para respaldar los resultados.

1. Microservicio de Gestión de Inventario

Endpoint: Crear una Habitación

- **Método:** POST
- **URL:** `/rooms`

Solicitud

```
{
  "room_number": 104,
  "room_type": "single"
}
```

Respuesta Esperada

- **Código:** 201 Created

```
{
  "room_id": 4,
  "room_number": 104,
  "room_type": "single",
  "status": "available"
}
```

Rooms

POST /rooms Registrar una nueva habitación.

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{  "room_number": 101,  "room_type": "single"}  
```

Execute

Clear

Curl

```
curl -X 'POST' \  'http://localhost:5000/rooms' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "room_number": 101,  "room_type": "single"}'
```

Request URL

http://localhost:5000/rooms

Server response

Code

Details

201

Response body

```
{  "room_id": 4,  "room_number": 101,  "room_type": "single",  "status": "available"}  
```

Download

Response headers

```
connection: keep-alive  content-length: 73  content-type: application/json; charset=utf-8  date: Sun, 15 Dec 2024 02:34:13 GMT  etag: W/"49--zcp13bC6adtmyI00e96L1XGSU"  keep-alive: timeout=5  x-powered-by: Express
```

Endpoint: Actualizar el Estado de una Habitación

- **Método:** PATCH
- **URL:** /rooms/{id}

Solicitud

```
{  "status": "maintenance"}  
```

Respuesta Esperada

- **Código:** 200 OK

```
{  "room_id": 4,  "room_number": 104,  "room_type": "single",  "status": "maintenance"}  
```

}

PATCH

/rooms/{id}

Actualizar el estado de una habitación.

Parameters

Cancel

Reset

Name	Description
id <small>* required</small>	ID de la habitación a actualizar.
integer (path)	<input type="text" value="4"/>

Request body required

application/json

```
{
  "status": "maintenance"
}
```

Execute

Clear

Antes:

200

Response body

```
[
  {
    "room_id": 1,
    "room_number": 101,
    "room_type": "single",
    "status": "available"
  },
  {
    "room_id": 2,
    "room_number": 102,
    "room_type": "double",
    "status": "available"
  },
  {
    "room_id": 3,
    "room_number": 103,
    "room_type": "single",
    "status": "maintenance"
  },
  {
    "room_id": 4,
    "room_number": 101,
    "room_type": "single",
    "status": "available"
  }
]
```

Download

Response headers

```
connection: keep-alive
content-length: 299
content-type: application/json; charset=utf-8
date: Sun, 15 Dec 2024 02:34:27 GMT
etag: W/"12b-mk2j+VRPOC3pn7XsmAcTiihv7kI"
keep-alive: timeout=5
x-powered-by: Express
```

Despues:

200

Response body

```
[
  {
    "room_id": 1,
    "room_number": 101,
    "room_type": "single",
    "status": "available"
  },
  {
    "room_id": 2,
    "room_number": 102,
    "room_type": "double",
    "status": "available"
  },
  {
    "room_id": 3,
    "room_number": 103,
    "room_type": "single",
    "status": "maintenance"
  },
  {
    "room_id": 4,
    "room_number": 101,
    "room_type": "single",
    "status": "maintenance"
  }
]
```

Download

Response headers

```
connection: keep-alive
content-length: 301
content-type: application/json; charset=utf-8
date: Sun, 15 Dec 2024 02:42:40 GMT
etag: W/"12d-Gor9wllblgK2K1MLrc8KSLQsha"
keep-alive: timeout=5
x-powered-by: Express
```

Endpoint: Listar Todas las Habitaciones

- **Método:** GET
- **URL:** /rooms

Respuesta Esperada

- **Código:** 200 OK

```
[  
  {  
    "room_id": 1,  
    "room_number": 101,  
    "room_type": "single",  
    "status": "available"  
  },  
  {  
    "room_id": 2,  
    "room_number": 102,  
    "room_type": "double",  
    "status": "maintenance"  
  }  
]
```



2. Microservicio de Gestión de Reservas

Endpoint: Crear una Reserva

- **Método:** POST
- **URL:** /reservations

Solicitud

```
{
  "room_id": 1,
  "customer_name": "Alice Doe",
  "start_date": "2024-12-25",
  "end_date": "2024-12-30"
}
```

Respuesta Esperada

- **Código:** 201 Created

json

Copiar código

```
{
  "reservation_id": 3,
  "room_id": 1,
  "customer_name": "Alice Doe",
  "start_date": "2024-12-25",
  "end_date": "2024-12-30",
  "status": "active"
}
```

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

- Method and URL:** POST /reservations. A subtitle reads "Crea una nueva reserva."
- Parameters:** A section labeled "Parameters" with "No parameters" listed below it. It includes "Cancel" and "Reset" buttons.
- Request body:** A section labeled "Request body" with a red "required" indicator. A dropdown menu shows "application/json".
- Request body content:** A text area containing the JSON:

```
{
  "start_date": "2024-12-20",
  "end_date": "2024-12-22",
  "room_type": "single",
  "customer_name": "Alice Cooper"
}
```
- Buttons:** At the bottom, there are "Execute" and "Clear" buttons.

Curl

```
curl -X 'POST' \
  'http://localhost:4000/reservations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "start_date": "2024-12-20",
    "end_date": "2024-12-22",
    "room_type": "single",
    "customer_name": "Alice Cooper"
  }'
```

Request URL

http://localhost:4000/reservations

Server response

Code	Details
201	<p>Response body</p> <pre>{ "reservation_id": 6, "room_id": 1, "customer_name": "Alice Cooper", "start_date": "2024-12-20T00:00:00.000Z", "end_date": "2024-12-22T00:00:00.000Z", "status": "confirmed" }</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 162 content-type: application/json; charset=utf-8 date: Sun, 15 Dec 2024 02:32:15 GMT etag: W/"a2-/TR/P68Ta+Z52grUWQ+6gkmb6c" keep-alive: timeout=5 x-powered-by: Express</pre>

Endpoint: Consultar una Reserva

- **Método:** GET
- **URL:** /reservations/{id}

Respuesta Esperada

- **Código:** 200 OK

```
{
  "reservation_id": 3,
  "room_id": 1,
  "customer_name": "Alice Doe",
  "start_date": "2024-12-25",
  "end_date": "2024-12-30",
  "status": "active"
}
```

GET /reservations/{id} Obtiene una reserva específica por ID.

Parameters

Cancel

Name	Description
id required	
integer	
(path)	
3	

Execute Clear

Curl

```
curl -X 'GET' \
  'http://localhost:4000/reservations/3' \
  -H 'accept: */*'
```

Request URL

http://localhost:4000/reservations/3

Server response

Code	Details
200	<p>Response body</p> <pre>{ "reservation_id": 3, "room_id": 2, "customer_name": "Jane Smith", "start_date": "2024-12-23T00:00:00.000Z", "end_date": "2024-12-25T00:00:00.000Z", "status": "confirmed" }</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 160 content-type: application/json; charset=utf-8 date: Sun, 15 Dec 2024 02:44:32 GMT etag: W/"a0-ifSStkdo2B5qczRQyzh7owlu/g" keep-alive: timeout=5 x-powered-by: Express</pre>

Endpoint: Cancelar una Reserva

- **Método:** DELETE
- **URL:** /reservations/{id}

Respuesta Esperada

- **Código:** 200 OK

```
{
  "message": "Reserva eliminada exitosamente."
}
```

DELETE /reservations/{id} Elimina una reserva específica por ID.

Parameters Cancel

Name	Description
id required	
integer	
(path)	

Execute Clear

Curl

```
curl -X 'DELETE' \
  'http://localhost:4000/reservations/3' \
  -H 'accept: */*'
```

Request URL

http://localhost:4000/reservations/3

Server response

Code	Details
200	<p>Response body</p> <pre>{ "message": "Reserva eliminada exitosamente." }</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 45 content-type: application/json; charset=utf-8 date: Sun, 15 Dec 2024 02:45:39 GMT etag: W/"2d-lUThEXfuWaeEQ5IHVSRpEAZE" keep-alive: timeout=5 x-powered-by: Express</pre>

Name	Description
id * required integer (path)	3

Execute
Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:4000/reservations/3' \
  -H 'accept: */*'

```

Request URL

```
http://localhost:4000/reservations/3

```

Server response

Code	Details
404	<p>Error: Not Found</p> <p>Response body</p> <pre>{ "error": "Reserva no encontrada." }</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 34 content-type: application/json; charset=utf-8 date: Sun, 15 Dec 2024 02:45:53 GMT etag: W/"22-pe9YZfnyTKmcCIAXHP9Fwa8zty" keep-alive: timeout=5 x-powered-by: Express</pre>

3. Microservicio SOAP (Verificación de Disponibilidad)

Endpoint: CheckAvailability

- **Método:** SOAP
- **WSDL URL:** /soap?wsdl

Solicitud (SOAP)

```
xml
Copiar código
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hot="http://example.org/HotelAvailability">
  <soapenv:Header/>
  <soapenv:Body>
    <hot:CheckAvailabilityRequest>
      <startDate>2024-12-25</startDate>
      <endDate>2024-12-30</endDate>
      <roomType>single</roomType>
    </hot:CheckAvailabilityRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Respuesta Esperada

```
xml
Copiar código
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Body>
    <hot:CheckAvailabilityResponse>
      <rooms>
```

```
<room>
  <room_id>1</room_id>
  <available_date>2024-12-25</available_date>
</room>
</rooms>
</hot:CheckAvailabilityResponse>
</soapenv:Body>
</soapenv:Envelope>
```

POST

/soap

Enviar solicitud SOAP al servicio de disponibilidad.

⌵

Parameters

Cancel

Reset

No parameters

Request body required

text/xml

⌵

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:hot="http://www.example.org/HotelAvailability/">
  <soapenv:Header/>
  <soapenv:Body>
    <hot:CheckAvailabilityRequest>
      <startDate>2024-12-26</startDate>
      <endDate>2024-12-28</endDate>
      <roomType>single</roomType>
    </hot:CheckAvailabilityRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Execute

Clear

Curl

```
curl -X 'POST' \
  'http://localhost:3000/soap' \
  -H 'accept: */*' \
  -H 'Content-Type: text/xml' \
  -d '{
    "soapenv:Envelope": {
      "xmlns:soapenv": "http://schemas.xmlsoap.org/soap/envelope/",
      "xmlns:hot": "http://www.example.org/HotelAvailability/",
      "Body": {
        "hot:CheckAvailabilityRequest": {
          "startDate": "2024-12-26",
          "endDate": "2024-12-28",
          "roomType": "single"
        }
      }
    }
  }'
```

Request URL

http://localhost:3000/soap

Server response

Code

Details

200

Response body

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="http://www.example.org/HotelAvailability/">
  <soap:Body>
    <tns:CheckAvailabilityResponse>
      <rooms>
        <room>
          <room_id>5</room_id>
          <available_date>2024-12-26</available_date>
        </room>
        <room>
          <room_id>6</room_id>
          <available_date>2024-12-27</available_date>
        </room>
        <room>
          <room_id>7</room_id>
          <available_date>2024-12-28</available_date>
        </room>
      </rooms>
    </tns:CheckAvailabilityResponse>
  </soap:Body>
</soap:Envelope>
```

Download

Response headers

```
connection: keep-alive
content-type: text/xml
date: Sun, 15 Dec 2024 01:51:50 GMT
keep-alive: timeout=5
transfer-encoding: chunked
x-powered-by: Express
```

URLs Principales

- **Inventario:** <http://localhost:5000/api-docs>
 - **Reservas:** <http://localhost:4000/api-docs>
 - **SOAP:** <http://localhost:3000/api-docs>
-

Notas Finales

Para garantizar la calidad de las pruebas:

- Utiliza Swagger para probar la mayoría de los endpoints HTTP.
- Usa Postman para pruebas más avanzadas o escenarios específicos.
- Para SOAP, utiliza SoapUI y verifica la estructura XML de la solicitud y respuesta.