

Documentación de Despliegue – CoWork Manager

Joaquín López

Contenido

Documentación de Despliegue – CoWork Manager	1
1. Introducción	3
2. Información General de la API	3
3. Ejecución del Servicio	3
4. Módulo A – Gestión de Salas	4
5. Módulo B – Gestión de Reservas	8
6. conclusión	13

1. Introducción

El presente documento describe el funcionamiento y la forma de consumo de la API **CoWork Manager**, desarrollada como parte de la Evaluación Parcial 2 del módulo *Integración de Plataformas*.

La API permite gestionar **salas de reuniones** y **reservas**, siguiendo los principios REST, utilizando formato JSON y ejecutándose en un entorno local (localhost).

Este documento sirve como **contrato de consumo**, es decir, con esta información es posible probar y validar completamente la API sin necesidad de revisar el código fuente.

2. Información General de la API

- **Tipo de API:** RESTful
- **Formato de intercambio:** JSON
- **Servidor:** Localhost
- **Puerto:** 3000
- **Base URL:**

<http://localhost:3000/api/v1>

<http://localhost:3000/docs#/>

3. Ejecución del Servicio

Para ejecutar la API en entorno local se deben instalar las dependencias necesarias y levantar el servidor con Uvicorn.

- `python -m pip install fastapi uvicorn`
- `python -m uvicorn main:app --reload --port 3000`

Una vez iniciado el servidor, la API queda disponible y se puede probar mediante el navegador o utilizando Swagger UI en:

- <http://localhost:3000/docs>

4. Módulo A – Gestión de Salas

Endpoint 1: Listar Salas

Este endpoint permite obtener el listado completo de las salas registradas en el sistema.

- **URL:**
http://localhost:3000/api/v1/salas
- **Método:** GET
- **Payload:** No aplica

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

- Method:** GET
- URL:** /api/v1/salas
- Parameters:** No parameters
- Execute:** Button to execute the request
- Clear:** Button to clear the request
- Responses:** Section showing the response details
- Curl:** curl -X 'GET' \ 'http://localhost:3000/api/v1/salas' \ -H 'accept: application/json'
- Request URL:** http://localhost:3000/api/v1/salas
- Server response:** 200
- Response body:** []
- Response headers:** content-length: 2, content-type: application/json, date: Thu, 22 Jan 2026 05:22:45 GMT, server: unicorn

Endpoint 2: Obtener Detalle de Sala

Permite consultar la información detallada de una sala específica utilizando su identificador.

- **URL:**
`http://localhost:3000/api/v1/salas/{sala_id}`
- **Método:** GET
- **Payload:** No aplica

GET `/api/v1/salas/{sala_id}` Obtener Sala ^

Parameters Cancel

Name	Description
sala_id * required	
integer	
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:3000/api/v1/salas/1' \
  -H 'accept: application/json'
```

Request URL

`http://localhost:3000/api/v1/salas/1`

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 1, "nombre": "Sala A", "capacidad": 10, "ubicacion": "Norte" }</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 61 content-type: application/json date: Thu, 22 Jan 2026 05:24:40 GMT server: uvicorn</pre></div></div>

Responses

Endpoint 3: Crear Sala

Este endpoint permite registrar una nueva sala dentro del sistema.

El nombre de la sala no puede repetirse.

- **URL:**
`http://localhost:3000/api/v1/salas`
- **Método:** POST

POST /api/v1/salas Crear Sala

Parameters

No parameters

Request body required

application/json

Edit Value | Schema

```
{  "nombre": "Sala A",  "capacidad": 10,  "ubicacion": "Norte"}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \  'http://localhost:3000/api/v1/salas' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "nombre": "Sala A",  "capacidad": 10,  "ubicacion": "Norte"  }'
```

Request URL

`http://localhost:3000/api/v1/salas`

Code Details

400 undocumented

Error: Bad Request

Response body

```
{  "detail": {    "error": "Nombre duplicado"  }  }
```

Download

Response headers

```
content-length: 39  content-type: application/json  date: Thu,22 Jan 2026 05:26:09 GMT  server: uvicorn
```

Responses

Endpoint 4: Eliminar Sala

Permite eliminar una sala del sistema utilizando su ID.

- **URL:**
`http://localhost:3000/api/v1/salas/{sala_id}`
- **Método:** DELETE
- **Payload:** No aplica

DELETE `/api/v1/salas/{sala_id}` Eliminar Sala ^

Parameters Cancel

Name	Description
sala_id * required integer (path)	<input type="text" value="2"/>

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:3000/api/v1/salas/2' \
-H 'accept: */*'
```

Request URL

```
http://localhost:3000/api/v1/salas/2
```

Server response

Code	Details
204	<div>Response headers<div>content-type: application/json date: Thu, 22 Jan 2026 05:27:10 GMT server: uvicorn</div></div>

Responses

Code	Description	Links
204	Successful Response	No links

5. Módulo B – Gestión de Reservas

Endpoint 5: Crear Reserva

Este endpoint permite registrar una reserva asociada a una sala existente.

Las fechas deben venir en formato ISO-8601 y la sala debe existir previamente.

- **URL:**
`http://localhost:3000/api/v1/reservas`
- **Método:** POST

Request body required

application/json

Edit Value | Schema

```
{
  "sala_id": 3,
  "nombre_solicitante": "Joaquin Lopez",
  "fecha_inicio": "2026-01-20",
  "fecha_fin": "2026-01-22"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/api/v1/reservas' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "sala_id": 3,
    "nombre_solicitante": "Joaquin Lopez",
    "fecha_inicio": "2026-01-20",
    "fecha_fin": "2026-01-22"
  }'
```

Request URL

http://localhost:3000/api/v1/reservas

Server response

Code Details

201

Response body

```
{
  "id": 1,
  "mensaje": "Creado"
}
```

Curl

```
curl -X 'POST' \
  'http://localhost:3000/api/v1/reservas' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "sala_id": 1,
    "nombre_solicitante": "Joaquin Lopez",
    "fecha_inicio": "2026-01-20",
    "fecha_fin": "2026-01-22"
  }'
```

Request URL

http://localhost:3000/api/v1/reservas

Server response

Code Details

400

undocumented

Error: Bad Request

Response body

```
{
  "detail": {
    "error": "sala_id no existe"
  }
}
```

Response headers

```
content-length: 40
content-type: application/json
date: Thu, 22 Jan 2026 05:29:55 GMT
server: uvicorn
```

Responses

Endpoint 6: Historial de Reservas

Permite obtener el historial de reservas registradas en el sistema.

Adicionalmente, se puede filtrar por sala mediante un parámetro de consulta.

- **URL:**
`http://localhost:3000/api/v1/reservas`
- **Método:** GET

GET

/api/v1/reservas

Historial Reservas

^

Parameters

Cancel

Name	Description
sala_id	<input type="text" value="3"/>
integer (integer null) (query)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:3000/api/v1/reservas?sala_id=3' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:3000/api/v1/reservas?sala_id=3
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>[{ "id": 1, "sala_id": 3, "nombre_solicitante": "Joaquin Lopez", "fecha_inicio": "2026-01-20", "fecha_fin": "2026-01-22" }]</pre></div><div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-length: 112 content-type: application/json date: Thu, 22 Jan 2026 05:30:41 GMT server: uvicorn</pre></div></div>

Responses

Endpoint 7: Modificar Reserva

Permite modificar parcialmente una reserva existente, específicamente la hora de término (fecha_fin).

- **URL:**
`http://localhost:3000/api/v1/reservas/{reserva_id}`
- **Método:** PATCH

PATCH /api/v1/reservas/{reserva_id} Modificar Reserva

Parameters

Cancel

Reset

Name	Description
reserva_id * required	
integer (path)	<input type="text" value="1"/>

Request body required

application/json

Edit Value | Schema

```
{  "fecha_fin": "2026-01-23"}
```

Execute

Clear

Responses

Curl

```
curl -X 'PATCH' \
  'http://localhost:3000/api/v1/reservas/1' \
  -H 'accept: application/json' \
  -H 'Content-type: application/json' \
  -d '{
    "fecha_fin": "2026-01-23"
  }'
```

Code

Details

200

Response body

```
{  "id": 1,  "mensaje": "Actualizado",  "fecha_fin": "2026-01-23"}
```

Download

Response headers

```
content-length: 57
content-type: application/json
date: Thu, 22 Jan 2026 05:31:53 GMT
server: uvicorn
```

Endpoint 8: Cancelar Reserva

Permite cancelar (eliminar) una reserva existente utilizando su identificador.

- **URL:**
`http://localhost:3000/api/v1/reservas/{reserva_id}`
- **Método:** DELETE
- **Payload:** No aplica

DELETE `/api/v1/reservas/{reserva_id}` Cancelar Reserva

Parameters Cancel

Name	Description
reserva_id * required integer (path)	<input type="text" value="2"/>

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \  
  'http://localhost:3000/api/v1/reservas/2' \  
  -H 'accept: */*'
```

Request URL

```
http://localhost:3000/api/v1/reservas/2
```

Server response

Code	Details
204	<div>Response headers<pre>content-type: application/json date: Thu, 22 Jan 2026 05:32:33 GMT server: uvicorn</pre></div>

Responses

Code	Description	Links
204	Successful Response	No links

6. conclusión

La API cumple con los principios REST, utilizando correctamente los verbos HTTP y los códigos de estado según la operación realizada.

Todas las entradas y salidas utilizan formato JSON y se implementan validaciones básicas para asegurar la consistencia de los datos.

Este documento permite probar completamente la API sin ambigüedades, cumpliendo con los requerimientos funcionales y no funcionales definidos en la evaluación.