

# API Documentation

- [ProjectPlanning Cloud Persistence API - Documentación Completa](#)
  - [Tabla de Contenidos](#)
  - [Información del Grupo](#)
    - [Integrantes](#)
    - [Universidad y Cátedra](#)
  - [URLs de Servicios](#)
    - [API Cloud Persistence \(Producción\)](#)
    - [Documentación Interactiva](#)
  - [Introducción](#)
    - [¿Qué es esta API?](#)
    - [Características Principales](#)
    - [Stack Tecnológico](#)
  - [Autenticación](#)
    - [Flujo de Autenticación JWT](#)
    - [Headers Requeridos](#)
    - [Información de Tokens](#)
  - [Endpoints de Autenticación](#)
    - [1|Registrar Nuevo Usuario](#)
    - [2|Login de Usuario](#)
    - [3|Refrescar Token](#)
  - [Endpoints de Proyectos](#)
    - [1|Crear Proyecto](#)
    - [2|Obtener Proyecto por ID](#)
    - [3|Actualizar Proyecto \(Parcial\)](#)
    - [4|Eliminar Proyecto](#)
  - [Endpoints de Pedidos](#)
    - [1|Crear Pedido para Etapa Existente](#)
    - [2|Listar Pedidos de Proyecto](#)
    - [3|Eliminar Pedido](#)
  - [Endpoints de Ofertas](#)
    - [1|Crear Oferta para Pedido](#)
    - [2|Listar Ofertas de Pedido](#)
    - [3|Aceptar Oferta](#)
    - [4|Rechazar Oferta](#)
    - [5|Confirmar Realización de Oferta](#)
    - [6|Obtener Mis Compromisos](#)
  - [Enumeraciones](#)
    - [EstadoProyecto](#)
    - [TipoPedido](#)
    - [EstadoPedido](#)
    - [EstadoOferta](#)
    - [UserRole](#)
  - [Flujo Completo de Ejemplo](#)
    - [Paso 1: Registrar Dos Usuarios](#)
    - [Paso 2: Login con Ambos Usuarios](#)
    - [Paso 3: María Crea un Proyecto](#)
    - [Paso 4: Carlos Crea una Oferta](#)
    - [Paso 5: María ve las Ofertas del Pedido](#)
    - [Paso 6: María Acepta la Oferta de Carlos](#)
    - [Paso 7: Carlos Confirma que Realizó el Trabajo](#)
    - [Paso 8: Carlos Visualiza sus Compromisos](#)
  - [Códigos de Error](#)
    - [Códigos HTTP Estándar](#)
    - [Errores por Endpoint](#)
    - [Ejemplos de Respuestas de Error](#)
    - [Cómo Interpretar Errores 422 \(Validación\)](#)
  - [Instrucciones para Pruebas](#)
    - [Quick Start - La Forma Más Rápida](#)
    - [Opción 1: Swagger UI \(Recomendado\)](#)
    - [Opción 2: Postman](#)
    - [Opción 3: cURL \(Terminal\)](#)
  - [Checklist de Pruebas](#)
    - [Autenticación](#)
    - [Proyectos](#)
    - [Pedidos](#)
    - [Ofertas](#)
  - [Soporte](#)

## ProjectPlanning Cloud Persistence API - Documentación Completa

Versión: 1.0.0 Fecha de Entrega: Octubre 2025

---

### Tabla de Contenidos

1. [Información del Grupo](#)
2. [URLs de Servicios](#)
3. [Introducción](#)
4. [Autenticación](#)
5. [Endpoints de Autenticación](#)

- 
- 6. [Endpoints de Proyectos](#)
  - 7. [Endpoints de Pedidos](#)
  - 8. [Endpoints de Ofertas](#)
  - 9. [Enumeraciones](#)
  - 10. [Flujo Completo de Ejemplo](#)
  - 11. [Códigos de Error](#)
  - 12. [Instrucciones para Pruebas](#)
- 

## Información del Grupo

### Integrantes

Nombre	Rol	Email
Mateo Spinetti	[Desarrollador]	<a href="mailto:mateospinetti1@gmail.com">mateospinetti1@gmail.com</a>
Luciano Equiel Wagner	[Desarrollador]	<a href="mailto:lucianowagner2003@gmail.com">lucianowagner2003@gmail.com</a>
Matias Santiago Ramos Giacosa	[Desarrollador]	<a href="mailto:matiasramosgi@gmail.com">matiasramosgi@gmail.com</a>

### Universidad y Cátedra

- **Universidad:** Facultad de informatica UNLP
  - **Materia:** Desarrollo de software en sistemas distribuidos
  - **Período:** 2025
- 

## URLs de Servicios

### API Cloud Persistence (Producción)

**URL Base:** <https://project-planning-cloud-api.onrender.com>

#### Acceso Directo:

- **API v1:** <https://project-planning-cloud-api.onrender.com/api/v1>
- **Swagger/Documentación Interactiva:** <https://project-planning-cloud-api.onrender.com/docs>
- **OpenAPI JSON:** <https://project-planning-cloud-api.onrender.com/openapi.json>
- **Health Check:** <https://project-planning-cloud-api.onrender.com/health>

### Documentación Interactiva

Para explorar y probar todos los endpoints **sin necesidad de instalar nada**:

[Abre Swagger UI aquí](#)

En Swagger UI podrás:

- Ver todos los endpoints disponibles
  - Probar cada endpoint directamente
  - Obtener automáticamente ejemplos de request/response
  - Ver códigos de error y documentación detallada
- 

## Introducción

### ¿Qué es esta API?

La **ProjectPlanning Cloud Persistence API** es un servicio backend REST construido con **FastAPI** que gestiona toda la persistencia de datos del sistema ProjectPlanning. Esta API:

- Maneja autenticación de usuarios con JWT
- Gestiona proyectos con etapas anidadas
- Administra pedidos (solicitudes de recursos)
- Procesa ofertas de usuarios

### Características Principales

**Validación Pydantic** - Validación completa de entrada/salida **JWT Authentication** - Tokens de acceso y refresco **CORS Configurado** - Compatible con proxy API **Docker** - Despliegue containerizado **PostgreSQL 15+** - Base de datos relacional

### Stack Tecnológico

- **Framework:** FastAPI
  - **Python:** 3.12+
  - **ORM:** SQLAlchemy 2.0 (async)
  - **Base de Datos:** PostgreSQL 15+
  - **Autenticación:** JWT con bcrypt
  - **Validación:** Pydantic v2
  - **Server:** Uvicorn
  - **Deployment:** Docker + docker-compose
- 

## Autenticación

### Flujo de Autenticación JWT

Esta API usa **JWT (JSON Web Tokens)** para autenticación:

#### Paso 1: Registro

`POST /api/v1/auth/register`  
→ Recibe: email, password, nombre, apellido, ong, role

→ Devuelve: usuario creado

#### Paso 2: Login

POST /api/v1/auth/login  
→ Recibe: email, password  
→ Devuelve: access\_token (15 min) + refresh\_token (24h)

#### Paso 3: Usar Token

Incluye el access\_token en todas las peticiones protegidas:

Authorization: Bearer {access\_token}

#### Paso 4: Refrescar Token (cuando expira)

POST /api/v1/auth/refresh  
→ Recibe: refresh\_token  
→ Devuelve: nuevo access\_token + refresh\_token

### Headers Requeridos

Para endpoints protegidos, incluye:

Authorization: Bearer eyJ0exAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...  
Content-Type: application/json

### Información de Tokens

Propiedad	Valor
Algoritmo	HS256
Access Token Expiry	15 minutos
Refresh Token Expiry	24 horas (1440 min)
Tipo de Token	Bearer

## Endpoints de Autenticación

### 1 Registrar Nuevo Usuario

Crea una nueva cuenta de usuario en el sistema.

**Método:** POST **Ruta:** /api/v1/auth/register **Autenticación:** No requerida (público) **Código de Respuesta:** 201 Created

#### Parámetros

Campo	Tipo	Requerido	Descripción	Restricciones
email	String	Sí	Email del usuario	Debe ser único, formato válido
password	String	Sí	Contraseña	Mínimo 8 caracteres
nombre	String	Sí	Nombre del usuario	-
apellido	String	Sí	Apellido del usuario	-
ong	String	Sí	Nombre de organización	-
role	enum	No	Rol del usuario	MEMBER (default) o COUNCIL

#### Body de Prueba

```
{  
    "email": "juan.perez@ong.com",  
    "password": "SecurePassword123",  
    "nombre": "Juan",  
    "apellido": "Pérez",  
    "ong": "ONG Solidaria",  
    "role": "MEMBER"  
}
```

#### Response Exitoso (201)

```
{  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "email": "juan.perez@ong.com",  
    "nombre": "Juan",  
    "apellido": "Pérez",  
    "ong": "ONG Solidaria",  
    "role": "MEMBER",  
    "created_at": "2024-10-22T14:30:00+00:00",  
    "updated_at": "2024-10-22T14:30:00+00:00"  
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
400	Email ya registrado	{"detail": "A user with this email already exists"}	Usa un email diferente
400	Contraseña demasiado larga	{"detail": "Password cannot exceed 72 bytes in length"}	Usa una contraseña más corta
400	Datos de registro inválidos	{"detail": "Invalid registration data"}	Revisa que todos los campos cumplan los requisitos
422	Error de validación Pydantic	{"detail": [{"loc": ["body", "email"], "msg": "invalid email format", "type": "value_error.email"}]}	Revisa el formato de los datos (email válido, campos requeridos)

### Instrucciones para Probar

#### Opción 1: Swagger UI (Recomendado - Sin instalación)

- Abre: <https://project-planning-cloud-api.onrender.com/docs>
- Busca "POST /api/v1/auth/register"

3. Click "Try it out"
4. Rellena los campos con los datos del ejemplo
5. Click "Execute"

#### Opción 2: cURL

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/register \
-H "Content-Type: application/json" \
-d '{
  "email": "juan.perez@ong.com",
  "password": "SecurePassword123",
  "nombre": "Juan",
  "apellido": "Pérez",
  "ong": "ONG Solidaria",
  "role": "MEMBER"
}'
```

#### Opción 3: Postman

1. Nueva petición → POST
2. URL: <https://project-planning-cloud-api.onrender.com/api/v1/auth/register>
3. Tab "Body" → raw → JSON
4. Pega el JSON del ejemplo
5. Click "Send"

## 2. Login de Usuario

Obtiene los tokens JWT para acceder a endpoints protegidos.

**Método:** POST **Ruta:** /api/v1/auth/login **Autenticación:** No requerida (público) **Código de Respuesta:** 200 OK

#### Parámetros

Campo	Tipo	Requerido	Descripción
email	String	Sí	Email registrado
password	String	Sí	Contraseña

#### Body de Prueba

```
{
  "email": "juan.perez@ong.com",
  "password": "SecurePassword123"
}
```

#### Response Exitoso (200)

```
{
  "access_token": "eyJ0eXAi0jKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIi0iI1NTBlODQwMC1lMjliLTQxZDQtYTxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6Imp1YW4ucGVyZXPAb25nLmNvbSIsInJvbGUi0iJNRU1CRViLCJleHaiOjE20j.eyJ0eXAi0jKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIi0iI1NTBlODQwMC1lMjliLTQxZDQtYTxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6Imp1YW4ucGVyZXPAb25nLmNvbSIsInJvbGUi0iJNRU1CRViLCJleHaiOjE20j",
  "refresh_token": "eyJ0eXAi0jKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIi0iI1NTBlODQwMC1lMjliLTQxZDQtYTxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6Imp1YW4ucGVyZXPAb25nLmNvbSIsInJvbGUi0iJNRU1CRViLCJleHaiOjE20j",
  "token_type": "bearer"
}
```

#### Guardar Tokens

**Importante:** Guarda estos tokens en tu cliente (sesión, localStorage, etc.):

- access\_token → Usar para peticiones autenticadas (válido 15 min)
- refresh\_token → Guardar seguro para renovar tokens (válido 24h)

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Credenciales incorrectas	{"detail": "Incorrect email or password"}	Verifica que el email y contraseña sean correctos
422	Error de validación Pydantic	{"detail": [{"loc": ["body", "email"], "msg": "field required", "type": "value_error.missing"}]}	Revisa que todos los campos requeridos estén presentes

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado - Sin instalación)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/auth/login"
3. Click "Try it out"
4. Rellena los campos con los datos del ejemplo
5. Click "Execute"
6. Copia el access\_token de la respuesta

##### Opción 2: cURL

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "juan.perez@ong.com",
  "password": "SecurePassword123"
}'
```

##### Guardar token en variable (bash):

```
TOKEN=$(curl -s -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{"email": "juan.perez@ong.com", "password": "SecurePassword123"}' \
| jq -r '.access_token')

echo $TOKEN
```

**3 Refrescar Token**

Obtiene un nuevo access\_token cuando el anterior expira.

**Método:** POST **Ruta:** /api/v1/auth/refresh **Autenticación:** No requerida (público) **Código de Respuesta:** 200 OK

## Parámetros

Campo	Tipo	Requerido	Descripción
refresh_token	String	Sí	Token de refresco del login

## **Body de Prueba**

```
{  
    "refresh_token":  
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtYTcxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6Imp1YW4ucGVyZXpAb25nLmNvbSIsInJvbGUiOiJNRU1CRVIiLCJleHAiOjE2  
}
```

## Response Exitoso (200)

```
{  
    "access_token":  
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjIiLTQxZDQtYTcxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6ImplYW4ucGVyZXpAb25nLmNvbSIsInJvbGUiOiJNRU1CRViilCJleHAiOjE2C  
    "refresh_token":  
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjIiLTQxZDQtYTcxNi00NDY2NTU0NDAwMDAiLCJlbWFpbCI6ImplYW4ucGVyZXpAb25nLmNvbSIsInJvbGUiOiJNRU1CRViilCJleHAiOjE2C  
    "token_type": "bearer"  
}
```

## Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Refresh token inválido o expirado	{"detail": "Invalid refresh token"}	Haz login nuevamente para obtener nuevos tokens
401	Usuario no encontrado	{"detail": "User not found"}	El usuario asociado al token ya no existe, haz login nuevamente
422	Error de validación Pydantic	{"detail": [{"loc": ["body", "refresh_token"], "msg": "field required", "type": "value_error.missing"}]}	Verifica que el campo refresh_token esté presente

## Endpoints de Proyectos

1 Clear Proyecto

Crea un nuevo proyecto con etapas y pedidos anidados en una sola transacción.

**Método:** POST **Ruta:** /api/v1/projects **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 201 Created **Propietario:** El usuario autenticado

## Parámetros

| Campo | Tipo | Requerido | Descripción | | ----- | ----- | ----- | ----- | | titulo | string | Sí | Título del proyecto | Mínimo 5 caracteres | | descripción | string | Sí | Descripción detallada | Mínimo 20 caracteres | | tipo | string | Sí | Tipo de proyecto | Ej: "Infraestructura", "Social" | | país | string | Sí | País | Ej: "Argentina" | | provincia | string | Sí | Provincia | Ej: "Buenos Aires" | | ciudad | string | Sí | Ciudad | Ej: "La Plata" | | barrio | string | No | Barrio/Localidad | Opcional | | estado | enum | No | Estado del proyecto | Default: en\_planificacion (ver enums) | | bonita\_case\_id | string | No | ID de caso Bonita | Opcional, para integración | | bonita\_process\_instance\_id | integer | No | ID instancia Bonita | Opcional | | etapas | array | No | Array de etapas anidadas | Ver estructura abajo |

## Estructura de Etapas

```
{  
    "nombre": "string - Nombre de la etapa",  
    "descripcion": "string - Descripción de la etapa",  
    "fecha_inicio": "YYYY-MM-DD",  
    "fecha_fin": "YYYY-MM-DD",  
    "pedidos": [  
        {  
            "tipo": "enum - Tipo de pedido",  
            "descripcion": "string - Descripción",  
            "monto": "float - Opcional, para económico",  
            "moneda": "string - Opcional, código de moneda",  
            "cantidad": "integer - Opcional, para otros tipos",  
            "unidad": "string - Opcional, unidad de medida"  
        }  
    ]  
}
```

## **Body de Prueba (Ejemplo Completo)**

```
{  
  "titulo": "Centro Comunitario La Plata",  
  "descripcion": "Construcción de un nuevo centro comunitario con servicios sociales, biblioteca y áreas de educación",  
  "tipo": "Infraestructura Social",  
  "pais": "Argentina",  
  "provincia": "Buenos Aires",  
  "ciudad": "La Plata",  
  "barrio": "Centro",  
  "estado": "en_planificacion",  
  "bonita_case_id": null,  
  "bonita_process_instance_id": null,  
  "etapas": [  
    {  
      "nombre": "Fase 1 - Cimientos",  
      "descripcion": "Preparación del terreno y construcción de cimientos",  
      "fecha_inicio": "2024-11-01",  
      "fecha_fin": "2024-12-31",  
      "pedidos": [  
        {  
          "tipo": "economico",  
          "descripcion": "Presupuesto para materiales de cimentación",  
          "monto": 50000.0,  
          "moneda": "ARS"  
        }  
      ]  
    }  
  ]  
}
```

```

        },
        {
            "tipo": "mano_obra",
            "descripcion": "Mano de obra para excavación y cimientos",
            "cantidad": 20,
            "unidad": "jornadas"
        }
    ],
    {
        "nombre": "Fase 2 - Estructura",
        "descripcion": "Construcción de estructura principal del edificio",
        "fecha_inicio": "2025-01-01",
        "fecha_fin": "2025-03-31",
        "pedidos": [
            {
                "tipo": "materiales",
                "descripcion": "Acero y hormigón para estructura",
                "cantidad": 100,
                "unidad": "toneladas"
            },
            {
                "tipo": "transporte",
                "descripcion": "Transporte de materiales de construcción",
                "cantidad": 10,
                "unidad": "viajes"
            }
        ]
    }
}

```

#### Response Exitoso (201)

```
{
    "id": "123e4567-e89b-12d3-a456-426614174000",
    "user_id": "550e8400-e29b-41d4-a716-446655440000",
    "titulo": "Centro Comunitario La Plata",
    "descripcion": "Construcción de un nuevo centro comunitario con servicios sociales, biblioteca y áreas de educación",
    "tipo": "Infraestructura Social",
    "pais": "Argentina",
    "provincia": "Buenos Aires",
    "ciudad": "La Plata",
    "barrio": "Centro",
    "estado": "en planificación",
    "bonita_case_id": null,
    "bonita_process_instance_id": null,
    "created_at": "2024-10-22T14:30:00+00:00",
    "updated_at": "2024-10-22T14:30:00+00:00",
    "etapas": [
        {
            "id": "223e4567-e89b-12d3-a456-426614174111",
            "proyecto_id": "123e4567-e89b-12d3-a456-426614174000",
            "nombre": "Fase 1 - Cimientos",
            "descripcion": "Preparación del terreno y construcción de cimientos",
            "fecha_inicio": "2024-11-01",
            "fecha_fin": "2024-12-31",
            "pedidos": [
                {
                    "id": "323e4567-e89b-12d3-a456-426614174222",
                    "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
                    "tipo": "económico",
                    "descripcion": "Presupuesto para materiales de cimentación",
                    "estado": "PENDIENTE",
                    "monto": 50000.0,
                    "moneda": "ARS",
                    "cantidad": null,
                    "unidad": null
                },
                {
                    "id": "323e4567-e89b-12d3-a456-426614174223",
                    "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
                    "tipo": "mano_obra",
                    "descripcion": "Mano de obra para excavación y cimientos",
                    "estado": "PENDIENTE",
                    "monto": null,
                    "moneda": null,
                    "cantidad": 20,
                    "unidad": "jornadas"
                }
            ]
        },
        {
            "id": "223e4567-e89b-12d3-a456-426614174112",
            "proyecto_id": "123e4567-e89b-12d3-a456-426614174000",
            "nombre": "Fase 2 - Estructura",
            "descripcion": "Construcción de estructura principal del edificio",
            "fecha_inicio": "2025-01-01",
            "fecha_fin": "2025-03-31",
            "pedidos": [
                {
                    "id": "323e4567-e89b-12d3-a456-426614174224",
                    "etapa_id": "223e4567-e89b-12d3-a456-426614174112",
                    "tipo": "materiales",
                    "descripcion": "Acero y hormigón para estructura",
                    "estado": "PENDIENTE",
                    "monto": null,
                    "moneda": null,
                    "cantidad": 100,
                    "unidad": "toneladas"
                }
            ]
        }
    ]
}
```

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado - Sin instalación)

- Abre: <https://project-planning-cloud-api.onrender.com/docs>
- Busca "POST /api/v1/projects"
- Click "Try it out"
- Obtén un token con POST /api/v1/auth/login primero

5. En el authorization button de arriba, pega: Bearer {tu\_token}
6. Completa el JSON con los datos del ejemplo
7. Click "Execute"

#### Opción 2: cURL

```
# Primero obtén un token con login
TOKEN="tu_access_token_aquí"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/projects \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "titulo": "Centro Comunitario La Plata",
  "descripcion": "Construcción de un nuevo centro comunitario con servicios sociales, biblioteca y áreas de educación",
  "tipo": "Infraestructura Social",
  "pais": "Argentina",
  "provincia": "Buenos Aires",
  "ciudad": "La Plata",
  "barrio": "Centro",
  "estado": "en_planificación",
  "etapas": [
    {
      "nombre": "Fase 1 - Cimientos",
      "descripcion": "Preparación del terreno y construcción de cimientos",
      "fecha_inicio": "2024-11-01",
      "fecha_fin": "2024-12-31",
      "pedidos": [
        {
          "tipo": "económico",
          "descripcion": "Presupuesto para materiales de cimentación",
          "monto": 50000.00,
          "moneda": "ARS"
        }
      ]
    }
  ]
}'
```

#### Opción 3: Postman

1. Nueva petición → POST
2. URL: <https://project-planning-cloud-api.onrender.com/api/v1/projects>
3. Tab "Authorization" → Type: "Bearer Token" → Token: [pega tu access\_token]
4. Tab "Body" → raw → JSON
5. Pega el JSON del ejemplo
6. Click "Send"

## 2 Obtener Proyecto por ID

Recupera un proyecto específico con todas sus etapas y pedidos anidados.

**Método:** GET **Ruta:** /api/v1/projects/{project\_id} **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK

#### Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a obtener

#### Ejemplo de Ruta

```
GET /api/v1/projects/123e4567-e89b-12d3-a456-426614174000
```

#### Response Exitoso (200)

Retorna la estructura completa del proyecto igual que en el POST (ver ejemplo anterior).

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido en el header Authorization
404	Proyecto no encontrado	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}	Verifica que el project_id sea correcto

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "GET /api/v1/projects/{project\_id}"
3. Click "Try it out"
4. Pega el UUID en "project\_id"
5. Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aquí"
PROJECT_ID="123e4567-e89b-12d3-a456-426614174000"

curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID \
-H "Authorization: Bearer $TOKEN"
```

## 3 Actualizar Proyecto (Parcial)

Actualiza campos específicos del proyecto usando PATCH (solo se actualizan los campos proporcionados).

**Método:** PATCH **Ruta:** /api/v1/projects/{project\_id} **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK **Restricción:** Solo el propietario del proyecto puede actualizar

#### Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a actualizar

#### Parámetros Actualizables

Todos los campos son opcionales. Solo se actualizan los que proporcionas:

Campo	Tipo	Descripción
titulo	string	Nuevo título
descripcion	string	Nueva descripción
tipo	string	Nuevo tipo
pais	string	Nuevo país
provincia	string	Nueva provincia
ciudad	string	Nueva ciudad
barrio	string	Nuevo barrio
estado	enum	Nuevo estado
bonita_case_id	string	ID de caso Bonita
bonita_process_instance_id	integer	ID de instancia Bonita

#### Body de Prueba

```
{
  "estado": "buscando_financiamiento",
  "bonita_case_id": "CASE-2024-001",
  "bonita_process_instance_id": 12345
}
```

#### Response Exitoso (200)

Retorna el proyecto actualizado con todos sus datos.

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario	{"detail": "Forbidden - User is not the owner of this resource"}	Solo el dueño del proyecto puede actualizar
404	Proyecto no existe	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}	Verifica que el project_id sea correcto
422	Validación fallida	{"detail": [{"loc": ["body", "bonita_case_id"], "msg": "string does not match regex", "type": "value_error.str.regex"}]}	Revisa el formato de los datos proporcionados

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

- Abre: <https://project-planning-cloud-api.onrender.com/docs>
- Busca "PATCH /api/v1/projects/{project\_id}"
- Click "Try it out"
- Pega el UUID en "{project\_id}"
- Completa el JSON con los datos del ejemplo
- Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
PROJECT_ID="123e4567-e89b-12d3-a456-426614174000"

curl -X PATCH https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "estado": "buscando_financiamiento",
  "bonita_case_id": "CASE-2024-001"
}'
```

## 4 Eliminar Proyecto

Elimina un proyecto y **toda su estructura anidada** (etapas, pedidos, ofertas).

**Método:** DELETE **Ruta:** /api/v1/projects/{project\_id} **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 204 No Content **Restricción:** Solo el propietario del proyecto puede eliminar \*\* Cascada:\*\* Elimina también etapas, pedidos y ofertas

#### Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a eliminar

#### Confirmación

La eliminación es **permanente e irrevocable**. No hay confirmación adicional.

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede eliminarlo
404	Proyecto no encontrado	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}	Verifica que el project_id sea correcto

## Instrucciones para Probar

### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "DELETE /api/v1/projects/{project\_id}"
3. Click "Try it out"
4. Pega el UUID en "project\_id"
5. Click "Execute"

### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
PROJECT_ID="123e4567-e89b-12d3-a456-426614174000"

curl -X DELETE https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID \
-H "Authorization: Bearer $TOKEN"
```

### Resultado esperado:

- Sin body en la respuesta
- Status code: 204 No Content

## Endpoints de Pedidos

### 1. Crear Pedido para Etapa Existente

Crea un nuevo pedido dentro de una etapa específica de un proyecto.

**Método:** POST **Ruta:** /api/v1/projects/{project\_id}/etapas/{etapa\_id}/pedidos **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 201 Created **Restricción:** Solo el propietario del proyecto

#### Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto dueño
etapa_id	UUID	ID de la etapa dentro del proyecto

#### Parámetros

Campo	Tipo	Requerido	Descripción
tipo	enum	Sí	Tipo de pedido
descripcion	string	Sí	Descripción detallada
monto	float	No	Monto (para tipo económico)
moneda	string	No	Código de moneda (ARS, USD, etc)
cantidad	integer	No	Cantidad (para otros tipos)
unidad	string	No	Unidad de medida

#### Body de Prueba

```
{
  "tipo": "economico",
  "descripcion": "Presupuesto para pintura de las paredes interiores",
  "monto": 15000.0,
  "moneda": "ARS"
}
```

Ejemplo alternativo (materiales):

```
{
  "tipo": "materiales",
  "descripcion": "Ladrillos para mampostería de paredes",
  "cantidad": 5000,
  "unidad": "ladrillos"
}
```

#### Response Exitoso (201)

```
{
  "id": "423e4567-e89b-12d3-a456-426614174333",
  "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
  "tipo": "economico",
  "descripcion": "Presupuesto para pintura de las paredes interiores",
  "estado": "PENDIENTE",
  "monto": 15000.0,
  "moneda": "ARS",
  "cantidad": null,
  "unidad": null
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario del proyecto	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede crear pedidos
404	Proyecto no encontrado	{"detail": "Proyecto with id ... not found"}	Verifica que el project_id sea correcto
404	Etapa no encontrada	{"detail": "Etapa with id ... not found in proyecto"}	Verifica que el etapa_id pertenezca al proyecto
422	Validación fallida	{"detail": [{"loc": ["body", "tipo"], "msg": "value is not a valid enumeration member", "type": "type_error.enum"}]}	Revisa que el tipo sea válido (economico, materiales, mano obra, transporte, equipamiento)

## Instrucciones para Probar

### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/projects/{project\_id}/etapas/{etapa\_id}/pedidos"
3. Click "Try it out"
4. Pega los UUIDs de proyecto y etapa
5. Completa el JSON con los datos del ejemplo
6. Click "Execute"

### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
PROJECT_ID="123e4567-e89b-12d3-a456-426614174000"
ETAPA_ID="223e4567-e89b-12d3-a456-426614174111"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID/etapas/$ETAPA_ID/pedidos \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "tipo": "economico",
  "descripcion": "Presupuesto para pintura de las paredes interiores",
  "monto": 15000.00,
  "moneda": "ARS"
}'
```

## 2 Listar Pedidos de Proyecto

Obtiene todos los pedidos de un proyecto con filtrado opcional por estado.

**Método:** GET **Ruta:** /api/v1/projects/{project\_id}/pedidos **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK

### Path Parameters

#### Parámetro Tipo Descripción

project\_id UUID ID del proyecto

### Query Parameters (Opcionales)

Parámetro	Tipo	Descripción
estado	enum	Filtrar por estado: PENDIENTE o COMPROMETIDO o COMPLETADO

### Ejemplos de Ruta

```
GET /api/v1/projects/123e4567-e89b-12d3-a456-426614174000/pedidos
GET /api/v1/projects/123e4567-e89b-12d3-a456-426614174000/pedidos?estado=PENDIENTE
GET /api/v1/projects/123e4567-e89b-12d3-a456-426614174000/pedidos?estado=COMPLETADO
```

### Response Exitoso (200)

```
[{"id": "423e4567-e89b-12d3-a456-426614174333",
 "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
 "tipo": "economico",
 "descripcion": "Presupuesto para pintura de las paredes interiores",
 "estado": "PENDIENTE",
 "monto": 15000.0,
 "moneda": "ARS",
 "cantidad": null,
 "unidad": null
},
 {"id": "423e4567-e89b-12d3-a456-426614174334",
 "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
 "tipo": "mano_obra",
 "descripcion": "Mano de obra para pintura",
 "estado": "COMPROMETIDO",
 "monto": null,
 "moneda": null,
 "cantidad": 5,
 "unidad": "jornadas"
}]
```

### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
404	Proyecto no encontrado	{"detail": "Proyecto with id ... not found"}	Verifica que el project_id sea correcto
422	Filtro de estado inválido	{"detail": [{"loc": ["query", "estado"], "msg": "string does not match regex", "type": "value_error.str.regex"}]}	El parámetro estado debe ser "pendiente" o "completado"

## Instrucciones para Probar

### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "GET /api/v1/projects/{project\_id}/pedidos"
3. Click "Try it out"
4. Pega el UUID del proyecto
5. (Opcional) Usa el parámetro estado para filtrar
6. Click "Execute"

### Opción 2: cURL

```

TOKEN="tu_access_token_aqui"
PROJECT_ID="123e4567-e89b-12d3-a456-426614174000"

# Todos los pedidos
curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID/pedidos \
-H "Authorization: Bearer $TOKEN"

# Solo pendientes
curl -X GET "https://project-planning-cloud-api.onrender.com/api/v1/projects/$PROJECT_ID/pedidos?estado=PENDIENTE" \
-H "Authorization: Bearer $TOKEN"

```

---

### 3 Eliminar Pedido

Elimina un pedido específico (también elimina sus ofertas asociadas).

**Método:** DELETE **Ruta:** /api/v1/pedidos/{pedido\_id} **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 204 No Content **Restricción:** Solo el propietario del proyecto dueño \*\* Cascada:\*\* Elimina también todas las ofertas del pedido

#### Path Parameters

Parámetro	Tipo	Descripción
pedido_id	UUID	ID del pedido a eliminar

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario del proyecto	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede eliminar pedidos
404	Pedido no encontrado	{"detail": "Pedido with id ... not found"}	Verifica que el pedido_id sea correcto

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "DELETE /api/v1/pedidos/{pedido\_id}"
3. Click "Try it out"
4. Pega el UUID del pedido
5. Click "Execute"

##### Opción 2: cURL

```

TOKEN="tu_access_token_aqui"
PEDIDO_ID="423e4567-e89b-12d3-a456-426614174333"

curl -X DELETE https://project-planning-cloud-api.onrender.com/api/v1/pedidos/$PEDIDO_ID \
-H "Authorization: Bearer $TOKEN"

```

---

## Endpoints de Ofertas

### 1 Crear Oferta para Pedido

Crea una nueva oferta para un pedido específico. Un usuario propone sus servicios/productos para cubrirlo.

**Método:** POST **Ruta:** /api/v1/pedidos/{pedido\_id}/ofertas **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 201 Created **Restricción:** Pedido debe estar en estado PENDIENTE

#### Path Parameters

Parámetro	Tipo	Descripción
pedido_id	UUID	ID del pedido a ofertar

#### Parámetros

| Campo | Tipo | Requerido | Descripción | | ----- | ----- | ----- | ----- | ----- | | descripción | string | Sí | Descripción de la oferta | Mínimo 10 caracteres | | monto\_ofrecido | float | No | Monto ofrecido (opcional) | Para comparar con presupuesto |

#### Body de Prueba

```
{
  "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura con materiales de primera calidad. Garantizo buen acabado y entrega a tiempo.",
  "monto_ofrecido": 14500.0
}
```

#### Response Exitoso (201)

```
{
  "id": "523e4567-e89b-12d3-a456-426614174444",
  "pedido_id": "423e4567-e89b-12d3-a456-426614174333",
  "user_id": "550e8400-e29b-41d4-a716-446655440001",
  "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura con materiales de primera calidad. Garantizo buen acabado y entrega a tiempo.",
  "monto_ofrecido": 14500.0,
  "estado": "pendiente",
  "created_at": "2024-10-22T15:00:00+00:00",
  "updated_at": "2024-10-22T15:00:00+00:00"
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
404	Pedido no encontrado	{"detail": "Pedido with id ... not found"}	Verifica que el pedido_id sea correcto

```
422    Validación fallida      {"detail": [{"loc": ["body", "descripcion"], "msg": "field required", "type": "value_error.missing"}]}           La descripción es requerida
```

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/pedidos/{pedido\_id}/ofertas"
3. Click "Try it out"
4. Pega el UUID del pedido
5. Completa el JSON con los datos del ejemplo
6. Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
PEDIDO_ID="423e4567-e89b-12d3-a456-426614174333"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/pedidos/$PEDIDO_ID/ofertas \
-H "Authorization: Bearer $TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura con materiales de primera calidad.",
  "monto_ofrecido": 14500.00
}'
```

## 2 Listar Ofertas de Pedido

Obtiene todas las ofertas para un pedido específico (con detalles del usuario oferente).

**Método:** GET **Ruta:** /api/v1/pedidos/{pedido\_id}/ofertas **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK **Restricción:** Solo el propietario del proyecto puede ver las ofertas

#### Path Parameters

##### Parámetro Tipo Descripción

pedido\_id UUID ID del pedido

#### Response Exitoso (200)

```
[ {
  "id": "523e4567-e89b-12d3-a456-426614174444",
  "pedido_id": "423e4567-e89b-12d3-a456-426614174333",
  "user_id": "550e8400-e29b-41d4-a716-446655440001",
  "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura...",
  "monto_ofrecido": 14500.0,
  "estado": "pendiente",
  "created_at": "2024-10-22T15:00:00+00:00",
  "updated_at": "2024-10-22T15:00:00+00:00",
  "user": {
    "id": "550e8400-e29b-41d4-a716-446655440001",
    "email": "maria.gonzalez@empresa.com",
    "nombre": "María",
    "apellido": "González",
    "ong": "Empresa Pintores"
  }
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario del proyecto	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede ver las ofertas de sus pedidos
404	Pedido no encontrado	{"detail": "Pedido with id ... not found"}	Verifica que el pedido_id sea correcto

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "GET /api/v1/pedidos/{pedido\_id}/ofertas"
3. Click "Try it out"
4. Pega el UUID del pedido
5. Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
PEDIDO_ID="423e4567-e89b-12d3-a456-426614174333"

curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/pedidos/$PEDIDO_ID/ofertas \
-H "Authorization: Bearer $TOKEN"
```

## 3 Aceptar Oferta

El propietario del proyecto acepta una oferta, comprometiéndose a usar los servicios del oferente.

**Método:** POST **Ruta:** /api/v1/ofertas/{oferta\_id}/accept **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK **Restricción:** Solo el propietario del proyecto

#### Efecto Cascada:

- Oferta estado → aceptada
- Pedido estado → COMPROMETIDO
- Otras ofertas para mismo pedido → rechazada

#### Path Parameters

Parámetro	Tipo	Descripción
oferta_id	UUID	ID de la oferta a aceptar

#### Body

Sin body requerido.

#### Response Exitoso (200)

```
{  
    "id": "523e4567-e89b-12d3-a456-426614174444",  
    "pedido_id": "423e4567-e89b-12d3-a456-426614174333",  
    "user_id": "550e8400-e29b-41d4-a716-446655440001",  
    "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura...",  
    "monto_ofrecido": 14500.0,  
    "estado": "aceptada",  
    "created_at": "2024-10-22T15:00:00+00:00",  
    "updated_at": "2024-10-22T15:10:00+00:00"  
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario del proyecto	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede aceptar ofertas
404	Oferta no encontrada	{"detail": "Oferta with id ... not found"}	Verifica que el oferta_id sea correcto
400	Pedido ya completado	{"detail": "Cannot accept oferta for completed pedido"}	No se pueden aceptar ofertas de pedidos completados
400	Pedido ya comprometido	{"detail": "Cannot accept oferta for committed pedido"}	No se pueden aceptar ofertas de pedidos ya comprometidos

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/ofertas/{oferta\_id}/accept"
3. Click "Try it out"
4. Pega el UUID de la oferta
5. Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"  
OFERTA_ID="523e4567-e89b-12d3-a456-426614174444"  
  
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/ofertas/$OFERTA_ID/accept \  
-H "Authorization: Bearer $TOKEN"
```

## 4 Rechazar Oferta

El propietario del proyecto rechaza una oferta.

**Método:** POST **Ruta:** /api/v1/ofertas/{oferta\_id}/reject **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK **Restricción:** Solo el propietario del proyecto

#### Path Parameters

Parámetro	Tipo	Descripción
oferta_id	UUID	ID de la oferta a rechazar

#### Response Exitoso (200)

```
{  
    "id": "523e4567-e89b-12d3-a456-426614174444",  
    "pedido_id": "423e4567-e89b-12d3-a456-426614174333",  
    "user_id": "550e8400-e29b-41d4-a716-446655440001",  
    "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura...",  
    "monto_ofrecido": 14500.0,  
    "estado": "rechazada",  
    "created_at": "2024-10-22T15:00:00+00:00",  
    "updated_at": "2024-10-22T15:15:00+00:00"  
}
```

#### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el propietario del proyecto	{"detail": "You are not the owner of this project"}	Solo el dueño del proyecto puede rechazar ofertas
404	Oferta no encontrada	{"detail": "Oferta with id ... not found"}	Verifica que el oferta_id sea correcto

#### Instrucciones para Probar

##### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/ofertas/{oferta\_id}/reject"
3. Click "Try it out"
4. Pega el UUID de la oferta
5. Click "Execute"

##### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"  
OFERTA_ID="523e4567-e89b-12d3-a456-426614174444"
```

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/ofertas/$OFERTA_ID/reject \
-H "Authorization: Bearer $TOKEN"
```

---

## 5 Confirmar Realización de Oferta

El usuario que creó la oferta confirma que realizó el trabajo/servicio.

**Método:** POST **Ruta:** /api/v1/ofertas/{oferta\_id}/confirmar-realizacion **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK **Restricción:** Solo el creador de la oferta **Precondiciones:**

- Oferta debe estar en estado aceptada
- Pedido debe estar en estado COMPROMETIDO

### Path Parameters

#### Parámetro Tipo Descripción

oferta\_id UUID ID de la oferta

### Body

Sin body requerido.

### Response Exitoso (200)

```
{
  "message": "Realización confirmada exitosamente",
  "success": true,
  "oferta_id": "523e4567-e89b-12d3-a456-426614174444",
  "oferta_estado": "aceptada",
  "pedido_id": "423e4567-e89b-12d3-a456-426614174333",
  "pedido_estado_anterior": "COMPROMETIDO",
  "pedido_estado_nuevo": "COMPLETADO",
  "confirmed_at": "2024-10-22T15:30:00+00:00"
}
```

### Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{"detail": "Invalid or expired token"}	Proporciona un access_token válido
403	No eres el creador de la oferta	{"detail": "You are not the creator of this oferta"}	Solo quien creó la oferta puede confirmar su realización
404	Oferta no encontrada	{"detail": "Oferta with id ... not found"}	Verifica que el oferta_id sea correcto
400	Compromiso ya confirmado	{"detail": "This commitment has already been confirmed and marked as completed"}	Esta oferta ya fue confirmada previamente
400	Estado de pedido incorrecto	{"detail": "Cannot confirm realization. Pedido is in state pendiente, expected comprometido"}	El pedido debe estar en estado COMPROMETIDO para poder confirmar

### Instrucciones para Probar

#### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "POST /api/v1/ofertas/{oferta\_id}/confirmar-realizacion"
3. Click "Try it out"
4. Pega el UUID de la oferta
5. Click "Execute"

#### Opción 2: cURL

```
TOKEN="tu_access_token_aqui"
OFERTA_ID="523e4567-e89b-12d3-a456-426614174444"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/ofertas/$OFERTA_ID/confirmar-realizacion \
-H "Authorization: Bearer $TOKEN"
```

---

## 6 Obtener Mis Compromisos

Obtiene todas las ofertas que el usuario autenticado ha creado (sus compromisos).

**Método:** GET **Ruta:** /api/v1/ofertas/mis-compromisos **Autenticación:** Requerida (Bearer Token) **Código de Respuesta:** 200 OK

### Query Parameters (Opcionales)

Parámetro	Tipo	Descripción
estado_pedido	enum	Filtrar por estado del pedido: COMPROMETIDO O COMPLETADO

### Ejemplos de Ruta

```
GET /api/v1/ofertas/mis-compromisos
GET /api/v1/ofertas/mis-compromisos?estado_pedido=COMPROMETIDO
GET /api/v1/ofertas/mis-compromisos?estado_pedido=COMPLETADO
```

### Response Exitoso (200)

```
[
  {
    "id": "523e4567-e89b-12d3-a456-426614174444",
    "pedido_id": "423e4567-e89b-12d3-a456-426614174333",
    "user_id": "550e8400-e29b-41d4-a716-446655440001",
    "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura...",
    "monto_ofrecido": 14500.0,
    "estado": "aceptada",
    "created_at": "2024-10-22T15:00:00+00:00",
    "updated_at": "2024-10-22T15:10:00+00:00",
    "pedido": {
      "id": "423e4567-e89b-12d3-a456-426614174333",
      "user_id": "550e8400-e29b-41d4-a716-446655440001",
      "descripcion": "Tengo disponibilidad inmediata para realizar trabajos de pintura...",
      "monto_ofrecido": 14500.0,
      "estado": "aceptada",
      "created_at": "2024-10-22T15:00:00+00:00",
      "updated_at": "2024-10-22T15:10:00+00:00"
    }
  }
]
```

```

        "id": "423e4567-e89b-12d3-a456-426614174333",
        "etapa_id": "223e4567-e89b-12d3-a456-426614174111",
        "tipo": "economico",
        "descripcion": "Presupuesto para pintura de las paredes interiores",
        "estado": "COMPROMETIDO",
        "monto": 15000.0,
        "moneda": "ARS",
        "cantidad": null,
        "unidad": null
    }
]

```

## Errores Posibles

Código	Descripción	Ejemplo de Error	Solución
401	Token inválido o faltante	{ "detail": "Invalid or expired token"}	Proporciona un access_token válido
422	Filtro de estado inválido	{ "detail": [ { "loc": [ "query", "estado_pedido" ], "msg": "value is not a valid enumeration member", "type": "type_error.enum" } ] }	El parámetro estado_pedido debe ser un valor válido del enum EstadoPedido

## Instrucciones para Probar

### Opción 1: Swagger UI (Recomendado)

1. Abre: <https://project-planning-cloud-api.onrender.com/docs>
2. Busca "GET /api/v1/ofertas/mis-compromisos"
3. Click "Try it out"
4. (Opcional) Usa el parámetro estado\_pedido para filtrar
5. Click "Execute"

### Opción 2: cURL

```

TOKEN="tu_access_token_aqui"

# Todos los compromisos
curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/ofertas/mis-compromisos \
-H "Authorization: Bearer $TOKEN"

# Solo comprometidos
curl -X GET "https://project-planning-cloud-api.onrender.com/api/v1/ofertas/mis-compromisos?estado_pedido=COMPROMETIDO" \
-H "Authorization: Bearer $TOKEN"

```

## Enumeraciones

### EstadoProyecto

Estados disponibles para los proyectos:

```

"borrador"           # Proyecto en borrador
"en_planificacion"  # En fase de planificación (default)
"buscando_financiamiento" # Buscando fondos
"en_ejecucion"       # En ejecución
"completo"          # Proyecto completado

```

Ejemplo de uso:

```
{
    "estado": "en_planificacion"
}
```

### TipoPedido

Tipos disponibles para los pedidos:

```

"economico"      # Solicitud de presupuesto/dinero
"materiales"     # Solicitud de materiales
"mano_obra"       # Solicitud de mano de obra
"transporte"     # Solicitud de transporte
"equipamiento"   # Solicitud de equipamiento/máquinas

```

### Recomendaciones de uso:

- economico → Usa monto + moneda
- Otros tipos → Usa cantidad + unidad

Ejemplos:

```

// Económico
{
    "tipo": "economico",
    "monto": 50000.00,
    "moneda": "ARS"
}

// Materiales
{
    "tipo": "materiales",
    "cantidad": 100,
    "unidad": "metros"
}

// Mano de obra
{
    "tipo": "mano_obra",
    "cantidad": 20,
    "unidad": "jornadas"
}

```

### EstadoPedido

Estados disponibles para los pedidos:

```
"PENDIENTE"      # Sin oferta aceptada (default)
"COMPROMETIDO"   # Con oferta aceptada, esperando confirmación
"COMPLETADO"     # Realización confirmada
```

#### Flujo de estado:

PENDIENTE → (aceptar oferta) → COMPROMETIDO → (confirmar realización) → COMPLETADO

---

#### EstadoOferta

Estados disponibles para las ofertas:

```
"pendiente"      # Recién creada, esperando decisión (default)
"aceptada"       # Proyecto aceptó la oferta
"rechazada"      # Proyecto rechazó la oferta
```

---

#### UserRole

Roles disponibles para usuarios:

```
"MEMBER"        # Usuario regular
"COUNCIL"        # Miembro de consejo (elevados permisos)
```

---

## Flujo Completo de Ejemplo

A continuación, un ejemplo paso a paso de cómo usar la API desde cero:

### Paso 1: Registrar Dos Usuarios

#### Usuario 1 - Proyecto Owner (María)

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/register \
-H "Content-Type: application/json" \
-d '{
  "email": "maria@ong.com",
  "password": "Password123",
  "nombre": "María",
  "apellido": "López",
  "ong": "ONG Social",
  "role": "MEMBER"
}'
```

Guarda el id returnedo. Ejemplo: 550e8400-e29b-41d4-a716-446655440000

#### Usuario 2 - Oferente (Carlos)

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/register \
-H "Content-Type: application/json" \
-d '{
  "email": "carlos@empresa.com",
  "password": "Password456",
  "nombre": "Carlos",
  "apellido": "García",
  "ong": "Empresa Constructora",
  "role": "MEMBER"
}'
```

Guarda el id returnedo. Ejemplo: 550e8400-e29b-41d4-a716-446655440001

### Paso 2: Login con Ambos Usuarios

#### María (Dueña del Proyecto)

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "maria@ong.com",
  "password": "Password123"
}'
```

Guarda el access\_token. Ejemplo para usar:

```
MARIA_TOKEN="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..."
```

#### Carlos (Oferente)

```
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{
  "email": "carlos@empresa.com",
  "password": "Password456"
}'
```

Guarda el access\_token:

```
CARLOS_TOKEN="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9..."
```

### Paso 3: María Crea un Proyecto

```
MARIA_TOKEN="tu_token_aqui"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/projects \
-H "Authorization: Bearer $MARIA_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "titulo": "Renovación Centro Comunitario",
  "descripcion": "Renovación completa del centro comunitario incluyendo pintura, pisos y techos",
  "tipo": "Infraestructura",
  "pais": "Argentina",
  "provincia": "Buenos Aires",
  "ciudad": "La Plata",
  "barrio": "Centro",
  "estado": "en_planificacion",
  "etapas": [
    {
      "nombre": "Remodelación Interior",
      "descripcion": "Todas las reparaciones interiores",
    }
  ]
}'
```

```

    "fecha_inicio": "2024-11-01",
    "fecha_fin": "2024-12-31",
    "pedidos": [
        {
            "tipo": "economico",
            "descripcion": "Presupuesto para pintura de interiores",
            "monto": 15000.00,
            "moneda": "ARS"
        }
    ]
}

```

#### Guarda los IDs retornados:

- project\_id: ID del proyecto (ej: 123e4567-e89b-12d3-a456-426614174000)
- pedido\_id: ID del pedido (ej: 423e4567-e89b-12d3-a456-426614174333)

#### Paso 4: Carlos Crea una Oferta

```

CARLOS_TOKEN="tu_token_aqui"
PEDIDO_ID="423e4567-e89b-12d3-a456-426614174333"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/pedidos/$PEDIDO_ID/ofertas \
-H "Authorization: Bearer $CARLOS_TOKEN" \
-H "Content-Type: application/json" \
-d '{
    "descripcion": "Soy pintor profesional con 15 años de experiencia. Ofrezco trabajo de calidad con materiales premium y garantía de acabado perfecto.",
    "monto_ofrecido": 14000.00
}'

```

#### Guarda el oferta\_id retornado:

- Ejemplo: 523e4567-e89b-12d3-a456-426614174444

#### Paso 5: María ve las Ofertas del Pedido

```

MARIA_TOKEN="tu_token_aqui"
PEDIDO_ID="423e4567-e89b-12d3-a456-426614174333"

curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/pedidos/$PEDIDO_ID/ofertas \
-H "Authorization: Bearer $MARIA_TOKEN"

```

Verá la oferta de Carlos con sus detalles.

#### Paso 6: María Acepta la Oferta de Carlos

```

MARIA_TOKEN="tu_token_aqui"
OFERTA_ID="523e4567-e89b-12d3-a456-426614174444"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/ofertas/$OFERTA_ID/accept \
-H "Authorization: Bearer $MARIA_TOKEN"

```

#### Cambios automáticos:

- Oferta estado: pendiente → aceptada
- Pedido estado: PENDIENTE → COMPROMETIDO

#### Paso 7: Carlos Confirma que Realizó el Trabajo

```

CARLOS_TOKEN="tu_token_aqui"
OFERTA_ID="523e4567-e89b-12d3-a456-426614174444"

curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/ofertas/$OFERTA_ID/confirmar-realizacion \
-H "Authorization: Bearer $CARLOS_TOKEN"

```

#### Cambios automáticos:

- Pedido estado: COMPROMETIDO → COMPLETADO

#### Paso 8: Carlos Visualiza sus Compromisos

```

CARLOS_TOKEN="tu_token_aqui"

curl -X GET https://project-planning-cloud-api.onrender.com/api/v1/ofertas/mis-compromisos \
-H "Authorization: Bearer $CARLOS_TOKEN"

```

Verá todas sus ofertas aceptadas con estado aceptada y sus pedidos asociados.

## Códigos de Error

Esta sección describe **todos los códigos de error posibles** que la API puede retornar, organizados por categoría.

### Códigos HTTP Estándar

Código	Nombre	Descripción
200	OK	Petición exitosa
201	Created	Recurso creado exitosamente
204	No Content	Eliminación exitosa (sin body)
400	Bad Request	Solicitud incorrecta o regla de negocio violada
401	Unauthorized	Sin autenticación o token inválido
403	Forbidden	No tienes permisos para esta acción
404	Not Found	Recurso no existe
422	Unprocessable Entity	Validación de datos fallida
500	Internal Server Error	Error inesperado del servidor

### Errores por Endpoint

## Autenticación

POST /api/v1/auth/register

Código	Error	Ejemplo
400	Email ya registrado	{"detail": "A user with this email already exists"}
400	Contraseña demasiado larga	{"detail": "Password cannot exceed 72 bytes in length"}
400	Datos inválidos	{"detail": "Invalid registration data"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "email"], "msg": "invalid email format", "type": "value_error.email"}]}

POST /api/v1/auth/login

Código	Error	Ejemplo
401	Credenciales incorrectas	{"detail": "Incorrect email or password"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "email"], "msg": "field required", "type": "value_error.missing"}]}

POST /api/v1/auth/refresh

Código	Error	Ejemplo
401	Refresh token inválido	{"detail": "Invalid refresh token"}
401	Usuario no encontrado	{"detail": "Invalid refresh token"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "refresh_token"], "msg": "field required", "type": "value_error.missing"}]}

## Proyectos

POST /api/v1/projects

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "titulo"], "msg": "ensure this value has at least 5 characters", "type": "value_error.any_str.min_length"}]}
422	Validación del servicio	{"detail": "Invalid proyecto data"}
500	Error del servidor	{"detail": "Error creating proyecto: Database error"}

GET /api/v1/projects/{project\_id}

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
404	Proyecto no encontrado	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}

PATCH /api/v1/projects/{project\_id}

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
403	No eres el dueño	{"detail": "Forbidden - User is not the owner of this resource"}
404	Proyecto no encontrado	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "bonita_case_id"], "msg": "string does not match regex", "type": "value_error.str.regex"}]}

DELETE /api/v1/projects/{project\_id}

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
403	No eres el dueño	{"detail": "You are not the owner of this project"}
404	Proyecto no encontrado	{"detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"}

## Pedidos

POST /api/v1/projects/{project\_id}/etapas/{etapa\_id}/pedidos

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
403	No eres el dueño del proyecto	{"detail": "You are not the owner of this project"}
404	Proyecto no encontrado	{"detail": "Proyecto with id ... not found"}
404	Etapa no encontrada	{"detail": "Etapa with id ... not found in proyecto"}
422	Validación Pydantic	{"detail": [{"loc": ["body", "tipo"], "msg": "value is not a valid enumeration member", "type": "type_error.enum"}]}

GET /api/v1/projects/{project\_id}/pedidos

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
404	Proyecto no encontrado	{"detail": "Proyecto with id ... not found"}
422	Filtro estado inválido	{"detail": [{"loc": ["query", "estado"], "msg": "string does not match regex", "type": "value_error.str.regex"}]}

DELETE /api/v1/pedidos/{pedido\_id}

Código	Error	Ejemplo
401	Token inválido o faltante	{"detail": "Invalid or expired token"}
403	No eres el dueño del proyecto	{"detail": "You are not the owner of this project"}
404	Pedido no encontrado	{"detail": "Pedido with id ... not found"}

## Ofertas

**POST /api/v1/pedidos/{pedido\_id}/ofertas**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
404	Pedido no encontrado {"detail": "Pedido with id ... not found"}	
422	Validación Pydantic {"detail": [{"loc": ["body", "descripcion"], "msg": "field required", "type": "value_error.missing"}]}	

**GET /api/v1/pedidos/{pedido\_id}/ofertas**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
403	No eres el dueño del proyecto {"detail": "You are not the owner of this project"}	
404	Pedido no encontrado {"detail": "Pedido with id ... not found"}	

**POST /api/v1/ofertas/{oferta\_id}/accept**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
403	No eres el dueño del proyecto {"detail": "You are not the owner of this project"}	
404	Oferta no encontrada {"detail": "Oferta with id ... not found"}	
400	Pedido ya completado {"detail": "Cannot accept oferta for completed pedido"}	
400	Pedido ya comprometido {"detail": "Cannot accept oferta for committed pedido"}	

**POST /api/v1/ofertas/{oferta\_id}/reject**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
403	No eres el dueño del proyecto {"detail": "You are not the owner of this project"}	
404	Oferta no encontrada {"detail": "Oferta with id ... not found"}	

**POST /api/v1/ofertas/{oferta\_id}/confirmar-realizacion**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
403	No eres el creador de la oferta {"detail": "You are not the creator of this oferta"}	
404	Oferta no encontrada {"detail": "Oferta with id ... not found"}	
400	Compromiso ya confirmado {"detail": "This commitment has already been confirmed and marked as completado"}	
400	Estado de pedido incorrecto {"detail": "Cannot confirm realization. Pedido is in state pendiente, expected comprometido"}	

**GET /api/v1/ofertas/mis-compromisos**

Código	Error	Ejemplo
401	Token inválido o faltante {"detail": "Invalid or expired token"}	
422	Filtro estado inválido {"detail": [{"loc": ["query", "estado_pedido"], "msg": "value is not a valid enumeration member", "type": "type_error.enum"}]}	

## Ejemplos de Respuestas de Error

### Error 401 - Sin Autenticación

```
{  
    "detail": "Invalid or expired token"  
}
```

### Error 403 - Sin Permiso

```
{  
    "detail": "You are not the owner of this project"  
}
```

### Error 404 - No Encontrado

```
{  
    "detail": "Proyecto with id 123e4567-e89b-12d3-a456-426614174000 not found"  
}
```

### Error 422 - Validación Fallida

```
{  
    "detail": [  
        {  
            "loc": ["body", "email"],  
            "msg": "invalid email format",  
            "type": "value_error.email"  
        },  
        {  
            "loc": ["body", "password"],  
            "msg": "field required",  
            "type": "value_error.missing"  
        }  
    ]  
}
```

### Error 400 - Regla de Negocio Violada

```
{  
    "detail": "A user with this email already exists"  
}
```

### Error 500 - Error del Servidor

```
{  
    "detail": "Error creating proyecto: Unexpected database error"  
}
```

}

---

## Cómo Interpretar Errores 422 (Validación)

Los errores de validación retornan un array de objetos con:

- **loc**: Ubicación del error (ej: ["body", "email"] significa error en el campo email del body)
- **msg**: Mensaje descriptivo del error
- **type**: Tipo de error Pydantic

**Ejemplo completo:**

```
{  
    "detail": [  
        {  
            "loc": ["body", "titulo"],  
            "msg": "ensure this value has at least 5 characters",  
            "type": "value_error.any_str.min_length"  
        }  
    ]  
}
```

**Solución:** El campo `titulo` debe tener al menos 5 caracteres.

---

## Instrucciones para Pruebas

### Quick Start - La Forma Más Rápida

¡Sin instalación, solo 3 pasos!

1. Abre en tu navegador:

<https://project-planning-cloud-api.onrender.com/docs>

2. Haz click en "POST /api/v1/auth/register" → "Try it out"

3. Completa los datos de prueba:

```
{  
    "email": "profesor@test.com",  
    "password": "Test1234",  
    "nombre": "Profesor",  
    "apellido": "Corrector",  
    "ong": "Universidad"  
}
```

4. Click "Execute"

¡Listo! Ya tienes un usuario. Ahora puedes probar cualquier endpoint directamente desde Swagger.

---

### Opción 1: Swagger UI (Recomendado)

Esta es la forma más fácil y no requiere instalar nada.

1. Abre en tu navegador:

<https://project-planning-cloud-api.onrender.com/docs>

2. Autentízate:

- Busca "POST /api/v1/auth/login"
- Click "Try it out"
- Usa los datos del paso anterior para login
- Click "Execute"
- Copia el `access_token`
- Haz click en el botón **"Authorize"** (arriba a la derecha)
- Pega: `Bearer {tu_access_token}`
- Click "Authorize"

3. Prueba cualquier endpoint:

- Todos los endpoints están listados
- Click "Try it out" en cualquiera
- Completa los parámetros/body
- Click "Execute"

---

### Opción 2: Postman

Si prefieres usar Postman:

1. Descarga: <https://www.postman.com/downloads/>
2. Crea nueva petición POST
3. URL: <https://project-planning-cloud-api.onrender.com/api/v1/auth/register>
4. Body (JSON): usa el ejemplo de arriba
5. Click "Send"
6. Para endpoints protegidos:
  - Tab "Authorization" → Type: "Bearer Token"
  - Token: pega tu `access_token`

---

### Opción 3: cURL (Terminal)

```
# 1. Registrar  
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/register \  
-H "Content-Type: application/json" \  
-d '{"email":"profesor@test.com","password":"Test1234","nombre":"Profesor","apellido":"Corrector","ong":"Universidad"}'
```

```
# 2. Login
curl -X POST https://project-planning-cloud-api.onrender.com/api/v1/auth/login \
-H "Content-Type: application/json" \
-d '{"email":"profesor@test.com", "password":"Test1234"}'
```

---

## Checklist de Pruebas

Use este checklist para validar que todos los endpoints funcionan correctamente:

### Autenticación

- POST /api/v1/auth/register - Crear nuevo usuario
- POST /api/v1/auth/login - Obtener tokens
- POST /api/v1/auth/refresh - Refrescar token

### Proyectos

- POST /api/v1/projects - Crear proyecto con etapas y pedidos anidados
- GET /api/v1/projects/{project\_id} - Obtener proyecto completo
- PATCH /api/v1/projects/{project\_id} - Actualizar proyecto
- DELETE /api/v1/projects/{project\_id} - Eliminar proyecto
- GET /api/v1/projects/{project\_id}/pedidos - Listar pedidos de proyecto

### Pedidos

- POST /api/v1/projects/{project\_id}/etapas/{etapa\_id}/pedidos - Crear pedido
- GET /api/v1/projects/{project\_id}/pedidos?estado=PENDIENTE - Filtrar pendientes
- DELETE /api/v1/pedidos/{pedido\_id} - Eliminar pedido

### Ofertas

- POST /api/v1/pedidos/{pedido\_id}/ofertas - Crear oferta
- GET /api/v1/pedidos/{pedido\_id}/ofertas - Listar ofertas para pedido
- POST /api/v1/ofertas/{oferta\_id}/accept - Aceptar oferta
- POST /api/v1/ofertas/{oferta\_id}/reject - Rechazar oferta
- POST /api/v1/ofertas/{oferta\_id}/confirmar-realizacion - Confirmar realización
- GET /api/v1/ofertas/mis-compromisos - Ver mis compromisos

---

## Sopporte

- **Swagger Interactivo:** <https://project-planning-cloud-api.onrender.com/docs>
- **Health Check:** <https://project-planning-cloud-api.onrender.com/health>
- **OpenAPI JSON:** <https://project-planning-cloud-api.onrender.com/openapi.json>

---

Fin de Documentación