

API Documentation

Contents

1	ProjectPlanning Cloud Persistence API - Documentación Completa	2
1.1	Tabla de Contenidos	2
1.2	Información del Grupo	3
1.2.1	Integrantes	3
1.2.2	Universidad y Cátedra	3
1.3	URLs de Servicios	3
1.3.1	API Cloud Persistence (Producción)	3
1.3.2	Documentación Interactiva	3
1.4	Introducción	4
1.4.1	¿Qué es esta API?	4
1.4.2	Características Principales	4
1.4.3	Stack Tecnológico	4
1.5	Autenticación	4
1.5.1	Flujo de Autenticación JWT	4
1.5.2	Headers Requeridos	5
1.5.3	Información de Tokens	5
1.6	Endpoints de Autenticación	5
1.6.1	1␣ Registrar Nuevo Usuario	5
1.6.2	2␣ Login de Usuario	7
1.6.3	3␣ Refrescar Token	9
1.7	Endpoints de Proyectos	9
1.7.1	1␣ Crear Proyecto	9
1.7.2	2␣ Obtener Proyecto por ID	15
1.7.3	3␣ Actualizar Proyecto (Parcial)	15
1.7.4	4␣ Eliminar Proyecto	17
1.8	Endpoints de Pedidos	18
1.8.1	1␣ Crear Pedido para Etapa Existente	18
1.8.2	2␣ Listar Pedidos de Proyecto	20
1.8.3	3␣ Eliminar Pedido	22
1.9	Endpoints de Ofertas	22
1.9.1	1␣ Crear Oferta para Pedido	22
1.9.2	2␣ Listar Ofertas de Pedido	24
1.9.3	3␣ Aceptar Oferta	25
1.9.4	4␣ Rechazar Oferta	27
1.9.5	5␣ Confirmar Realización de Oferta	28
1.9.6	6␣ Obtener Mis Compromisos	29
1.10	Enumeraciones	30

1.10.1 EstadoProyecto	30
1.10.2 TipoPedido	31
1.10.3 EstadoPedido	31
1.10.4 EstadoOferta	32
1.10.5 UserRole	32
1.11 Flujo Completo de Ejemplo	32
1.11.1 Paso 1: Registrar Dos Usuarios	32
1.11.2 Paso 2: Login con Ambos Usuarios	33
1.11.3 Paso 3: María Crea un Proyecto	34
1.11.4 Paso 4: Carlos Crea una Oferta	34
1.11.5 Paso 5: María ve las Ofertas del Pedido	35
1.11.6 Paso 6: María Acepta la Oferta de Carlos	35
1.11.7 Paso 7: Carlos Confirma que Realizó el Trabajo	35
1.11.8 Paso 8: Carlos Visualiza sus Compromisos	36
1.12 Códigos de Error	36
1.12.1 Códigos HTTP Comunes	36
1.12.2 Ejemplos de Errores	36
1.13 Instrucciones para Pruebas	37
1.13.1 Quick Start - La Forma Más Rápida	37
1.13.2 Opción 1: Swagger UI (Recomendado)	38
1.13.3 Opción 2: Postman	38
1.13.4 Opción 3: cURL (Terminal)	38
1.14 Checklist de Pruebas	39
1.14.1 Autenticación	39
1.14.2 Proyectos	39
1.14.3 Pedidos	39
1.14.4 Ofertas	39
1.15 Soporte	39

1 ProjectPlanning Cloud Persistence API - Documentación Completa

Versión: 1.0.0 **Fecha de Entrega:** Octubre 2025

1.1 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

1.2 Información del Grupo

1.2.1 Integrantes

Nombre	Rol	Email
Mateo Spinetti	[Desarrollador]	[ma-teospinetti1@gmail.com]
Luciano Equiel Wagner	[Desarrollador]	[lucianowagner2003@gmail.com]
Matias Santiago Ramos Giacosa	[Desarrollador]	[matiasramosgi@gmail.com]

1.2.2 Universidad y Cátedra

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

1.3 URLs de Servicios

1.3.1 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>

1.3.2 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
-

1.4 Introducción

1.4.1 ¿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

1.4.2 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

1.4.3 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
-

1.5 Autenticación

1.5.1 Flujo de Autenticación JWT

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

1.5.1.1 Paso 1: Registro

POST /api/v1/auth/register

→ Recibe: email, password, nombre, apellido, ong, role

→ Devuelve: usuario creado

1.5.1.2 Paso 2: Login

POST /api/v1/auth/login

→ Recibe: email, password

→ Devuelve: access_token (15 min) + refresh_token (24h)

1.5.1.3 Paso 3: Usar Token Incluye el access_token en todas las peticiones protegidas:

Authorization: Bearer {access_token}

1.5.1.4 Paso 4: Refrescar Token (cuando expira)

POST /api/v1/auth/refresh

→ Recibe: refresh_token

→ Devuelve: nuevo access_token + refresh_token

1.5.2 Headers Requeridos

Para endpoints protegidos, incluye:

Authorization: Bearer eyJ0eXAI0iJKV1QiLCJhbGciOiJIUzI1NiJ9 ...

Content-Type: application/json

1.5.3 Información de Tokens

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

1.6 Endpoints de Autenticación

1.6.1 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

1.6.1.1 Parámetros

Campo	Tipo	Re-querido	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	-
organization	string	Sí	Nombre de organización	-
role	enum	No	Rol del usuario	MEMBER (default) o COUNCIL

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

1.6.1.2 Body de Prueba

```
{
  "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"
}
```

1.6.1.3 Response Exitoso (201)

1.6.1.4 Errores Posibles

Código	Descripción	Solución
400	Email ya registrado	Usa un email diferente
422	Validación fallida	Revisa el formato de los datos
500	Error del servidor	Contacta al administrador

1.6.1.5 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/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"

1.6.2 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

1.6.2.1 Parámetros

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

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

1.6.2.2 Body de Prueba

```
{
  "access_token":
    ↪ "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtY"
    ↪ ,
  "refresh_token":
    ↪ "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtY"
    ↪ ,
}
```

```
  "token_type": "bearer"  
}
```

1.6.2.3 Response Exitoso (200)

1.6.2.4 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)

1.6.2.5 Errores Posibles

Código	Descripción	Solución
401	Credenciales inválidas	Verifica email y contraseña
422	Validación fallida	Revisa el formato
500	Error del servidor	Contacta al administrador

1.6.2.6 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
```


1.6.3 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

1.6.3.1 Parámetros

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

```
{
  "refresh_token":
    ↪ "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtY"
}
```

1.6.3.2 Body de Prueba

```
{
  "access_token":
    ↪ "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtY"
    ↪ ,
  "refresh_token":
    ↪ "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI1NTBlODQwMC1lMjliLTQxZDQtY"
    ↪ ,
  "token_type": "bearer"
}
```

1.6.3.3 Response Exitoso (200)

1.6.3.4 Errores Posibles

Código	Descripción	Solución
401	Token inválido o expirado	Haz login nuevamente
422	Validación fallida	Verifica el token
500	Error del servidor	Contacta al administrador

1.7 Endpoints de Proyectos

1.7.1 1 Crear 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

1.7.1.1 Parámetros

Campo	Tipo	Re-querido	Descripción	
titulo	string	Sí	Título del proyecto	Mínimo 5 caracteres
descripcion	string	Sí	Descripción detallada	Mínimo 20 caracteres
tipo	string	Sí	Tipo de proyecto	Ej: "Infraestructura", "Social"
pais	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	string	No	ID instancia Bonita	Opcional
etapas	array	No	Array de etapas anidadas	Ver estructura abajo

```
{
  "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 economico",
      "moneda": "string - Opcional, código de moneda",
      "cantidad": "integer - Opcional, para otros tipos",
      "unidad": "string - Opcional, unidad de medida"
    }
  ]
}
```

1.7.1.2 Estructura de Etapas

```
{
  "titulo": "Centro Comunitario La Plata",
  "descripcion": "Construcción de un nuevo centro comunitario con
  ↳ servicios sociales, biblioteca y áreas de educación",
  "etapas": [
    {
      "nombre": "Estructura de Etapas",
      "descripcion": "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 economico",
          "moneda": "string - Opcional, código de moneda",
          "cantidad": "integer - Opcional, para otros tipos",
          "unidad": "string - Opcional, unidad de medida"
        }
      ]
    }
  ]
}
```

```
"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"
      }
    ]
  }
}

```

1.7.1.3 Body de Prueba (Ejemplo Completo)

```

{
  "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_planificacion",
  "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": "economico",
          "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"
    }
  ]
}
]
}

```

1.7.1.4 Response Exitoso (201)

1.7.1.5 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/projects"
3. Click "Try it out"
4. 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_aqui"

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_planificacion",
  "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": 50000.00,
          "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"

1.7.2 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

1.7.2.1 Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a obtener

1.7.2.2 Ejemplo de Ruta

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

1.7.2.3 Response Exitoso (200) Retorna la estructura completa del proyecto igual que en el POST (ver ejemplo anterior).

1.7.2.4 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_aqui"

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"

1.7.3 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

1.7.3.1 Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a actualizar

1.7.3.2 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

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

1.7.3.3 Body de Prueba

1.7.3.4 Response Exitoso (200) Retorna el proyecto actualizado con todos sus datos.

1.7.3.5 Errores Posibles

Código	Descripción	Solución
403	No eres el propietario	Solo el dueño del proyecto puede actualizar
404	Proyecto no existe	Verifica el project_id
422	Validación fallida	Revisa el formato de los datos

1.7.3.6 Instrucciones para Probar Opción 1: Swagger UI (Recomendado)

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

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"
```

```
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"
  }'
```

1.7.4 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

1.7.4.1 Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto a eliminar

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

1.7.4.3 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

1.8 Endpoints de Pedidos

1.8.1 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

1.8.1.1 Path Parameters

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

1.8.1.2 Parámetros

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

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

1.8.1.3 Body de Prueba Ejemplo alternativo (materiales):

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

{
  "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
}
```

1.8.1.4 Response Exitoso (201)

1.8.1.5 Errores Posibles

Código	Descripción	Solución
403	No eres el propietario	Solo el dueño puede crear pedidos
404	Proyecto o etapa no existen	Verifica los IDs
422	Validación fallida	Revisa el tipo y descripción

1.8.1.6 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"
}'
```

1.8.2 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

1.8.2.1 Path Parameters

Parámetro	Tipo	Descripción
project_id	UUID	ID del proyecto

1.8.2.2 Query Parameters (Opcionales)

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

1.8.2.3 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

```
[
  {
    "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"  
  }  
]
```

1.8.2.4 Response Exitoso (200)

1.8.2.5 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"
```

1.8.3 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

1.8.3.1 Path Parameters

Parámetro	Tipo	Descripción
pedido_id	UUID	ID del pedido a eliminar

1.8.3.2 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"

1.9 Endpoints de Ofertas

1.9.1 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

1.9.1.1 Path Parameters

Parámetro	Tipo	Descripción
pedido_id	UUID	ID del pedido a ofertar

1.9.1.2 Parámetros

Campo	Tipo	Re-querido	Descripción	
descripcion	string	Sí	Descripción de la oferta	Mínimo 10 caracteres
monto_ofrecido	float	No	Monto ofrecido (opcional)	Para comparar con presupuesto

```
{
  "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
}
```

1.9.1.3 Body de Prueba

```
{
  "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"
}
```

1.9.1.4 Response Exitoso (201)

1.9.1.5 Errores Posibles

Código	Descripción	Solución
400	Pedido no está en PENDIENTE	Solo se puede ofertar a pedidos pendientes
404	Pedido no existe	Verifica el pedido_id
422	Validación fallida	La descripción debe tener mínimo 10 caracteres

1.9.1.6 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

}'

1.9.2 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

1.9.2.1 Path Parameters

Parámetro	Tipo	Descripción
pedido_id	UUID	ID del pedido

[

{

"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"
    }
  }
]

```

1.9.2.2 Response Exitoso (200)

1.9.2.3 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"

```

1.9.3 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

1.9.3.1 Path Parameters

Parámetro	Tipo	Descripción
oferta_id	UUID	ID de la oferta a aceptar

1.9.3.2 Body Sin body requerido.

```
{
  "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"
}
```

1.9.3.3 Response Exitoso (200)

1.9.3.4 Errores Posibles

Código	Descripción	Solución
403	No eres el propietario	Solo el dueño del proyecto
404	Oferta no existe	Verifica el oferta_id
400	Oferta no está pendiente	Solo ofertas pendientes se pueden aceptar

1.9.3.5 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"

1.9.4 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

1.9.4.1 Path Parameters

Parámetro	Tipo	Descripción
oferta_id	UUID	ID de la oferta a rechazar

```
{
  "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"
}
```

1.9.4.2 Response Exitoso (200)

1.9.4.3 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"
```

1.9.5 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

1.9.5.1 Path Parameters

Parámetro	Tipo	Descripción
oferta_id	UUID	ID de la oferta

1.9.5.2 Body Sin body requerido.

```
{
  "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"
}
```

1.9.5.3 Response Exitoso (200)

1.9.5.4 Errores Posibles

Código	Descripción	Solución
403	No eres el creador	Solo quien creó la oferta puede confirmar
404	Oferta no existe	Verifica el oferta_id
400	Oferta no aceptada	Solo ofertas aceptadas se pueden confirmar
400	Pedido no comprometido	El pedido debe estar en estado COMPROMETIDO

1.9.5.5 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"
```

1.9.6 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

1.9.6.1 Query Parameters (Opcionales)

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

1.9.6.2 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
```

```
[
  {
    "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",
      "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
  }
]

```

1.9.6.3 Response Exitoso (200)

1.9.6.4 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
↪ \
  -H "Authorization: Bearer $TOKEN"

```

1.10 Enumeraciones

1.10.1 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"  
}
```

1.10.2 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"  
}
```

1.10.3 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

1.10.4 EstadoOferta

Estados disponibles para las ofertas:

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

1.10.5 UserRole

Roles disponibles para usuarios:

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

1.11 Flujo Completo de Ejemplo

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

1.11.1 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 retornado. 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 retornado. Ejemplo: 550e8400-e29b-41d4-a716-446655440001

1.11.2 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 ... "

1.11.3 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)

1.11.4 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

1.11.5 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.

1.11.6 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

1.11.7 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

1.11.8 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.

1.12 Códigos de Error

1.12.1 Códigos HTTP Comunes

Código	Nombre	Descripción	Solución
200	OK	Petición exitosa	-
201	Created	Recurso creado exitosamente	-
204	No Content	Eliminación exitosa	-
400	Bad Request	Solicitud incorrecta	Revisa el body y parámetros
401	Unauthorized	Sin autenticación o token inválido	Proporciona un access_token válido
403	Forbidden	No tienes permiso	Solo propietarios pueden hacer esto
404	Not Found	Recurso no existe	Verifica los IDs proporcionados
422	Unprocessable Entity	Validación fallida	Revisa el formato de los datos
500	Internal Server Error	Error del servidor	Contacta al administrador

1.12.2 Ejemplos de Errores

Error 401 - Token Inválido

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

Error 403 - No Autorizado

```
{
  "detail": "Not authorized"
}
```

Error 422 - Validación

```
{
  "detail": [
    {
      "loc": ["body", "email"],
      "msg": "invalid email format",
      "type": "value_error"
    }
  ]
}
```

1.13 Instrucciones para Pruebas

1.13.1 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.

1.13.2 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. **Autentzate:**

- 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"
-

1.13.3 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
-

1.13.4 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","apel"

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"}'
```

1.14 Checklist de Pruebas

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

1.14.1 Autenticación

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

1.14.2 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

1.14.3 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

1.14.4 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
-

1.15 Soporte

- **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>

Para más detalles técnicos: revisar CLAUDE.md

Fin de Documentación