

Manual de Instalación - CondoPlus

Tabla de Contenidos

- [Introducción](#)
- [Requisitos del Sistema](#)
- [Instalación del Proyecto](#)
- [Configuración de Supabase](#)
- [Configuración de Variables de Entorno](#)
- [Ejecución en Desarrollo](#)
- [Build y Despliegue en Producción](#)
- [Solución de Problemas Comunes](#)

Introducción

CondoPlus es una plataforma SaaS diseñada para la gestión integral de condominios en Chile. Proporciona herramientas para administradores y residentes, facilitando la comunicación, gestión financiera, reserva de espacios comunes, y más.

Arquitectura del Sistema

El proyecto está organizado en las siguientes carpetas principales:

- client/**: Frontend desarrollado con React + Vite + Tailwind CSS
- server/**: Backend opcional con Node.js y Express
- shared/**: Código compartido entre frontend y backend
- docs/**: Documentación del proyecto

Requisitos del Sistema

Software Necesario

- Node.js** (versión 18 o superior)
 - Descargar desde: <https://nodejs.org/>
 - Verificar instalación: `node --version`
- npm** (incluido con Node.js) o **yarn**
 - Verificar instalación: `npm --version`
- Git** (para clonar el repositorio)
 - Descargar desde: <https://git-scm.com/>
 - Verificar instalación: `git --version`
- Navegador Web Moderno**
 - Google Chrome (recomendado)
 - Firefox
 - Microsoft Edge
 - Safari

Requisitos de Hardware

- Procesador**: Dual-core 2.0 GHz o superior
- RAM**: 4 GB mínimo (8 GB recomendado)
- Espacio en disco**: 500 MB para el proyecto y dependencias
- Conexión a Internet**: Requerida para Supabase y dependencias

Instalación del Proyecto

1. Clonar el Repositorio

Abra una terminal o línea de comandos y ejecute:

```
git clone https://github.com/tu-usuario/condoplus-app.git  
cd condoplus-app
```

2. Instalar Dependencias del Cliente

El cliente (frontend) es la parte principal de la aplicación:

```
cd client  
npm install
```

Este proceso descargará e instalará todas las dependencias necesarias, incluyendo:

- React 19.2.0
- React Router DOM 7.9.6
- Tailwind CSS 4.1.17
- Supabase Client 2.84.0
- Y otras bibliotecas necesarias

Tiempo estimado: 2-5 minutos dependiendo de la velocidad de su conexión a Internet.

3. Instalar Dependencias del Servidor (Opcional)

Si planea utilizar el backend opcional:

```
cd ../server  
npm install
```

Este proceso instalará:

- Express 4.18.2
- CORS 2.8.5
- Dotenv 16.3.1

Configuración de Supabase

CondoPlus utiliza **Supabase** como backend-as-a-service para autenticación, base de datos y almacenamiento.

1. Crear una Cuenta en Supabase

1. Visite <https://supabase.com>
2. Haga clic en "Start your project"
3. Regístrese con su correo electrónico o cuenta de GitHub

2. Crear un Nuevo Proyecto

1. En el dashboard de Supabase, haga clic en "New Project"
2. Complete los siguientes campos:
 - **Name:** CondoPlus (o el nombre que prefiera)
 - **Database Password:** Elija una contraseña segura y guárdenla en un lugar seguro
 - **Region:** Seleccione la región más cercana a sus usuarios (ej: South America para Chile)
 - **Pricing Plan:** Free (para desarrollo) o Pro (para producción)
3. Haga clic en "Create new project"
4. Espere 1-2 minutos mientras Supabase configura su proyecto

3. Configurar la Base de Datos

Una vez creado el proyecto, debe ejecutar los scripts SQL para crear la estructura de la base de datos:

1. En el panel izquierdo de Supabase, haga clic en el ícono SQL (Editor SQL)
2. Ejecute los siguientes scripts en orden (ubicados en la raíz del proyecto):

a. Crear esquema de usuarios:

```
-- Ejecutar: extend_users_schema.sql  
-- Ejecutar: add_rut_to_users.sql  
-- Ejecutar: allow_admin_update_users.sql
```

b. Crear esquema de incidencias:

```
-- Ejecutar: create_incidents_schema.sql  
-- Ejecutar: fix_incidents_constraints.sql
```

c. Crear esquema de pagos:

```
-- Ejecutar: fix_payments_final.sql
```

d. Configurar políticas de seguridad:

```
-- Ejecutar: fix_rls_final.sql
```

3. Para cada archivo, copie su contenido y péguelo en el editor SQL, luego haga clic en "Run"

4. Obtener Credenciales del Proyecto

1. En el panel izquierdo, haga clic en el ícono de configuración (⚙)
2. Seleccione "API" en el menú
3. Copie los siguientes valores:

- **Project URL:** URL de su proyecto Supabase
- **anon public:** Clave pública anónima

¡IMPORTANTE! Guarde estas credenciales en un lugar seguro. Las necesitará en el siguiente paso.

Configuración de Variables de Entorno

1. Crear Archivo de Configuración

En la carpeta `client/`, cree un archivo llamado `.env` (si no existe):

En Windows:

```
cd client  
copy .env.example .env
```

En macOS/Linux:

```
cd client  
cp .env.example .env
```

2. Configurar Variables de Entorno

Abra el archivo `.env` con un editor de texto y configure las siguientes variables:

```
VITE_SUPABASE_URL=https://tu-proyecto.supabase.co  
VITE_SUPABASE_ANON_KEY=tu-clave-anonima-aqui
```

Reemplace:

- `https://tu-proyecto.supabase.co` con la **Project URL** que copió de Supabase
- `tu-clave-anonima-aqui` con la **anon public** key que copió de Supabase

3. Verificar Configuración

Asegúrese de que:

- El archivo `.env` esté en la carpeta `client/`
- Las variables comienzan con `VITE_` (importante para Vite)
- No haya espacios alrededor del signo =
- El archivo `.env` NO esté incluido en el control de versiones (debe estar en `.gitignore`)

Ejecución en Desarrollo

Iniciar el Cliente (Frontend)

1. Abra una terminal en la carpeta `client/`:

```
cd client
```

2. Ejecute el servidor de desarrollo:

```
npm run dev
```

3. Verá un mensaje similar a:

```
VITE v7.2.4 ready in 500 ms  
→ Local: http://localhost:5173/  
→ Network: use --host to expose
```

4. Abra su navegador y visite: <http://localhost:5173/>

Iniciar el Servidor (Opcional)

Si está utilizando el backend opcional:

1. Abra una **nueva** terminal en la carpeta `server/`:

```
cd server
```

2. Ejecute el servidor:

```
npm run dev
```

3. El servidor estará disponible en: <http://localhost:3000/>

Verificar Funcionamiento

1. La página de login debería cargarse correctamente
2. No debería haber errores en la consola del navegador (F12 → Console)
3. Debería poder ver el formulario de inicio de sesión

Build y Despliegue en Producción

1. Compilar el Cliente

Desde la carpeta `client/`:

```
npm run build
```

Este comando:

- Compila y optimiza todo el código React
- Genera archivos estáticos en la carpeta `client/dist/`
- Minimiza CSS y JavaScript
- Optimiza imágenes y recursos

Tiempo estimado: 30-60 segundos

2. Vista Previa Local

Para probar la versión de producción localmente:

```
npm run preview
```

Esto iniciará un servidor local con la versión compilada en: <http://localhost:4173/>

3. Despliegue en Plataformas Cloud

Opción A: Vercel (Recomendado)

1. Instale Vercel CLI:

```
npm install -g vercel
```

2. Desde la carpeta `client/`, ejecute:

```
vercel
```

3. Siga las instrucciones en pantalla

4. Configure las variables de entorno en el dashboard de Vercel

Opción B: Netlify

1. Instale Netlify CLI:

```
npm install -g netlify-cli
```

2. Desde la carpeta client/, ejecute:

```
netlify init
```

3. Configure:

- o **Build command:** npm run build
- o **Publish directory:** dist

4. Configure las variables de entorno en el dashboard de Netlify

Opción C: Manual

1. Suba el contenido de la carpeta client/dist/ a su servidor web
2. Configure el servidor para redirigir todas las rutas a index.html (para React Router)

Solución de Problemas Comunes

Error: "Missing Supabase environment variables"

Causa: Las variables de entorno no están configuradas correctamente.

Solución:

1. Verifique que el archivo .env exista en client/
2. Asegúrese de que las variables comienzan con VITE_
3. Reinicie el servidor de desarrollo después de modificar .env

Error: "Cannot find module 'xyz'"

Causa: Las dependencias no están instaladas correctamente.

Solución:

```
cd client
rm -rf node_modules package-lock.json
npm install
```

Error: "Port 5173 is already in use"

Causa: El puerto está siendo utilizado por otra aplicación.

Solución:

1. Cierre la aplicación que está usando el puerto
2. O configure un puerto diferente en vite.config.js:

```
export default {
  server: {
    port: 3000
  }
}
```

La aplicación no se conecta a Supabase

Causa: Credenciales incorrectas o proyecto Supabase no configurado.

Solución:

1. Verifique las credenciales en .env
2. Asegúrese de que el proyecto Supabase esté activo
3. Verifique que las tablas de base de datos estén creadas
4. Revise la consola del navegador (F12) para ver errores específicos

Error de CORS en producción

Causa: Configuración incorrecta en Supabase.

Solución:

1. En Supabase Dashboard → Authentication → URL Configuration
2. Agregue su dominio de producción a "Site URL" y "Redirect URLs"

Página en blanco después del build

Causa: Rutas incorrectas en producción.

Solución:

1. Verifique que su servidor esté configurado para SPA (Single Page Application)
2. Todas las rutas deben redirigir a `index.html`
3. En Netlify, cree un archivo `_redirects`:

```
/*    /index.html    200
```

Error: "Failed to resolve import"

Causa: Importaciones con rutas incorrectas.

Solución:

1. Verifique que las rutas de importación sean correctas
2. Use rutas relativas o alias configurados en `vite.config.js`
3. Limpie la caché: `npm run dev -- --force`

Soporte y Recursos Adicionales

Documentación de Tecnologías

- **React:** <https://react.dev/>
- **Vite:** <https://vitejs.dev/>
- **Tailwind CSS:** <https://tailwindcss.com/>
- **Supabase:** <https://supabase.com/docs>
- **React Router:** <https://reactrouter.com/>

Contacto

Para soporte técnico o reportar problemas, contacte al equipo de desarrollo de CondoPlus.

Versión del Manual: 1.0

Fecha: Diciembre 2025

Última Actualización: 01/12/2025