



LA RESERVA - DESARROLLO WEB

PARTE B: Instalación y Configuración

Versión: 1.0

Fecha: Octubre 2025

Tiempo estimado: 30-45 minutos



CONTENIDO

1. Verificación de Requisitos Previos
 2. Instalación de pnpm
 3. Creación del Proyecto Astro
 4. Instalación de Dependencias
 5. Configuración de Astro
 6. Configuración de Tailwind CSS
 7. Configuración de TypeScript
 8. Variables de Entorno
 9. Configuración de Git y Prettier
 10. Verificación Final
-

1. VERIFICACIÓN DE REQUISITOS PREVIOS

Antes de comenzar, verifica que tienes todo listo:



Checklist de Software

```
# Verificar Node.js (debe ser 18.14.1 o superior)
node --version
# Salida esperada: v18.x.x o v20.x.x
```

```
# Verificar Git
git --version
# Salida esperada: git version 2.x.x
```

```
# Verificar que NO tengas Astro instalado globalmente (no es necesario)
# Solo verificación, no hay problema si no existe
```

✓ Checklist de Cuentas

- ☐ Cuenta de GitHub creada
 - ☐ Cuenta de Supabase creada (free tier)
 - ☐ Cuenta de Vercel creada
 - ☐ Cuenta de Resend creada (opcional por ahora)
-

2. INSTALACIÓN DE pnpm

pnpm es más rápido y eficiente que npm. Lo usaremos en todo el proyecto.

```
# Instalar pnpm globalmente
npm install -g pnpm

# Verificar instalación
pnpm --version
# Salida esperada: 8.x.x o superior

# (Opcional) Configurar pnpm para usar shamefully-hoist si tienes problemas
pnpm config set shamefully-hoist true
```

¿Por qué pnpm?

- ⚡ 3x más rápido que npm
 - 💾 Ahorra espacio en disco (80% menos)
 - 🔒 Más seguro por defecto
 - 🚀 Perfecto para monorepos
-

3. CREACIÓN DEL PROYECTO ASTRO

Paso 3.1: Crear el proyecto

```
# Navega a la carpeta donde quieres crear el proyecto
cd ~/proyectos # O donde prefieras
```

```
# Crear proyecto con pnpm
pnpm create astro@latest la-reserva

# Durante la instalación, selecciona:
# └─ astro    Launch sequence initiated.
# |
# ◆ How would you like to start your new project?
# |   ○ Use blog template
# |   ○ Use documentation template
# |   ● Empty ← SELECCIONA ESTA
# |
# ◆ Install dependencies?
# |   ● Yes (recommended) ← SELECCIONA ESTA
# |
# ◆ Do you plan to write TypeScript?
# |   ● Yes ← SELECCIONA ESTA
# |
# ◆ How strict should TypeScript be?
# |   ○ Relaxed
# |   ● Strict ← SELECCIONA ESTA
# |
# ◆ Initialize a new git repository?
# |   ● Yes ← SELECCIONA ESTA
```

Paso 3.2: Entrar al proyecto

```
cd la-reserva

# Verificar estructura inicial
ls -la

# Deberías ver: astro.config.mjs, package.json, tsconfig.json, src/, publ
```

4. INSTALACIÓN DE DEPENDENCIAS

Instalaremos todas las librerías necesarias para el proyecto.

4.1 Dependencias de Producción

```
# React y tipos
pnpm add react react-dom
pnpm add -D @types/react @types/react-dom

# Astro integrations
pnpm add @astrojs/react @astrojs/tailwind

# Tailwind CSS y plugins
pnpm add tailwindcss @tailwindcss/typography @tailwindcss/forms
pnpm add -D prettier prettier-plugin-astro prettier-plugin-tailwindcss

# Supabase
pnpm add @supabase/supabase-js

# Forms y validación
pnpm add react-hook-form zod @hookform/resolvers

# UI Components
pnpm add @radix-ui/react-dialog @radix-ui/react-dropdown-menu
pnpm add @radix-ui/react-select @radix-ui/react-toast
pnpm add @radix-ui/react-tabs @radix-ui/react-label
pnpm add @radix-ui/react-avatar @radix-ui/react-tooltip

# Animaciones
pnpm add framer-motion

# Calendario
pnpm add react-big-calendar
pnpm add date-fns

# Iconos
pnpm add lucide-react

# Utilidades
pnpm add clsx tailwind-merge
pnpm add nanoid

# Emails (Resend)
pnpm add resend

# (Opcional) Maps
pnpm add @vis.gl/react-google-maps
```

4.2 Dependencias de Desarrollo

```
# TypeScript utilities
pnpm add -D typescript

# Astro specific
pnpm add -D astro-compress astro-seo

# PostCSS (necesario para Tailwind)
pnpm add -D postcss autoprefixer
```

4.3 Verificar instalación

```
# Ver el archivo package.json
cat package.json

# Deberías ver todas las dependencias listadas
```

5. CONFIGURACIÓN DE ASTRO

5.1 Configurar integraciones

Edita `astro.config.mjs`:

```
// astro.config.mjs
import { defineConfig } from 'astro/config';
import react from '@astrojs/react';
import tailwind from '@astrojs/tailwind';

export default defineConfig({
  // Integraciones
  integrations: [
    react(),
    tailwind({
      // Aplicar estilos base de Tailwind
      applyBaseStyles: false, // Lo haremos manualmente para más control
    }),
  ],
});
```

```

// Output estático por defecto (SSG)
output: 'static',

// Build configuration
build: {
  inlineStylesheets: 'auto',
},

// Server configuration (para desarrollo)
server: {
  port: 3000,
  host: true, // Permite acceso desde red local
},

// Vite configuration
vite: {
  optimizeDeps: {
    exclude: ['@supabase/supabase-js'],
  },
},

// Experimental features
experimental: {
  contentCollectionCache: true,
},
});

```

5.2 Configurar rutas

Crea el archivo `src/env.d.ts` (si no existe):

```

/// <reference types="astro/client" />

interface ImportMetaEnv {
  readonly SUPABASE_URL: string;
  readonly SUPABASE_ANON_KEY: string;
  readonly PUBLIC_SITE_URL: string;
  readonly RESEND_API_KEY: string;
}

interface ImportMeta {

```

```
readonly env: ImportMetaEnv;  
}
```

6. CONFIGURACIÓN DE TAILWIND CSS

6.1 Generar archivo de configuración

```
# Crear archivo de configuración de Tailwind  
pnpx tailwindcss init
```

6.2 Configurar Tailwind

Edita `tailwind.config.mjs`:

```
/** @type {import('tailwindcss').Config} */  
export default {  
  content: ['./src/**/*.{astro,html,js,jsx,md,mdx,svelte,ts,tsx,vue}'],  
  theme: {  
    extend: {  
      colors: {  
        // Colores de La Reserva  
        primary: {  
          50: '#FEF9E7',  
          100: '#FDF3D0',  
          200: '#FAE7A1',  
          300: '#F7DB72',  
          400: '#F4CF43',  
          500: '#D4AF37', // Dorado principal  
          600: '#B8941F',  
          700: '#8A6F17',  
          800: '#5C4A0F',  
          900: '#2E2508',  
        },  
        secondary: {  
          50: '#F5F5F5',  
          100: '#E5E5E5',  
          200: '#CCCCCC',  
          300: '#B3B3B3',  
          400: '#999999',  
          500: '#2D2D2D',  
        },  
      },  
    },  
  },  
}
```

```

        600: '#1A1A1A', // Negro principal
        700: '#0D0D0D',
        800: '#000000',
        900: '#000000',
    },
    accent: {
        DEFAULT: '#F59E0B',
        50: '#FEF3E2',
        100: '#FDE8C5',
        200: '#FBD18B',
        300: '#F9BA51',
        400: '#F7A317',
        500: '#F59E0B',
        600: '#C47E09',
        700: '#935F07',
        800: '#623F04',
        900: '#312002',
    },
},
fontFamily: {
    display: ['Playfair Display', 'Georgia', 'serif'],
    body: ['Montserrat', 'system-ui', 'sans-serif'],
},
fontSize: {
    'hero': ['4.5rem', { lineHeight: '1.2', letterSpacing: '0.08em' }],
    'display': ['3rem', { lineHeight: '1.2', letterSpacing: '0.06em' }],
},
spacing: {
    '18': '4.5rem',
    '88': '22rem',
    '128': '32rem',
},
animation: {
    'fade-in': 'fadeIn 0.5s ease-in-out',
    'slide-up': 'slideUp 0.5s ease-out',
    'slide-down': 'slideDown 0.5s ease-out',
},
keyframes: {
    fadeIn: {
        '0%': { opacity: '0' },
        '100%': { opacity: '1' },
    },
    slideUp: {
        '0%': { transform: 'translateY(20px)', opacity: '0' },

```



```

    '100%': { transform: 'translateY(0)', opacity: '1' },
  },
  slideDown: {
    '0%': { transform: 'translateY(-20px)', opacity: '0' },
    '100%': { transform: 'translateY(0)', opacity: '1' },
  },
},
},
},
plugins: [
  require('@tailwindcss/typography'),
  require('@tailwindcss/forms'),
],
};

```

6.3 Crear archivo CSS global

Crea `src/styles/global.css` :

```
/* src/styles/global.css */

/* Import Tailwind base, components, utilities */
@tailwind base;
@tailwind components;
@tailwind utilities;

/* Import Google Fonts */
@import url('https://fonts.googleapis.com/css2?family=Playfair+Display:wght@400;500;600;700;800;900&display=swap');

/* Base styles */
@layer base {
  html {
    @apply scroll-smooth;
  }

  body {
    @apply font-body text-secondary-500 bg-white antialiased;
  }

  h1, h2, h3, h4, h5, h6 {
    @apply font-display text-secondary-600;
  }
}
```

```

h1 {
  @apply text-5xl md:text-6xl lg:text-hero font-bold tracking-wider;
}

h2 {
  @apply text-3xl md:text-4xl lg:text-display font-semibold tracking-wi
}

h3 {
  @apply text-2xl md:text-3xl font-bold;
}

h4 {
  @apply text-xl md:text-2xl font-bold;
}

p {
  @apply leading-relaxed;
}

a {
  @apply transition-colors duration-200;
}
}

/* Component utilities */
@layer components {
  .section {
    @apply py-16 md:py-24 lg:py-32;
  }

  .container-custom {
    @apply container mx-auto px-4 md:px-6 lg:px-8 max-w-7xl;
  }

  .btn {
    @apply inline-flex items-center justify-center px-6 py-3 font-semibol
      rounded-full transition-all duration-200 uppercase tracking-wi
      text-sm md:text-base disabled:opacity-50 disabled:cursor-not-a
  }

  .btn-primary {
    @apply btn bg-gradient-to-r from-primary-500 to-accent-500
      text-secondary-600 hover:from-primary-600 hover:to-accent-600

```

```

        shadow-lg hover:shadow-xl;
    }

    .btn-secondary {
        @apply btn bg-transparent border-2 border-primary-500
            text-primary-500 hover:bg-primary-500 hover:text-secondary-600;
    }

    .btn-outline {
        @apply btn bg-transparent border-2 border-white
            text-white hover:bg-white hover:text-secondary-600;
    }

    .card {
        @apply bg-white rounded-lg shadow-md hover:shadow-xl
            transition-shadow duration-300 overflow-hidden;
    }

    .input {
        @apply w-full px-4 py-3 border border-secondary-200
            rounded-lg focus:outline-none focus:ring-2
            focus:ring-primary-500 focus:border-transparent
            transition-all duration-200;
    }

    .label {
        @apply block text-sm font-semibold text-secondary-600 mb-2;
    }

    .hero-section {
        @apply relative min-h-screen flex items-center justify-center
            bg-secondary-600 text-white overflow-hidden;
    }

    .overlay {
        @apply absolute inset-0 bg-black bg-opacity-60;
    }
}

/* Utility classes */
@layer utilities {
    .text-balance {
        text-wrap: balance;
    }
}

```

```

.text-gradient {
  @apply bg-gradient-to-r from-primary-500 to-accent-500
    bg-clip-text text-transparent;
}

.scrollbar-hide {
  -ms-overflow-style: none;
  scrollbar-width: none;
}

.scrollbar-hide::-webkit-scrollbar {
  display: none;
}
}

/* Custom animations */
@keyframes shimmer {
  0% {
    background-position: -1000px 0;
  }
  100% {
    background-position: 1000px 0;
  }
}

.animate-shimmer {
  animation: shimmer 2s infinite linear;
  background: linear-gradient(
    to right,
    transparent 0%,
    rgba(212, 175, 55, 0.3) 50%,
    transparent 100%
  );
  background-size: 1000px 100%;
}

```

7. CONFIGURACIÓN DE TYPESCRIPT

El archivo `tsconfig.json` ya existe, pero lo mejoraremos:

```

{
  "extends": "astro/tsconfigs/strict",
  "compilerOptions": {
    "jsx": "react-jsx",
    "jsxImportSource": "react",
    "baseUrl": ".",
    "paths": {
      "@/*": ["src/*"],
      "@components/*": ["src/components/*"],
      "@layouts/*": ["src/layouts/*"],
      "@lib/*": ["src/lib/*"],
      "@utils/*": ["src/utils/*"],
      "@types/*": ["src/types/*"],
      "@styles/*": ["src/styles/*"]
    },
    "strict": true,
    "skipLibCheck": true,
    "resolveJsonModule": true,
    "allowSyntheticDefaultImports": true,
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true
  }
}

```

8. VARIABLES DE ENTORNO

8.1 Crear archivos de entorno

```

# Crear archivo .env para desarrollo
touch .env

```

```

# Crear archivo de ejemplo para el equipo
touch .env.example

```

8.2 Configurar **.env**

```

# .env (NO SUBIR A GIT)

# Supabase

```

```
SUPABASE_URL=https://tu-proyecto.supabase.co
SUPABASE_ANON_KEY=tu-clave-anonima-aqui
SUPABASE_SERVICE_KEY=tu-clave-de-servicio-aqui

# Site
PUBLIC_SITE_URL=http://localhost:3000

# Resend (Emails)
RESEND_API_KEY=re_tu_api_key_aqui

# WhatsApp Business API (Opcional)
WHATSAPP_BUSINESS_ID=tu-business-id
WHATSAPP_ACCESS_TOKEN=tu-access-token

# Google Maps (Opcional)
PUBLIC_GOOGLE_MAPS_API_KEY=tu-api-key-aqui
```

8.3 Configurar **.env.example**

```
# .env.example (SÍ SUBIR A GIT como plantilla)

# Supabase
SUPABASE_URL=https://tu-proyecto.supabase.co
SUPABASE_ANON_KEY=tu-clave-anonima
SUPABASE_SERVICE_KEY=tu-clave-de-servicio

# Site
PUBLIC_SITE_URL=http://localhost:3000

# Resend (Emails)
RESEND_API_KEY=re_XXXXXXXXXXXXX

# WhatsApp Business API
WHATSAPP_BUSINESS_ID=
WHATSAPP_ACCESS_TOKEN=

# Google Maps
PUBLIC_GOOGLE_MAPS_API_KEY=
```

8.4 Actualizar **.gitignore**

```
# Agregar al .gitignore (si no existe)
echo ".env" >> .gitignore
echo "node_modules/" >> .gitignore
echo "dist/" >> .gitignore
echo ".DS_Store" >> .gitignore
echo ".vercel" >> .gitignore
```

9. CONFIGURACIÓN DE GIT Y PRETTIER

9.1 Configurar Prettier

Crea `.prettierrc`:

```
{
  "semi": true,
  "singleQuote": true,
  "tabWidth": 2,
  "trailingComma": "es5",
  "printWidth": 80,
  "plugins": ["prettier-plugin-astro", "prettier-plugin-tailwindcss"],
  "overrides": [
    {
      "files": "*.astro",
      "options": {
        "parser": "astro"
      }
    }
  ]
}
```

Crea `.prettierignore`:

```
node_modules/
dist/
.astro/
pnpm-lock.yaml
package-lock.json
```

9.2 Configurar Git

```
# Inicializar repositorio (si no se hizo automáticamente)
git init

# Configurar usuario (si no está configurado globalmente)
git config user.name "Tu Nombre"
git config user.email "tu@email.com"

# Agregar todos los archivos
git add .

# Primer commit
git commit -m "🚀 Initial setup: Astro + React + Tailwind + Supabase"

# (Opcional) Conectar con GitHub
# Crea un repositorio en GitHub primero, luego:
git remote add origin https://github.com/tu-usuario/la-reserva.git
git branch -M main
git push -u origin main
```

9.3 Scripts útiles en `package.json`

Agrega estos scripts a tu `package.json` :

```
{
  "scripts": {
    "dev": "astro dev",
    "start": "astro dev",
    "build": "astro build",
    "preview": "astro preview",
    "astro": "astro",
    "format": "prettier --write .",
    "format:check": "prettier --check .",
    "type-check": "tsc --noEmit"
  }
}
```

10. VERIFICACIÓN FINAL

10.1 Probar el servidor de desarrollo


```
# Iniciar servidor
pnpm dev

# Deberías ver:
# 🚀 astro v4.x.x started in XXXms
#
# | Local      http://localhost:3000/
# | Network   http://192.168.x.x:3000/
```

10.2 Abrir en navegador

Abre <http://localhost:3000> en tu navegador. Deberías ver la página de inicio de Astro.

10.3 Verificar configuración

```
# Verificar que TypeScript funciona
pnpm type-check
# No debería haber errores

# Verificar que Prettier funciona
pnpm format:check
# Debería decir "All matched files use Prettier code style!"
```

10.4 Estructura de proyecto final

Tu proyecto debería verse así:

```
la-reserva/
├─ node_modules/
├─ public/
│   └─ favicon.svg
├─ src/
│   ├── env.d.ts
│   ├── pages/
│   │   └─ index.astro
│   └─ styles/
│       └─ global.css
├─ .env
├─ .env.example
└─ .gitignore
```

```
|— .prettierrc
|— .prettierignore
|— astro.config.mjs
|— package.json
|— pnpm-lock.yaml
|— tailwind.config.mjs
|— tsconfig.json
```

CHECKLIST DE COMPLETITUD

Antes de continuar al siguiente archivo (02C), verifica:

Software y herramientas:

- ☐ Node.js 18+ instalado y funcionando
- ☐ pnpm instalado globalmente
- ☐ Git configurado correctamente
- ☐ VS Code con extensiones instaladas

Proyecto:

- ☐ Proyecto Astro creado
- ☐ Todas las dependencias instaladas sin errores
- ☐ Archivos de configuración creados y configurados:
 - ☐ `astro.config.mjs`
 - ☐ `tailwind.config.mjs`
 - ☐ `tsconfig.json`
 - ☐ `.prettierrc`
 - ☐ `.env` y `.env.example`
- ☐ Estilos globales creados (`src/styles/global.css`)
- ☐ Git inicializado y primer commit realizado
- ☐ Servidor de desarrollo funciona (`pnpm dev`)
- ☐ No hay errores de TypeScript (`pnpm type-check`)

Cuentas configuradas:

- ☐ Supabase: proyecto creado, credenciales en `.env`
- ☐ Vercel: cuenta creada (configuración en fase de deploy)
- ☐ Resend: cuenta creada, API key en `.env`

- ☐ GitHub: repositorio creado y conectado
-



SOLUCIÓN DE PROBLEMAS COMUNES

Error: "Cannot find module 'astro'"

```
# Solución: Reinstalar dependencias
rm -rf node_modules pnpm-lock.yaml
pnpm install
```

Error: "Port 3000 is already in use"

```
# Solución: Cambiar puerto en astro.config.mjs
server: {
  port: 3001, // o cualquier otro puerto libre
}
```

Error: Tailwind no aplica estilos

```
# Solución: Verificar que global.css esté importado
# En src/pages/index.astro, agrega:
---
import '../styles/global.css';
---
```

Error: Variables de entorno no se leen

```
# Solución: Reiniciar servidor después de cambiar .env
# Ctrl+C para detener
pnpm dev # Iniciar de nuevo
```

Error: TypeScript arroja errores en archivos .astro

```
# Solución: Instalar extensión de Astro en VS Code
# Buscar: "Astro" en el marketplace de extensiones
```

PRÓXIMOS PASOS

Una vez completado este archivo, continúa con:

02C - Estructura y Estilos

- Crear estructura completa de carpetas
- Configurar layouts base
- Crear componentes reutilizables

SOPORTE

Si encuentras problemas:

1. Revisa la sección de Solución de Problemas
2. Consulta la documentación oficial de Astro: <https://docs.astro.build>
3. Pregunta al equipo de desarrollo

© 2025 La Reserva. Documentación técnica del proyecto.