**Guía de Instalación y Configuración del Ambiente de Desarrollo Web**

## 📋 Introducción

Configurar un ambiente de desarrollo web adecuado es el primer paso fundamental para cualquier desarrollador. Esta guía te llevará paso a paso through el proceso de instalación y configuración de todas las herramientas necesarias.

---

## 🛠️ 1. Editores de Código

**Visual Studio Code (Recomendado)**

**Proceso de instalación:**

1. Visitar code.visualstudio.com
2. Descargar la versión para tu sistema operativo
3. Ejecutar el instalador y seguir los pasos
4. Seleccionar "Add to PATH" durante la instalación

**Extensiones esenciales:**

- HTML CSS Support
- JavaScript (ES6) code snippets
- Auto Rename Tag
- Live Server
- Prettier - Code formatter
- Bracket Pair Colorizer
- GitLens
- Material Icon Theme

**Configuración básica:**
En settings.json agregar:
{
"editor.fontSize": 14,
"editor.wordWrap": "on",
"files.autoSave": "afterDelay",

```
"emmet.includeLanguages": {
"javascript": "html"
}
}
```

---

### 🌐 2. Servidores Locales

**Opción A: XAMPP (Para PHP/MySQL)**

**Instalación:**

1. Descargar desde apachefriends.org
2. Ejecutar el instalador como administrador
3. Seleccionar componentes:
    - Apache (obligatorio)
    - MySQL (para bases de datos)
    - PHP (para backend)
    - phpMyAdmin (para gestión de BD)

**Configuración:**

- Puerto Apache: 80 (default) o 8080 si el 80 está ocupado
- Puerto MySQL: 3306
- Directorio raíz: C:\xampp\htdocs\ (Windows) o /Applications/XAMPP/htdocs/ (Mac)

**Verificación:**

1. Abrir Panel de Control de XAMPP
2. Iniciar Apache y MySQL
3. Abrir navegador en http://localhost
4. Debería aparecer la página de bienvenida de XAMPP

**Opción B: Live Server Extension (VS Code)**

**Para desarrollo frontend simple:**

1. Instalar extensión "Live Server" en VS Code
2. Abrir archivo HTML
3. Click derecho → "Open with Live Server"
4. Servidor automático en http://127.0.0.1:5500

---

### 📊 3. Control de Versiones - Git

**Instalación de Git**

**Windows:**

1. Descargar Git desde git-scm.com
2. Ejecutar instalador con configuraciones por defecto
3. Seleccionar "Git from the command line and also from 3rd-party software"

**macOS:**
Opción 1: Con Homebrew - brew install git
Opción 2: Descargar instalador oficial

**Linux (Ubuntu/Debian):**
sudo apt update
sudo apt install git

**Configuración inicial:**

git config --global user.name "Tu Nombre"
git config --global user.email "tu.email@ejemplo.com"
git config --global init.defaultBranch main

**Flujo de trabajo básico:**

**Inicializar repositorio**

git init

**Ver estado de archivos**

git status

**Añadir archivos al staging**

git add .

**Hacer commit de cambios**

git commit -m "Descripción del cambio"

**Conectar con repositorio remoto**

git remote add origin https://github.com/usuario/repositorio.git

**Subir cambios**

git push -u origin main

---

### 🎨 4. Herramientas de Diseño

**Figma (Diseño de Interfaces)**

1. Crear cuenta en figma.com
2. Descargar aplicación desktop (opcional)
3. Familiarizarse con:
    - Frames y artboards
    - Sistema de componentes
    - Auto-layout
    - Prototyping

**Adobe XD**

- Alternativa profesional de Adobe
- Ideal para prototipos interactivos
- Integración con Creative Cloud

---

### 🔧 5. Navegadores para Desarrollo

**Navegadores recomendados:**

- Google Chrome + DevTools
- Mozilla Firefox + Firefox Developer Edition
- Microsoft Edge + DevTools

**Configuración de Chrome DevTools:**

Accesos rápidos:
F12 - Abrir DevTools
Ctrl+Shift+I - Alternar DevTools
Ctrl+Shift+C - Selector de elementos

**Extensiones útiles:**

- Web Developer (barra de herramientas)
- ColorZilla (selector de colores)
- JSON Formatter (formateador de JSON)
- Lighthouse (auditoría de performance)

---

## ⚡ 6. Entornos de Ejecución

**Node.js (Para JavaScript en servidor)**

**Instalación:**

1. Descargar LTS version desde nodejs.org
2. Ejecutar instalador
3. Verificar instalación:
   node --version
   npm --version

**Gestión de paquetes:**

**Inicializar proyecto**

npm init -y

**Instalar dependencias**

npm install nombre-paquete

**Instalar dependencias de desarrollo**

npm install --save-dev nombre-paquete

**Ejecutar scripts**

npm start
npm run build

---

### 🐬 7. Contenedores (Opcional avanzado)

**Docker**

Ejemplo de Dockerfile para proyecto web:

```
FROM node:14
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

**Comandos básicos:**

```
docker --version
docker build -t mi-app .
docker run -p 3000:3000 mi-app
```

---

### 🔍 8. Herramientas de Testing

**Para HTML/CSS:**

- W3C Validator - Validación de código
- CSS Lint - Análisis de CSS
- BrowserStack - Testing multi-navegador

**Para JavaScript:**

- Jest - Framework de testing
- Cypress - Testing end-to-end

---

### 📝 9. Verificación de la Configuración

**Checklist de instalación:**

- Editor de código instalado y configurado
- Servidor local funcionando (XAMPP o similar)
- Git instalado y configurado
- Node.js y npm funcionando
- Navegadores actualizados
- Extensiones esenciales instaladas

**Prueba de funcionamiento:**

Crear archivo test.html:

<!DOCTYPE html><html> <head> <title>Test Ambiente</title> </head> <body> <h1>¡Ambiente configurado correctamente!</h1> <script> console.log("JavaScript funcionando"); </script> </body> </html>

---

### 🚨 10. Solución de Problemas Comunes

**Puerto 80 ocupado:**

En Windows:
netstat -ano | findstr :80
taskkill /PID [PID] /F

Cambiar puerto en XAMPP:
Editar httpd.conf -> Listen 8080

**Problemas con Git:**

Reconfigurar credenciales:
git config --global --unset credential.helper
git config --global credential.helper store

**Node.js no reconocido:**

- Reiniciar terminal/consola
- Verificar variables de entorno PATH
- Reinstalar Node.js

---

## 🎯 Conclusión

Un ambiente de desarrollo bien configurado es esencial para la productividad y el aprendizaje. Esta configuración te permitirá:

- Desarrollar eficientemente con herramientas modernas
- Probar localmente antes de deployar
- Colaborar usando control de versiones
- Debuggear efectivamente con las herramientas adecuadas

**¡Listo para comenzar a desarrollar!** 🚀