GUÍA DE SETUP DEL ENTORNO DE DESARROLLO

IFCD0110 - Confección y Publicación de Páginas Web

📋 ÍNDICE

---

1. INTRODUCCIÓN

¿Qué es un entorno de desarrollo?

Un entorno de desarrollo es el conjunto de herramientas, software y configuraciones que utilizarás para crear páginas web profesionales. Este setup te permitirá:

- Escribir código HTML, CSS y JavaScript eficientemente
- Controlar versiones de tu código
- Previsualizar tus páginas en diferentes navegadores
- Colaborar con otros desarrolladores
- Publicar tus proyectos en internet

Objetivos de esta guía

Al finalizar esta guía, tendrás:

✅ Un editor de código profesional configurado (VS Code)
✅ Sistema de control de versiones (Git y GitHub)

✅ Navegadores para testing cross-browser
✅ Herramientas complementarias instaladas
✅ Una estructura de proyecto organizada

Tiempo estimado de configuración

⏱️ **1-2 horas** (dependiendo de tu conexión a internet y sistema operativo)

---

2. REQUISITOS DEL SISTEMA

Requisitos mínimos

Componente	Windows	macOS	Linux
Sistema Operativo	Windows 10/11	macOS 10.14+	Ubuntu 20.04+ / Debian / Fedora
Procesador	Intel Core i3 o equivalente	Intel Core i3 o Apple Silicon	Intel Core i3 o equivalente
RAM	4 GB	4 GB	4 GB
Espacio en disco	5 GB libres	5 GB libres	5 GB libres
Conexión a internet	Requerida para instalación	Requerida para instalación	Requerida para instalación

Requisitos recomendados

Componente	Especificación
RAM	8 GB o más
Procesador	Intel Core i5 o superior
Espacio en disco	10 GB o más (para proyectos)
Pantalla	1920x1080 (Full HD)

---

3. INSTALACIÓN DE SOFTWARE BÁSICO

3.1 Visual Studio Code (VS Code)

Visual Studio Code es el editor de código que utilizaremos. Es gratuito, potente y ampliamente usado en la industria.

Windows

1. Visita: https://code.visualstudio.com/
2. Descarga el instalador para Windows (`.exe`)
3. Ejecuta el instalador
4. **Importante:** Durante la instalación, marca estas opciones:

- ✅ Agregar "Abrir con Code" al menú contextual de archivos
- ✅ Agregar "Abrir con Code" al menú contextual de directorios
- ✅ Registrar Code como editor predeterminado
- ✅ Agregar a PATH

5. Haz clic en "Instalar" y espera a que finalice
6. Marca "Ejecutar Visual Studio Code" y haz clic en "Finalizar"

macOS

Opción 1: Descarga directa

1. Visita: https://code.visualstudio.com/
2. Descarga VS Code para macOS
3. Abre el archivo `.zip` descargado
4. Arrastra "Visual Studio Code.app" a la carpeta "Aplicaciones"
5. Abre VS Code desde Aplicaciones

Opción 2: Homebrew (si ya lo tienes instalado)

```
brew install --cask visual-studio-code
```

Configurar el comando `code` en terminal:

1. Abre VS Code
2. Presiona `Cmd + Shift + P`
3. Escribe "shell command"
4. Selecciona "Shell Command: Install 'code' command in PATH"

Linux (Ubuntu/Debian)

Opción 1: Desde el sitio oficial

```
# Descarga el paquete .deb
wget -O code.deb https://code.visualstudio.com/sha/download?build=stable&os=linux-deb-x64

# Instala el paquete
sudo apt install ./code.deb

# Limpia el archivo descargado
rm code.deb
```

Opción 2: Repositorio oficial de Microsoft

```
# Instala dependencias
sudo apt update
sudo apt install software-properties-common apt-transport-https wget

# Importa la clave GPG de Microsoft
wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
```

```
# Agrega el repositorio
sudo add-apt-repository "deb [arch=amd64]
https://packages.microsoft.com/repos/vscode stable main"

# Instala VS Code
sudo apt update
sudo apt install code
```

Linux (Fedora/RHEL)

```
# Importa la clave GPG
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc

# Crea el repositorio
sudo sh -c 'echo -e "[code]\nname=Visual Studio
Code\nbaseurl=https://packages.microsoft.com/yumrepos/vscode\nenabled=1\ngpgcheck=
1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" >
/etc/yum.repos.d/vscode.repo'

# Instala VS Code
sudo dnf install code
```

3.2 Git

Git es el sistema de control de versiones más utilizado en el mundo.

Windows

Opción recomendada: Git for Windows

1. Visita: https://git-scm.com/download/win
2. Descarga el instalador (64-bit recomendado)
3. Ejecuta el instalador
4. **Configuración recomendada durante la instalación:**
    - Editor predeterminado: **Use Visual Studio Code as Git's default editor**
    - Ajustar PATH: **Git from the command line and also from 3rd-party software**
    - Ejecutable SSH: **Use bundled OpenSSH**
    - Librería HTTPS: **Use the OpenSSL library**
    - Fin de línea: **Checkout Windows-style, commit Unix-style line endings**
    - Emulador de terminal: **Use MinTTY**
    - Comportamiento de `git pull`: **Default (fast-forward or merge)**
    - Credential helper: **Git Credential Manager**
    - Características extra: Marca **Enable file system caching**
5. Haz clic en "Install"

Verificar instalación:

```
# Abre Git Bash o PowerShell
git --version
```

macOS

Opción 1: Homebrew (recomendada)

```
# Instala Homebrew si no lo tienes
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Instala Git
brew install git
```

Opción 2: Xcode Command Line Tools

```
xcode-select --install
```

Verificar instalación:

```
git --version
```

Linux (Ubuntu/Debian)

```
sudo apt update
sudo apt install git
```

Linux (Fedora)

```
sudo dnf install git
```

Verificar instalación:

```
git --version
```

3.3 Node.js y npm (Opcional pero recomendado)

Node.js te permitirá usar herramientas modernas de desarrollo web como servidores locales, bundlers y task runners.

Windows

1. Visita: https://nodejs.org/
2. Descarga la versión **LTS** (Long Term Support)
3. Ejecuta el instalador `.msi`
4. Acepta los términos y sigue el asistente (configuración predeterminada está bien)
5. Asegúrate de marcar "Automatically install the necessary tools"

Verificar instalación:

```
node --version
npm --version
```

macOS

Opción 1: Descarga directa

1. Visita: https://nodejs.org/
2. Descarga la versión LTS para macOS
3. Ejecuta el instalador `.pkg`

Opción 2: Homebrew

```
brew install node
```

Verificar instalación:

```
node --version
npm --version
```

Linux

Ubuntu/Debian:

```
# Instala Node.js 20.x (versión LTS actual)
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Fedora:

```
sudo dnf install nodejs
```

Verificar instalación:

```
node --version
npm --version
```

---

4. CONFIGURACIÓN DE VS CODE

4.1 Primer arranque

Al abrir VS Code por primera vez:

1. Selecciona tu idioma preferido (Español o English)
2. Elige un tema de color (puedes cambiarlo después)
3. Cierra las pestañas de bienvenida

4.2 Configuración básica

Acceder a la configuración

- **Windows/Linux:** `Ctrl + ,` o File > Preferences > Settings
- **macOS:** `Cmd + ,` o Code > Preferences > Settings

Configuraciones recomendadas

Método 1: Interfaz gráfica

Busca y configura estas opciones en el buscador de Settings:

Opción	Valor recomendado
Auto Save	`afterDelay`
Auto Save Delay	`1000` (1 segundo)
Font Size	`14`

Tab Size	`2`
Insert Spaces	☑ Activado
Format On Save	☑ Activado
Format On Paste	☑ Activado
Word Wrap	`on`
Line Numbers	`on`

Método 2: Archivo settings.json

1. Presiona `Ctrl/Cmd + Shift + P`
2. Escribe "Preferences: Open Settings (JSON)"
3. Añade esta configuración:

```
{
  "editor.fontSize": 14,
  "editor.tabSize": 2,
  "editor.insertSpaces": true,
  "editor.formatOnSave": true,
  "editor.formatOnPaste": true,
  "editor.wordWrap": "on",
  "editor.lineNumbers": "on",
  "editor.minimap.enabled": true,
  "editor.suggestSelection": "first",
  "editor.linkedEditing": true,
  "files.autoSave": "afterDelay",
  "files.autoSaveDelay": 1000,
  "files.encoding": "utf8",
  "files.eol": "\n",
  "emmet.includeLanguages": {
    "javascript": "javascriptreact"
  },
  "emmet.triggerExpansionOnTab": true,
  "html.format.wrapAttributes": "auto",
  "css.lint.unknownAtRules": "ignore",
  "workbench.colorTheme": "Default Dark+",
  "workbench.iconTheme": "vs-seti",
  "terminal.integrated.fontSize": 13,
  "liveServer.settings.donotShowInfoMsg": true,
  "prettier.singleQuote": true,
  "prettier.semi": true,
  "prettier.tabWidth": 2
}
```

4.3 Atajos de teclado esenciales

Generales

Acción	Windows/Linux	macOS
Paleta de comandos	`Ctrl + Shift + P`	`Cmd + Shift + P`
Búsqueda rápida de archivos	`Ctrl + P`	`Cmd + P`

Abrir terminal integrada	`Ctrl + Ñ`	`Ctrl + Ñ`
Cerrar archivo	`Ctrl + W`	`Cmd + W`
Guardar	`Ctrl + S`	`Cmd + S`
Guardar todo	`Ctrl + K S`	`Cmd + K S`

Edición

Acción	Windows/Linux	macOS
Duplicar línea	`Shift + Alt + Down`	`Shift + Option + Down`
Mover línea arriba/abajo	`Alt + Up/Down`	`Option + Up/Down`
Comentar línea	`Ctrl + /`	`Cmd + /`
Comentar bloque	`Shift + Alt + A`	`Shift + Option + A`
Selección múltiple	`Ctrl + D`	`Cmd + D`
Seleccionar todas las ocurrencias	`Ctrl + Shift + L`	`Cmd + Shift + L`
Multi-cursor	`Alt + Click`	`Option + Click`

Navegación

Acción	Windows/Linux	macOS
Ir a línea	`Ctrl + G`	`Cmd + G`
Buscar	`Ctrl + F`	`Cmd + F`
Reemplazar	`Ctrl + H`	`Cmd + H`
Buscar en archivos	`Ctrl + Shift + F`	`Cmd + Shift + F`
Explorador de archivos	`Ctrl + Shift + E`	`Cmd + Shift + E`

4.4 Emmet (ya incluido en VS Code)

Emmet te permite escribir HTML y CSS más rápido. Está activado por defecto.

Ejemplos de Emmet:

```
<!-- Escribe esto y presiona Tab -->
html:5

<!-- Se expande a: -->
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```

Más ejemplos:

```
<!-- Escribe y presiona Tab -->
div.container>ul.list>li.item*3

<!-- Resultado: -->
<div class="container">
  <ul class="list">
    <li class="item"></li>
    <li class="item"></li>
    <li class="item"></li>
  </ul>
</div>
```

---

5. EXTENSIONES ESENCIALES

Cómo instalar extensiones

1. Haz clic en el icono de extensiones (cuadrados en la barra lateral) o presiona `Ctrl/Cmd + Shift + X`
2. Busca el nombre de la extensión
3. Haz clic en "Install"

5.1 Extensiones imprescindibles

1. **GitHub Copilot** ☆ ESENCIAL PARA ESTE CURSO

- **ID:** `GitHub.copilot`
- **Función:** Asistente de IA que genera código y ofrece sugerencias
- **Requisito:** Cuenta de GitHub (gratis para estudiantes)
- **Nota:** Solicita acceso gratuito como estudiante en: https://education.github.com/

2. **Live Server**

- **ID:** `ritwickdey.LiveServer`
- **Función:** Servidor local con recarga automática
- **Uso:** Click derecho en archivo HTML > "Open with Live Server"

3. **Prettier - Code formatter**

- **ID:** `esbenp.prettier-vscode`
- **Función:** Formatea automáticamente tu código
- **Configuración:** Ya configurado en settings.json anterior

4. **Auto Rename Tag**

- **ID:** `formulahendry.auto-rename-tag`
- **Función:** Renombra automáticamente las etiquetas HTML emparejadas

5. **Auto Close Tag**
   - **ID:** `formulahendry.auto-close-tag`
   - **Función:** Cierra automáticamente las etiquetas HTML

6. **HTML CSS Support**
   - **ID:** `ecmel.vscode-html-css`
   - **Función:** Autocompletado de clases CSS en HTML

7. **IntelliSense for CSS class names**
   - **ID:** `Zignd.html-css-class-completion`
   - **Función:** Autocompletado inteligente de clases CSS

8. **Path Intellisense**
   - **ID:** `christian-kohler.path-intellisense`
   - **Función:** Autocompletado de rutas de archivos

9. **Tailwind CSS IntelliSense**
   - **ID:** `bradlc.vscode-tailwindcss`
   - **Función:** Autocompletado y documentación para Tailwind CSS

5.2 Extensiones muy recomendadas

9. **GitLens**
   - **ID:** `eamodio.gitlens`
   - **Función:** Superpoderes para Git (historial, blame, etc.)

10. **ES7+ React/Redux/React-Native snippets**
   - **ID:** `dsznajder.es7-react-js-snippets`
   - **Función:** Snippets para JavaScript moderno

11. **Color Highlight**
   - **ID:** `naumovs.color-highlight`
   - **Función:** Resalta colores en tu código CSS

12. **Better Comments**
   - **ID:** `aaron-bond.better-comments`
   - **Función:** Comentarios con colores según el tipo

13. **Indent Rainbow**
   - **ID:** `oderwat.indent-rainbow`
   - **Función:** Colorea la indentación para mejor lectura

14. **Bracket Pair Colorizer 2**

- **Nota:** Ya no es necesaria, VS Code lo incluye nativamente
- **Activar en settings.json:**

```
"editor.bracketPairColorization.enabled": true
```

15. **Material Icon Theme**

- **ID:** `PKief.material-icon-theme`
- **Función:** Iconos bonitos para tipos de archivo

16. **Live Sass Compiler**

- **ID:** `ritwickdey.live-sass`
- **Función:** Compila SASS/SCSS a CSS automáticamente

17. **Jupyter** (para notebooks de Python)

- **ID:** `ms-toolsai.jupyter`
- **Función:** Soporte para Jupyter Notebooks en VS Code
- **Nota:** Se instalará automáticamente con la extensión de Python

5.3 Extensiones para validación

18. **W3C Web Validator**

- **ID:** `CelianRiboulet.webvalidator`
- **Función:** Valida tu HTML según estándares W3C

19. **ESLint**

- **ID:** `dbaeumer.vscode-eslint`
- **Función:** Linter para JavaScript

20. **stylelint**

- **ID:** `stylelint.vscode-stylelint`
- **Función:** Linter para CSS

5.4 Instalar todas las extensiones de golpe

Copia y pega este comando en la terminal integrada de VS Code:

Windows/Linux:

```
code --install-extension GitHub.copilot
code --install-extension ritwickdey.LiveServer
code --install-extension esbenp.prettier-vscode
code --install-extension formulahendry.auto-rename-tag
code --install-extension formulahendry.auto-close-tag
code --install-extension ecmel.vscode-html-css
code --install-extension Zignd.html-css-class-completion
code --install-extension christian-kohler.path-intellisense
```

```
code --install-extension eamodio.gitlens
code --install-extension dsznajder.es7-react-js-snippets
code --install-extension naumovs.color-highlight
code --install-extension aaron-bond.better-comments
code --install-extension oderwat.indent-rainbow
code --install-extension PKief.material-icon-theme
code --install-extension ms-toolsai.jupyter
code --install-extension ms-python.python
```

macOS: (usa el mismo comando que arriba)

---

6. PYTHON Y JUPYTER NOTEBOOKS

Este curso incluye ejercicios y material en Jupyter Notebooks para facilitar el aprendizaje interactivo y la generación de código con IA.

6.1 Instalación de Python

Windows

Opción 1: Descarga desde python.org (Recomendada)

1. Visita: https://www.python.org/downloads/
2. Descarga Python 3.11 o superior (versión recomendada: 3.11.7)
3. **IMPORTANTE:** Durante la instalación:
    - ☑ Marca "Add Python to PATH"
    - ☑ Marca "Install pip"
    - Selecciona "Customize installation"
    - Marca "Install for all users" (opcional)
4. Haz clic en "Install"

Verificar instalación:

```
python --version
pip --version
```

macOS

Opción 1: Homebrew (Recomendada)

```
brew install python@3.11
```

Opción 2: Descarga directa

1. Visita: https://www.python.org/downloads/macos/
2. Descarga el instalador para macOS
3. Ejecuta el instalador `.pkg`

Verificar instalación:

```
python3 --version
pip3 --version
```

Linux (Ubuntu/Debian)

```
# Python 3.11 suele venir preinstalado
python3 --version

# Si no está instalado:
sudo apt update
sudo apt install python3 python3-pip python3-venv
```

Verificar instalación:

```
python3 --version
pip3 --version
```

6.2 Instalación de Jupyter

Opción 1: Instalar Jupyter Lab (Recomendada)

```
# Windows
pip install jupyterlab notebook ipykernel

# macOS/Linux
pip3 install jupyterlab notebook ipykernel
```

Opción 2: Usar Jupyter directamente en VS Code

Con la extensión Jupyter instalada en VS Code, puedes crear y ejecutar notebooks directamente sin instalar Jupyter Lab.

Crear un notebook:

1. `Ctrl/Cmd + Shift + P`
2. Escribe "Jupyter: Create New Blank Notebook"
3. Selecciona el kernel de Python

6.3 Configurar entorno virtual (Recomendado)

Los entornos virtuales aíslan las dependencias de cada proyecto.

Crear entorno virtual

```
# Navega a tu carpeta de proyectos
cd C:\Users\TuUsuario\Documents\IFCD0110

# Windows
python -m venv venv

# macOS/Linux
python3 -m venv venv
```

Activar entorno virtual

Windows (PowerShell):

```
.\venv\Scripts\Activate.ps1

# Si hay error de permisos:
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Windows (CMD):

```
venv\Scripts\activate.bat
```

macOS/Linux:

```
source venv/bin/activate
```

Desactivar entorno:

```
deactivate
```

Instalar dependencias del curso

Con el entorno virtual activado:

```
```bash
pip install -r requirements.txt
```

Si prefieres instalar manualmente:

```
pip install jupyter jupyterlab notebook ipykernel
pip install pandas numpy matplotlib
pip install requests beautifulsoup4
pip install black autopep8 pytest python-dotenv Pillow
```

6.4 Usar Jupyter Notebooks en VS Code

Abrir un notebook

1. Abre VS Code
2. Abre un archivo `.ipynb` (o crea uno nuevo)
3. VS Code detectará automáticamente que es un notebook
4. Selecciona el kernel de Python en la esquina superior derecha

Ejecutar celdas

- **Ejecutar celda actual:** `Shift + Enter`
- **Ejecutar celda y permanecer:** `Ctrl + Enter`
- **Ejecutar todas las celdas:** Click en "Run All" en la barra superior
- **Limpiar outputs:** Click en "Clear All Outputs"

Tipos de celdas

- **Code:** Para código Python ejecutable
- **Markdown:** Para texto explicativo (documentación)

Atajos útiles en notebooks

Acción	Atajo
Ejecutar celda	`Shift + Enter`
Añadir celda arriba	`A` (en modo comando)
Añadir celda abajo	`B` (en modo comando)
```

Eliminar celda	`DD` (en modo comando)
Cambiar a Code	`Y` (en modo comando)
Cambiar a Markdown	`M` (en modo comando)
Modo comando	`Esc`
Modo edición	`Enter`

6.5 Solución de problemas con Python

Python no se encuentra en PATH

Windows:

1. Panel de Control > Sistema > Configuración avanzada del sistema
2. Variables de entorno > Path > Editar
3. Añade:
    - `C:\Users\TuUsuario\AppData\Local\Programs\Python\Python311`
    - `C:\Users\TuUsuario\AppData\Local\Programs\Python\Python311\Scripts`

Error "Jupyter not found"

```
# Reinstalar Jupyter
pip uninstall jupyter jupyterlab notebook
pip install jupyter jupyterlab notebook
```

Kernel no aparece en VS Code

```
# Registrar kernel manualmente
python -m ipykernel install --user --name=venv
```

---

7. CONFIGURACIÓN DE GIT Y GITHUB

6.1 Configuración inicial de Git

Una vez instalado Git, configura tu identidad:

```
# Configura tu nombre (aparecerá en tus commits)
git config --global user.name "Tu Nombre"

# Configura tu email (usa el mismo que usarás en GitHub)
git config --global user.email "tu.email@ejemplo.com"

# Configura el editor predeterminado (VS Code)
git config --global core.editor "code --wait"

# Configura el nombre de la rama principal
git config --global init.defaultBranch main

# Mejora la salida de colores
git config --global color.ui auto
```

```
# Verificar configuración
git config --list
```

6.2 Crear cuenta en GitHub

4. Visita: https://github.com/
5. Haz clic en "Sign up"
6. Completa el registro:
    - **Email:** Usa uno válido (recibirás un correo de verificación)
    - **Password:** Mínimo 15 caracteres o 8 con número y minúscula
    - **Username:** Elige un nombre profesional (lo verán otros)
7. Verifica tu email
8. Completa el perfil opcional

6.3 Configurar SSH para GitHub (Recomendado)

SSH te permite conectar con GitHub sin introducir usuario/contraseña cada vez.

Generar clave SSH

Todos los sistemas operativos:

```
# Genera una nueva clave SSH
ssh-keygen -t ed25519 -C "tu.email@ejemplo.com"

# Si tu sistema no soporta ed25519, usa:
ssh-keygen -t rsa -b 4096 -C "tu.email@ejemplo.com"

# Presiona Enter para aceptar la ubicación predeterminada
# Puedes dejar la passphrase vacía o poner una contraseña adicional
```

Agregar clave SSH al agente

Windows (Git Bash):

```
# Inicia el agente SSH
eval "$(ssh-agent -s)"

# Agrega tu clave SSH
ssh-add ~/.ssh/id_ed25519
```

macOS:

```
# Inicia el agente SSH
eval "$(ssh-agent -s)"

# Agrega tu clave SSH al agente
ssh-add --apple-use-keychain ~/.ssh/id_ed25519
```

Linux:

```
# Inicia el agente SSH
eval "$(ssh-agent -s)"

# Agrega tu clave SSH
ssh-add ~/.ssh/id_ed25519
```

Agregar clave SSH a GitHub

1. **Copia tu clave SSH pública:**

Windows (Git Bash):

```
cat ~/.ssh/id_ed25519.pub | clip
```

macOS:

```
pbcopy < ~/.ssh/id_ed25519.pub
```

Linux:

```
cat ~/.ssh/id_ed25519.pub
# Copia manualmente la salida
```

2. **En GitHub:**
    - Ve a Settings (tu foto de perfil > Settings)
    - En el menú lateral: "SSH and GPG keys"
    - Haz clic en "New SSH key"
    - **Title:** Ponle un nombre descriptivo (ej: "Mi Laptop Personal")
    - **Key:** Pega tu clave SSH pública
    - Haz clic en "Add SSH key"
3. **Verificar conexión:**

```
ssh -T git@github.com

# Deberías ver: "Hi username! You've successfully authenticated..."
```

7.4 Configurar HTTPS con Credential Manager (Alternativa a SSH)

Si prefieres HTTPS en lugar de SSH:

Windows: Git Credential Manager viene con Git for Windows, ya está configurado.

macOS:

```
git config --global credential.helper osxkeychain
```

Linux:

```
git config --global credential.helper store
```

7.5 Comandos Git básicos

Crear un repositorio nuevo

```
# Navega a tu carpeta de proyecto
cd /ruta/a/tu/proyecto

# Inicializa Git
git init

# Agrega archivos
git add .
```

```
# Haz tu primer commit
git commit -m "Primer commit"
```

Clonar un repositorio existente

```
# Con SSH
git clone git@github.com:usuario/repositorio.git

# Con HTTPS
git clone https://github.com/usuario/repositorio.git
```

Workflow básico

```
# Ver estado de archivos
git status

# Agregar archivos al staging
git add nombre-archivo.html
# O agregar todos los cambios
git add .

# Hacer commit
git commit -m "Descripción de los cambios"

# Ver historial
git log
git log --oneline

# Subir cambios a GitHub
git push origin main

# Descargar cambios desde GitHub
git pull origin main
```

Branches (ramas)

```
# Ver ramas
git branch

# Crear nueva rama
git branch nombre-rama

# Cambiar a una rama
git checkout nombre-rama

# Crear y cambiar a una rama (atajo)
git checkout -b nombre-rama

# Fusionar rama a main
git checkout main
git merge nombre-rama

# Eliminar rama
git branch -d nombre-rama
```

7.6 Archivo .gitignore

Crea un archivo `.gitignore` en la raíz de tu proyecto para ignorar archivos que no quieres subir:

```
# Dependencias
node_modules/
```

```
npm-debug.log*

# Python
__pycache__/
*.py[cod]
*$py.class
.Python
venv/
env/
.venv/
.ipynb_checkpoints/

# Archivos del sistema
.DS_Store
Thumbs.db
desktop.ini

# Archivos del editor
.vscode/
.idea/
*.swp
*.swo
*~

# Archivos temporales
*.tmp
*.temp
*.log

# Archivos de compilación
dist/
build/
*.min.css
*.min.js

# Variables de entorno
.env
.env.local

# Carpetas de testing
coverage/
.nyc_output/
```

---

8. NAVEGADORES PARA TESTING

Es crucial probar tus páginas web en diferentes navegadores para asegurar compatibilidad.

8.1 Navegadores esenciales

Google Chrome

- **Descarga:** https://www.google.com/chrome/
- **DevTools:** F12 o Ctrl+Shift+I (Cmd+Option+I en Mac)
- **Uso:** Navegador principal de desarrollo

Mozilla Firefox

- **Descarga:** https://www.mozilla.org/firefox/
- **DevTools:** F12 o Ctrl+Shift+I (Cmd+Option+I en Mac)
- **Uso:** Excelente para testing, DevTools muy potentes

Microsoft Edge

- **Descarga:** https://www.microsoft.com/edge
- **Nota:** Viene preinstalado en Windows 10/11
- **Uso:** Basado en Chromium, importante para usuarios Windows

Safari (solo macOS)

- **Preinstalado** en macOS
- **Habilitar DevTools:** Safari > Preferences > Advanced > Show Develop menu
- **DevTools:** Cmd+Option+I
- **Uso:** Crítico para testing en dispositivos Apple

8.2 Configurar DevTools en Chrome

Chrome DevTools es tu herramienta más importante para debugging.

Abrir DevTools

- **Método 1:** F12
- **Método 2:** Ctrl+Shift+I (Cmd+Option+I en Mac)
- **Método 3:** Click derecho > "Inspeccionar"

Paneles principales

1. **Elements:** Inspeccionar y modificar HTML/CSS en tiempo real
2. **Console:** Ver errores de JavaScript, ejecutar código
3. **Sources:** Debugging de JavaScript
4. **Network:** Ver peticiones HTTP, tiempos de carga
5. **Application:** LocalStorage, cookies, cache
6. **Lighthouse:** Auditorías de rendimiento y accesibilidad

Herramientas útiles

- **Responsive Design Mode:** Ctrl+Shift+M (Cmd+Option+M)
- **Screenshot:** En DevTools > Elementos > Click derecho en `<html>` > "Capture screenshot"
- **Color picker:** Click en cualquier color en CSS

8.3 Extensiones de navegador recomendadas

Para Chrome/Edge

1. **Web Developer**
    - URL: https://chrome.google.com/webstore/
    - Función: Herramientas de desarrollo avanzadas
2. **ColorZilla**
    - Función: Pipeta de colores avanzada
3. **WhatFont**
    - Función: Identifica fuentes en cualquier web
4. **Lighthouse**
    - Función: Ya viene integrado en Chrome DevTools
5. **JSON Viewer**
    - Función: Visualiza archivos JSON de forma legible
6. **Wappalyzer**
    - Función: Detecta tecnologías usadas en sitios web

Para Firefox

7. **Web Developer**
    - Similar a la versión de Chrome
8. **Firefox Color**
    - Personaliza el tema de Firefox

---

9. HERRAMIENTAS DE IA PARA DESARROLLO

Este curso integra herramientas de IA para acelerar el desarrollo y aprendizaje.

9.1 GitHub Copilot

Configuración:

9. Instala la extensión en VS Code (ya debería estar instalada)
10. Inicia sesión con tu cuenta de GitHub
11. Si eres estudiante, activa GitHub Student Developer Pack
12. Acepta los términos de uso

Uso básico:

- **Autocompletado:** Escribe código y Copilot sugerirá continuaciones
- **Aceptar sugerencia:** Presiona `Tab`

- **Ver más sugerencias:** `Alt + ]` o `Alt + [`
- **Abrir panel de Copilot:** `Ctrl/Cmd + I`
- **Generar código desde comentario:** Escribe un comentario y presiona `Enter`

Ejemplos de prompts:

```
// Función para validar email con regex
// [Copilot generará el código]

// Crear array de 10 números aleatorios entre 1 y 100
// [Copilot generará el código]

// Función async para fetch de API con manejo de errores
// [Copilot generará el código]
```

9.2 ChatGPT (OpenAI)

URL: https://chat.openai.com/

Usos recomendados:

- Explicar conceptos de HTML/CSS/JavaScript
- Generar código de ejemplo
- Debugging (pega tu código con el error)
- Refactorizar código
- Crear tests
- Optimizar rendimiento

Prompts efectivos:

```
"Explica cómo funciona el flexbox en CSS con ejemplos"
"Crea una función JavaScript para validar formularios"
"Optimiza este código para mejor rendimiento: [código]"
"Encuentra el error en este código: [código con error]"
"Genera tests unitarios para esta función: [función]"
```

9.3 V0.dev (Vercel)

URL: https://v0.dev/

Función: Genera componentes React/Vue desde descripciones en lenguaje natural

Uso:

1. Describe el componente que quieres crear
2. V0 genera el código con Tailwind CSS
3. Copia y adapta el código a tu proyecto

Ejemplos:

```
"Create a responsive navigation bar with dropdown menu"
"Generate a product card component with image, title, price and button"
"Build a contact form with validation"
```

9.4 Phind.com

URL: https://www.phind.com/

Función: Buscador especializado en programación con IA

Ventajas:

- Respuestas específicas para desarrolladores
- Código de ejemplo actualizado
- Referencias a documentación oficial
- Búsqueda más precisa que Google para coding

9.5 Claude (Anthropic)

URL: https://claude.ai/

Función: Asistente de IA alternativo a ChatGPT

Ventajas:

- Excelente para código largo
- Bueno para refactorización
- Análisis de código complejo

9.6 Tabnine

Extensión VS Code: `TabNine.tabnine-vscode`

Función: Alternativa a Copilot (tiene versión gratuita)

Características:

- Autocompletado de código con IA
- Funciona offline
- Soporta múltiples lenguajes

9.7 Buenas prácticas con IA

✅ DO (Hacer)

- Usa IA para aprender y entender conceptos
- Pide explicaciones del código generado
- Revisa y adapta el código generado
- Usa IA para boilerplate y código repetitivo
- Pide múltiples soluciones y compáralas
- Usa IA para debugging y optimización

❌ DON'T (No hacer)

- No copies código sin entenderlo

- No dependas 100% de la IA
- No uses código de IA en producción sin revisarlo
- No compartas código confidencial con IAs públicas
- No asumas que el código de IA es siempre correcto
- No uses IA para hacer trampa en evaluaciones

9.8 Prompts efectivos para desarrollo web

Para HTML

```
"Crea una estructura HTML5 semántica para una página de producto con header, nav, main, aside y footer"

"Genera un formulario de contacto accesible con validación HTML5"

"Crea un template de email responsive en HTML"
```

Para CSS

```
"Crea un diseño CSS Grid de 3 columnas que sea responsive"

"Genera animaciones CSS para un botón hover effect"

"Crea un sistema de diseño con CSS variables para colores y tipografía"

"Optimiza este CSS para mejor rendimiento: [código CSS]"
```

Para JavaScript

```
"Crea una clase JavaScript para validar formularios con regex"

"Implementa un carrito de compras con LocalStorage"

"Crea una función para hacer fetch con manejo de errores y retry logic"

"Refactoriza este código usando async/await en lugar de callbacks: [código]"
```

Para Debugging

```
"Este código da error [error], ayúdame a solucionarlo: [código]"

"¿Por qué mi CSS no se aplica correctamente? [código HTML y CSS]"

"Mi función JavaScript no retorna el valor esperado: [función]"
```

---

10. HERRAMIENTAS ADICIONALES

10.1 Servidores locales

Live Server (Ya instalado como extensión)

- Uso: Click derecho en HTML > "Open with Live Server"
- Puerto: http://localhost:5500

Python SimpleHTTPServer (alternativa)

```
# Python 3
python -m http.server 8000

# Python 2
python -m SimpleHTTPServer 8000

# Abre: http://localhost:8000
```

Node.js http-server (alternativa)

```
# Instalar globalmente
npm install -g http-server

# Usar
http-server -p 8000

# Abre: http://localhost:8000
```

8.2 Herramientas de línea de comandos

npm packages útiles

```
# Prettier (formateo de código)
npm install -g prettier

# ESLint (linting JavaScript)
npm install -g eslint

# Browsersync (sincronización multi-navegador)
npm install -g browser-sync
```

Testing (se instalarán por proyecto)

- **Jest:** Framework de testing unitario
- **Playwright:** Testing end-to-end (E2E)

10.3 Herramientas de diseño y assets

Figma

- **URL:** https://www.figma.com/
- **Tipo:** Herramienta de diseño colaborativa
- **Uso:** Diseñar mockups y prototipos

GIMP

- **URL:** https://www.gimp.org/
- **Tipo:** Editor de imágenes (alternativa gratuita a Photoshop)
- **Uso:** Editar y optimizar imágenes

Inkscape

- **URL:** https://inkscape.org/

- **Tipo:** Editor de gráficos vectoriales
- **Uso:** Crear y editar SVG

TinyPNG

- **URL:** https://tinypng.com/
- **Tipo:** Compresor de imágenes online
- **Uso:** Optimizar PNG y JPG

10.4 Herramientas online útiles

Can I Use

- **URL:** https://caniuse.com/
- **Función:** Verificar compatibilidad de características CSS/HTML/JS

CodePen

- **URL:** https://codepen.io/
- **Función:** Editor online para prototipos rápidos

CSS Tricks

- **URL:** https://css-tricks.com/
- **Función:** Recursos y tutoriales de CSS

MDN Web Docs

- **URL:** https://developer.mozilla.org/
- **Función:** Documentación oficial de web technologies

Google Fonts

- **URL:** https://fonts.google.com/
- **Función:** Fuentes gratuitas para tus proyectos

Font Awesome

- **URL:** https://fontawesome.com/
- **Función:** Iconos vectoriales

10.5 Validadores

W3C Markup Validator

- **URL:** https://validator.w3.org/
- **Función:** Valida tu HTML

W3C CSS Validator

- **URL:** https://jigsaw.w3.org/css-validator/

- **Función:** Valida tu CSS

WAVE Web Accessibility Evaluator

- **URL:** https://wave.webaim.org/
- **Función:** Evalúa accesibilidad web

---

11. ESTRUCTURA DE CARPETAS

9.1 Estructura básica recomendada

```
mi-proyecto/
│
├── index.html
├── about.html
├── contact.html
│
├── css/
│   ├── style.css
│   ├── normalize.css
│   └── responsive.css
│
├── js/
│   ├── main.js
│   └── utils.js
│
├── img/
│   ├── logo.png
│   ├── hero-banner.jpg
│   └── icons/
│       ├── icon-1.svg
│       └── icon-2.svg
│
├── fonts/
│   └── custom-font.woff2
│
├── assets/
│   └── documents/
│       └── brochure.pdf
│
├── .gitignore
├── README.md
└── package.json (si usas npm)
```

11.2 Convenciones de nombres

Archivos y carpetas

- **Usar minúsculas:** `style.css` ✅ no `Style.css` ❌
- **Usar guiones:** `mi-archivo.html` ✅ no `mi_archivo.html` ❌
- **Ser descriptivo:** `hero-banner.jpg` ✅ no `img1.jpg` ❌
- **Sin espacios:** `about-us.html` ✅ no `about us.html` ❌
- **Sin caracteres especiales:** `contacto.html` ✅ no `contáctó.html` ❌

HTML

```
<!-- IDs: camelCase -->
<div id="mainContent"></div>

<!-- Clases: kebab-case -->
<div class="header-navigation"></div>

<!-- Data attributes: kebab-case -->
<div data-user-id="123"></div>
```

CSS

```
/* Clases: kebab-case */
.header-navigation {
}

/* IDs: camelCase */
#mainContent {
}

/* Variables CSS: kebab-case */
:root {
  --primary-color: #007bff;
}
```

JavaScript

```
// Variables y funciones: camelCase
const userName = "Juan";
function getUserData() {}

// Clases: PascalCase
class UserProfile {}

// Constantes: UPPER_SNAKE_CASE
const MAX_USERS = 100;
```

11.3 Crear estructura desde terminal

Windows (PowerShell):

```
# Crear proyecto
New-Item -ItemType Directory -Path "mi-proyecto"
cd mi-proyecto

# Crear carpetas
New-Item -ItemType Directory -Path "css", "js", "img", "fonts", "assets"

# Crear archivos
New-Item -ItemType File -Path "index.html", "css\style.css", "js\main.js"
```

macOS/Linux (Bash):

```
# Crear proyecto
mkdir mi-proyecto
cd mi-proyecto

# Crear carpetas
mkdir -p css js img fonts assets/documents
```

```
# Crear archivos
touch index.html css/style.css js/main.js

# Inicializar Git
git init

# Crear .gitignore
echo "node_modules/" > .gitignore
```

11.4 Template HTML5 básico

Crea `index.html` con este contenido:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta name="description" content="Descripción de tu página" />
    <meta name="keywords" content="html, css, javascript" />
    <meta name="author" content="Tu Nombre" />

    <title>Mi Proyecto Web</title>

    <!-- CSS -->
    <link rel="stylesheet" href="css/style.css" />

    <!-- Favicon -->
    <link rel="icon" type="image/x-icon" href="img/favicon.ico" />
  </head>
  <body>
    <header>
      <nav>
        <!-- Navegación -->
      </nav>
    </header>

    <main>
      <section>
        <!-- Contenido principal -->
      </section>
    </main>

    <footer>
      <!-- Pie de página -->
    </footer>

    <!-- JavaScript -->
    <script src="js/main.js"></script>
  </body>
</html>
```

---

12. VERIFICACIÓN DEL SETUP

12.1 Checklist de verificación

Ejecuta estos comandos en tu terminal para verificar que todo está instalado correctamente:

```
# VS Code
code --version

# Git
git --version

# Node.js
node --version

# npm
npm --version

# Verificar configuración de Git
git config --list
```

Salida esperada:

```
# VS Code
1.85.0 (o superior)

# Git
git version 2.42.0 (o superior)

# Node.js
v20.10.0 (o superior)

# npm
10.2.0 (o superior)
```

12.2 Proyecto de prueba

Crea un proyecto simple para verificar que todo funciona:

Paso 1: Crear carpeta y archivos

```
mkdir test-proyecto
cd test-proyecto
code .
```

Paso 2: Crear `index.html`

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Test Setup</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <h1>¡Mi entorno funciona!</h1>
    <p class="success">✅ VS Code instalado</p>
    <p class="success">✅ Live Server funcionando</p>
    <p class="success">✅ CSS aplicado</p>

    <button onclick="testJS()">Click para probar JavaScript</button>

    <script src="script.js"></script>
  </body>
</html>
```

Paso 3: Crear `style.css`

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  color: white;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  padding: 20px;
}

h1 {
  font-size: 3rem;
  margin-bottom: 2rem;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}

.success {
  font-size: 1.5rem;
  margin: 10px 0;
  animation: fadeIn 1s ease-in;
}

@keyframes fadeIn {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

button {
  margin-top: 2rem;
  padding: 15px 30px;
  font-size: 1.2rem;
  background-color: #fff;
  color: #667eea;
  border: none;
  border-radius: 50px;
  cursor: pointer;
  transition: all 0.3s ease;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
}

button:hover {
  transform: translateY(-3px);
  box-shadow: 0 6px 20px rgba(0, 0, 0, 0.3);
}
```

```
button:active {
  transform: translateY(0);
}
```

Paso 4: Crear `script.js`

```
function testJS() {
  alert("🎉 ¡JavaScript funciona correctamente!");
  console.log("✅ Todo configurado correctamente");
}

// Test en consola
console.log(
  "%c¡Entorno de desarrollo listo!",
  "color: #667eea; font-size: 20px; font-weight: bold;"
);
```

Paso 5: Probar Live Server

1. Click derecho en `index.html`
2. Selecciona "Open with Live Server"
3. Debería abrirse tu navegador en `http://localhost:5500`
4. Haz cambios en el CSS y verifica que se actualizan automáticamente

Paso 6: Inicializar Git

```
git init
git add .
git commit -m "Proyecto de prueba inicial"
```

12.3 Verificar extensiones de VS Code

1. Presiona `Ctrl/Cmd + Shift + X`
2. Verifica que están instaladas las extensiones mencionadas anteriormente
3. Cada una debe tener un botón "Disable" o "Uninstall" (no "Install")

12.4 Test de DevTools

1. Con tu página de prueba abierta, presiona F12
2. Ve a la pestaña "Console"
3. Escribe: `console.log('Test DevTools')`
4. Presiona Enter
5. Deberías ver el mensaje impreso

---

13. SOLUCIÓN DE PROBLEMAS COMUNES

13.1 VS Code

Problema: VS Code no abre desde terminal

Windows:

```
# Verifica que esté en PATH
where code

# Si no aparece, reinstala VS Code marcando "Add to PATH"
```

macOS:

```
# Instala el comando shell
# En VS Code: Cmd+Shift+P > "Shell Command: Install 'code' command in PATH"
```

Linux:

```
# Crea un symlink manualmente
sudo ln -s /usr/share/code/bin/code /usr/local/bin/code
```

Problema: Extensiones no funcionan

1. Recarga VS Code: `Ctrl/Cmd + Shift + P` > "Reload Window"
2. Deshabilita y vuelve a habilitar la extensión
3. Verifica actualizaciones: Help > Check for Updates
4. Reinstala la extensión

Problema: Live Server no inicia

- Verifica que no haya otro servicio en el puerto 5500
- Cierra otros Live Servers activos (barra inferior de VS Code)
- Reinstala la extensión
- Verifica que el archivo sea `.html`

13.2 Git

Problema: Git no reconocido en terminal

Windows:

```
# Verifica instalación
git --version

# Si no funciona, añade a PATH manualmente:
# Panel de Control > Sistema > Configuración avanzada del sistema
# Variables de entorno > Path > Editar
# Añade: C:\Program Files\Git\cmd
```

macOS/Linux:

```
# Verifica instalación
which git
```

```
# Si no está, reinstala desde el gestor de paquetes
```

Problema: Error SSH "Permission denied"

```
# Verifica que la clave está agregada
ssh-add -l

# Si no aparece, agrégala de nuevo
ssh-add ~/.ssh/id_ed25519

# Verifica permisos del archivo
chmod 600 ~/.ssh/id_ed25519
chmod 644 ~/.ssh/id_ed25519.pub
```

Problema: Git pide usuario y contraseña constantemente

Solución 1: Usar SSH en lugar de HTTPS

```
# Cambia remote de HTTPS a SSH
git remote set-url origin git@github.com:usuario/repositorio.git
```

Solución 2: Configurar credential helper

```
# Windows
git config --global credential.helper wincred

# macOS
git config --global credential.helper osxkeychain

# Linux
git config --global credential.helper store
```

Problema: "fatal: not a git repository"

```
# Asegúrate de estar en la carpeta correcta
pwd  # o cd en Windows

# Inicializa Git si es necesario
git init
```

13.3 Node.js y npm

Problema: npm command not found

Solución: Reinstala Node.js desde nodejs.org

Verificar PATH:

```
# Windows
echo %PATH%

# macOS/Linux
echo $PATH
```

Problema: Error de permisos al instalar paquetes globales

macOS/Linux:

```
# NO uses sudo, mejor cambia el directorio de npm
mkdir ~/.npm-global
```

```
npm config set prefix '~/.npm-global'
echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

Windows: Ejecuta PowerShell como Administrador

Problema: Versión antigua de npm

```
# Actualizar npm a la última versión
npm install -g npm@latest
```

13.4 Navegadores

Problema: DevTools no se abre

- Verifica que usas el atajo correcto: F12 o Ctrl+Shift+I
- En Chrome/Edge: Menú (⋮) > More Tools > Developer Tools
- Verifica que DevTools no esté en ventana separada (mira la barra de tareas)

Problema: Live reload no funciona

1. Limpia cache del navegador: Ctrl+Shift+Del
2. Recarga sin cache: Ctrl+F5 (Cmd+Shift+R en Mac)
3. Verifica que Live Server esté activo (barra inferior de VS Code)
4. Prueba otro navegador

13.5 Problemas generales

Problema: Caracteres raros en la terminal

Solución: Cambiar encoding a UTF-8

VS Code:

```
// settings.json
"terminal.integrated.defaultProfile.windows": "PowerShell",
"terminal.integrated.profiles.windows": {
  "PowerShell": {
    "source": "PowerShell",
    "args": ["-NoExit", "-Command", "chcp 65001"]
  }
}
```

Problema: Firewall bloquea Live Server

Windows:

1. Panel de Control > Sistema y seguridad > Firewall de Windows Defender
2. Configuración avanzada > Reglas de entrada
3. Nueva regla > Puerto > TCP > 5500
4. Permitir la conexión

Problema: Espacios en rutas causan errores

Solución: Evita espacios en nombres de carpetas

```
# Mal
C:\Mis Documentos\Mi Proyecto Web\

# Bien
C:\Documents\mi-proyecto-web\
```

---

14. RECURSOS ADICIONALES

14.1 Documentación oficial

HTML

- **MDN Web Docs:** https://developer.mozilla.org/es/docs/Web/HTML
- **W3C HTML Spec:** https://html.spec.whatwg.org/
- **HTML Reference:** https://htmlreference.io/

CSS

- **MDN CSS:** https://developer.mozilla.org/es/docs/Web/CSS
- **CSS Tricks:** https://css-tricks.com/
- **CSS Reference:** https://cssreference.io/
- **Can I Use:** https://caniuse.com/

JavaScript

- **MDN JavaScript:** https://developer.mozilla.org/es/docs/Web/JavaScript
- **JavaScript.info:** https://javascript.info/
- **Eloquent JavaScript:** https://eloquentjavascript.net/

Git

- **Pro Git Book:** https://git-scm.com/book/es/v2
- **GitHub Docs:** https://docs.github.com/es
- **Git Cheat Sheet:** https://training.github.com/downloads/es_ES/github-git-cheat-sheet/

14.2 Cursos y tutoriales

Plataformas gratuitas

- **freeCodeCamp:** https://www.freecodecamp.org/
- **The Odin Project:** https://www.theodinproject.com/
- **W3Schools:** https://www.w3schools.com/
- **Codecademy:** https://www.codecademy.com/ (tiene versión gratuita)

YouTube (canales recomendados)

- **Traversy Media** (inglés)
- **The Net Ninja** (inglés)
- **Fazt** (español)
- **MoureDev** (español)
- **midudev** (español)

Plataformas de pago (opcional)

- **Udemy:** Cursos desde 10-15€
- **Platzi:** Suscripción mensual
- **LinkedIn Learning:** Con prueba gratuita

14.3 Herramientas de práctica

Coding Challenges

- **Frontend Mentor:** https://www.frontendmentor.io/
- **CSS Battle:** https://cssbattle.dev/
- **JavaScript30:** https://javascript30.com/
- **100 Days CSS:** https://100dayscss.com/

Editores online

- **CodePen:** https://codepen.io/
- **JSFiddle:** https://jsfiddle.net/
- **CodeSandbox:** https://codesandbox.io/
- **StackBlitz:** https://stackblitz.com/

14.4 Comunidades

Foros y comunidades

- **Stack Overflow:** https://stackoverflow.com/
- **Dev.to:** https://dev.to/
- **Reddit - r/webdev:** https://reddit.com/r/webdev
- **Discord - Frontend Developers**

Slack/Discord de programación

- **Frontend Café** (Slack - español)
- **freeCodeCamp** (Discord - inglés)
- **The Programmer's Hangout** (Discord - inglés)

14.5 Newsletters y blogs

Newsletters

- **Frontend Focus:** https://frontendfoc.us/
- **JavaScript Weekly:** https://javascriptweekly.com/
- **CSS Weekly:** https://css-weekly.com/

Blogs recomendados

- **CSS-Tricks:** https://css-tricks.com/
- **Smashing Magazine:** https://www.smashingmagazine.com/
- **A List Apart:** https://alistapart.com/
- **web.dev:** https://web.dev/

14.6 Herramientas de diseño

Paletas de colores

- **Coolors:** https://coolors.co/
- **Adobe Color:** https://color.adobe.com/
- **Color Hunt:** https://colorhunt.co/

Tipografía

- **Google Fonts:** https://fonts.google.com/
- **Font Pair:** https://fontpair.co/
- **Type Scale:** https://type-scale.com/

Iconos

- **Font Awesome:** https://fontawesome.com/
- **Heroicons:** https://heroicons.com/
- **Feather Icons:** https://feathericons.com/
- **Ionicons:** https://ionic.io/ionicons

Imágenes stock

- **Unsplash:** https://unsplash.com/
- **Pexels:** https://www.pexels.com/
- **Pixabay:** https://pixabay.com/
- **Lorem Picsum:** https://picsum.photos/ (placeholders)

14.7 Cheat sheets útiles

- **HTML Cheat Sheet:** https://htmlcheatsheet.com/

- **CSS Cheat Sheet:** https://htmlcheatsheet.com/css/
- **Flexbox Cheat Sheet:** https://flexbox.malven.co/
- **Grid Cheat Sheet:** https://grid.malven.co/
- **Git Cheat Sheet:** https://education.github.com/git-cheat-sheet-education.pdf

14.8 Accesibilidad

- **WCAG Guidelines:** https://www.w3.org/WAI/WCAG21/quickref/
- **WAVE:** https://wave.webaim.org/
- **axe DevTools:** Extensión de navegador
- **A11y Project:** https://www.a11yproject.com/

14.9 Performance y optimización

- **PageSpeed Insights:** https://pagespeed.web.dev/
- **GTmetrix:** https://gtmetrix.com/
- **WebPageTest:** https://www.webpagetest.org/
- **Lighthouse:** Integrado en Chrome DevTools

14.10 Libros recomendados (gratuitos online)

- **Eloquent JavaScript:** https://eloquentjavascript.net/
- **You Don't Know JS:** https://github.com/getify/You-Dont-Know-JS
- **Learning JavaScript Design Patterns:** https://www.patterns.dev/

🎉 ¡ENHORABUENA!

Has completado la configuración de tu entorno de desarrollo profesional. Ahora estás listo para empezar a crear páginas web.

Próximos pasos

1. ✅ Practica con el proyecto de prueba
2. ✅ Familiarízate con VS Code y sus atajos
3. ✅ Crea tu primer repositorio en GitHub
4. ✅ Explora las DevTools del navegador
5. ✅ Comienza tu primer proyecto del curso IFCD0110

Recuerda

- **Practica diariamente:** La constancia es clave
- **Lee documentación:** MDN es tu mejor amigo
- **Experimenta:** No tengas miedo de romper cosas

- **Usa Git:** Haz commits frecuentes
- **Pide ayuda:** La comunidad está para ayudar

📞 SOPORTE

Si encuentras problemas no cubiertos en esta guía:

1. **Revisa la sección de solución de problemas**
2. **Consulta la documentación oficial** de cada herramienta
3. **Busca en Stack Overflow**
4. **Pregunta a tu instructor** del curso IFCD0110

¡Feliz coding! 🚀👨‍💻👩‍💻