

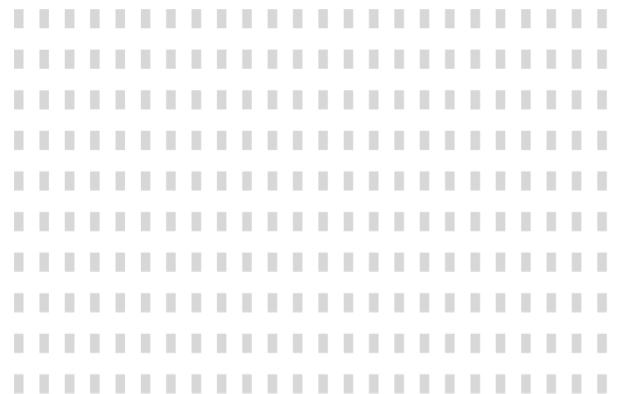


Universidad Tecnológica de *Tijuana*

# CONFIGURACIÓN DEL ENTORNO PARA DESARROLLO WEB

Tecnologías de la información  
Área desarrollo de software multiplataforma

Alumno: Diaz Juarez Oscar Armando  
Matricula: 0323105846  
Grupo: 3C  
Materia: Desarrollo de Aplicaciones Web  
Maestro(a): Ray Brunett Parra Galaviz



Fecha de entrega: 29 de agosto del 2024

## 1. Elegir un Editor de Código

El primer paso es seleccionar un editor de código o un IDE (Entorno de Desarrollo Integrado) que te permita escribir, organizar y gestionar tu código de manera eficiente. **Visual Studio Code** es uno de los más populares debido a su facilidad de uso, personalización mediante extensiones, y soporte para múltiples lenguajes de programación. **Sublime Text** y **JetBrains WebStorm** son otras opciones recomendadas, cada una con sus propias ventajas en cuanto a velocidad y características avanzadas.

## 2. Instalar Node.js

Node.js es esencial para cualquier entorno de desarrollo web, ya que permite ejecutar JavaScript en el servidor, facilitando tanto el desarrollo front-end como back-end. Además, Node.js incluye npm (Node Package Manager), una herramienta indispensable para instalar y gestionar dependencias y bibliotecas que necesitas para tu proyecto. La instalación de Node.js es simple y se puede verificar rápidamente mediante comandos en la terminal.

## 3. Configurar Git

Git es una herramienta de control de versiones que te permite rastrear y gestionar los cambios en tu código a lo largo del tiempo. Es fundamental para trabajar en equipo y mantener un historial organizado de tu proyecto. Instalar Git y configurar una cuenta en plataformas como **GitHub** o **GitLab** te permite sincronizar tu trabajo con repositorios remotos, facilitando la colaboración y el despliegue continuo de tu aplicación.

## 4. Instalar un Framework o Librería de Front-End

Dependiendo de la naturaleza de tu proyecto, necesitarás un framework o una librería para construir la interfaz de usuario (front-end). **React** es una opción popular por su flexibilidad y amplio ecosistema. **Vue.js** es conocido por su simplicidad y facilidad de integración, mientras que **Angular** ofrece un framework completo con una estructura robusta ideal para aplicaciones grandes y complejas. Instalar y configurar cualquiera de estas herramientas te proporcionará la base para desarrollar interfaces de usuario dinámicas y reactivas.

## 5. Configurar un Servidor Web Local

Para probar tu aplicación web mientras la desarrollas, es necesario configurar un servidor web local. Herramientas como **Live Server** para Visual Studio Code permiten lanzar tu aplicación en un servidor local con un solo clic, actualizándose automáticamente cada vez que guardas cambios en tu código. Alternativamente, **http-server** es una opción ligera que puedes instalar mediante npm, proporcionando un servidor HTTP simple para probar tu aplicación.

## 6. Configurar un Entorno de Back-End

Si tu proyecto incluye un componente de back-end, necesitarás configurar un entorno adecuado para manejar la lógica del servidor, la gestión de usuarios, bases de datos, y más. **Node.js con Express** es una combinación muy utilizada para aplicaciones de back-end en JavaScript, proporcionando una estructura ligera y fácil de configurar. **Django** es un framework robusto en Python que facilita el desarrollo rápido de aplicaciones web con todas las herramientas necesarias integradas, mientras que **Ruby on Rails** es ideal para desarrollar aplicaciones de manera rápida y eficiente en Ruby.

## 7. Configurar una Base de Datos

Las aplicaciones web suelen requerir una base de datos para almacenar y gestionar datos. Dependiendo de tus necesidades, puedes optar por una base de datos relacional como **MySQL** o **PostgreSQL**, que son ideales para estructuras de datos complejas y transacciones. Si tu proyecto maneja grandes volúmenes de datos no estructurados, una base de datos NoSQL como **MongoDB** podría ser más adecuada. Configurar y enlazar tu base de datos al back-end de tu aplicación es un paso crucial en el desarrollo.

## 8. Pruebas y Depuración

Para asegurar la calidad de tu aplicación, es importante configurar herramientas de pruebas y depuración. **Jest** es una opción popular para realizar pruebas unitarias en JavaScript, permitiéndote verificar que las diferentes partes de tu aplicación funcionen como se espera. Además, los depuradores integrados en editores de código como Visual Studio Code, junto con las herramientas de desarrollo del navegador, son esenciales para identificar y resolver errores en tu código.

## 9. Despliegue

Una vez que tu aplicación está lista para ser utilizada, necesitarás desplegarla en un servidor para que esté accesible a los usuarios. Dependiendo de la naturaleza de tu aplicación, puedes elegir entre servicios como **GitHub Pages** para sitios estáticos, o **Heroku** y **Netlify** para aplicaciones más complejas que requieren servidores dinámicos. Estos servicios simplifican el proceso de despliegue y gestión de tu aplicación en producción, permitiéndote enfocarte en su desarrollo y mejora continua.