# **🧠 TEMARIO ESPECÍFICO PARA DESARROLLO WEB FULL STACK**

## **🔰 NIVEL 0: INTRODUCCIÓN Y BASES**

### **1. Fundamentos de programación (con JavaScript)**

* Tipos de datos: string, number, boolean, null, undefined
* Variables (let, const)
* Operadores aritméticos y lógicos
* Estructuras de control (if, else, switch)
* Bucles (for, while)
* Funciones (declarativas, expresiones, flecha)
* Arrays y objetos
* Métodos comunes (map, filter, reduce)
* Introducción a la consola del navegador y depuración

## **🧱 NIVEL 1: FRONT-END BÁSICO**

### **2. HTML**

* Estructura de una página (<!DOCTYPE html>, <head>, <body>)
* Etiquetas semánticas (<header>, <section>, <article>, <footer>)
* Formularios y validaciones básicas
* Tablas, listas, enlaces, imágenes, videos
* Buenas prácticas de accesibilidad (atributos alt, aria-label)

### **3. CSS**

* Selectores, clases, IDs
* Colores, fuentes, márgenes, paddings, borders
* Box model
* Flexbox y Grid
* Posicionamiento (relative, absolute, fixed, sticky)
* Media queries y diseño responsive

### **4. JavaScript en el navegador**

* Manipulación del DOM
* Eventos (click, submit, keydown, etc.)
* Modificación de clases, estilos y atributos
* Manejo básico de formularios
* Temporizadores (setTimeout, setInterval)

## **⚛️ NIVEL 2: FRONT-END MODERNO**

### **5. JavaScript moderno (ES6+)**

* Desestructuración
* Template literals
* Operador ternario y cortocircuito
* Promesas y async/await
* Módulos (import/export)

### **6. Git y GitHub**

* git init, git add, git commit
* Branching y merge
* Subida de repos a GitHub
* Manejo de conflictos básicos

### **7. React.js (librería moderna de front-end)**

* JSX y componentes
* Props y state
* Hooks (useState, useEffect)
* Eventos y formularios en React
* Router (React Router)
* Manejo de listas y claves
* Fetch API desde React
* Context API (gestión de estado global)

## **⚙️ NIVEL 3: BACK-END CON NODE.JS**

### **8. Node.js + Express.js**

* Qué es Node.js y cómo funciona
* Estructura de una app Express
* Rutas, controladores y middlewares
* Servir archivos estáticos
* Lectura de datos con POST (body-parser)

### **9. Bases de datos**

* MongoDB + Mongoose:  
  + Crear modelos
  + CRUD completo
  + Validaciones
  + Relaciones simples (populate)
* Introducción a PostgreSQL (opcional)

### **10. Autenticación y seguridad**

* Registro y login con JWT
* Cifrado con bcrypt
* Validación con Express Validator
* Roles de usuario
* Middleware de autenticación
* Seguridad básica: CORS, headers, sanitización

## **🚀 NIVEL 4: HERRAMIENTAS MODERNAS Y DEPLOY**

### **11. Control de versiones y colaboración**

* Git avanzado: rebase, stash, reset
* GitHub Flow
* Pull requests y revisión de código

### **12. Testing básico**

* Test de componentes React (Jest + React Testing Library)
* Test de rutas Express (Supertest)

### **13. Deploy y CI/CD**

* Front-end:  
  + Deploy en Netlify / Vercel
* Back-end:  
  + Deploy en Render / Railway / Fly.io
  + Variables de entorno
* Introducción a Docker (opcional)

## **🧠 NIVEL 5: PROYECTOS Y PRÁCTICA REAL**

### **14. Proyectos sugeridos (por nivel)**

* Nivel 1: To-do list, portafolio personal, blog simple en HTML/CSS
* Nivel 2: Weather app con API, clon de Google Keep en React
* Nivel 3: CRUD completo MERN (Mongo + Express + React + Node)
* Nivel 4: E-commerce simple con autenticación y carrito
* Nivel 5: Plataforma tipo Trello o red social básica

### **15. Preparación profesional**

* Portafolio web propio
* LinkedIn y CV para desarrolladores
* Entrevistas técnicas (preguntas frecuentes + desafíos de código)

## **🎁 OPCIONAL: BONUS**

### **🔧 Herramientas que puedes ir aprendiendo**

* Tailwind CSS
* TypeScript (React + Node)
* GraphQL
* Prisma (ORM para bases SQL)
* WebSockets (para apps en tiempo real)
* Next.js (React con SSR y API routes)

¿Quieres que lo convierta en un PDF descargable o en una hoja de ruta visual tipo checklist? También puedo darte un cronograma (por semanas o meses) si lo quieres seguir paso a paso.