**# Ruta de Aprendizaje para Programadora Junior de React**

Aquí tienes una ruta de aprendizaje estructurada para convertirte en una programadora junior de React, teniendo en cuenta que ya tienes conocimientos básicos de HTML5 y CSS3.

**## Ruta Principal: Fundamentos de Desarrollo Frontend con React**

**### 1. JavaScript ES6+ (Fundamentos esenciales)**

- \*\*Conceptos clave\*\*:

- Variables (let, const) y scope

- Funciones (declarativas, expresiones, arrow functions)

- Arrays y métodos de array (map, filter, reduce)

- Objetos y destructuring

- Promesas y async/await

- Módulos (import/export)

- Spread operator y rest parameters

### 2. React Fundamentals

- \*\*Conceptos clave\*\*:

- JSX y su sintaxis

- Componentes (funcionales y de clase)

- Props y prop drilling

- Estado con useState hook

- Efectos secundarios con useEffect hook

- Listas y keys

- Manejo de eventos

- Conditional rendering

### 3. React Intermedio

- \*\*Conceptos clave\*\*:

- Hooks personalizados

- useReducer y useContext para manejo de estado

- React Router para navegación

- Component composition

- Render props y Higher-Order Components (HOC)

- Refs y forwardRef

- Optimización de rendimiento (React.memo, useMemo, useCallback)

### 4. Manejo de Estado Avanzado

- \*\*Conceptos clave\*\*:

- Context API para estado global

- Librerías de estado (Redux Toolkit o Zustand)

- Persistencia de estado (localStorage, sessionStorage)

- Patrones de manejo de estado

### 5. Pruebas (Testing)

- \*\*Conceptos clave\*\*:

- Jest como framework de testing

- React Testing Library

- Unit testing de componentes

- Mocking de funciones y APIs

- Testing de hooks personalizados

### 6. Build Tools y Deployment

- \*\*Conceptos clave\*\*:

- Create React App y Vite

- Webpack básico (conceptos)

- Build y optimización para producción

- Deployment en plataformas (Vercel, Netlify, GitHub Pages)

## Tecnologías Complementarias

### 1. Control de Versiones (Git)

- \*\*Conceptos clave\*\*:

- Repositorios y commits

- Ramas y merging

- Resolución de conflictos

- Plataformas (GitHub, GitLab, Bitbucket)

- Trabajo colaborativo (pull requests, code reviews)

### 2. Preprocesadores CSS y CSS-in-JS

- \*\*Conceptos clave\*\*:

- SASS/SCSS (variables, nesting, mixins)

- Styled-components o Emotion

- Frameworks CSS (Tailwind CSS, Bootstrap)

- Diseño responsive

- CSS Modules

### 3. Peticiones HTTP y APIs

- \*\*Conceptos clave\*\*:

- Fetch API

- Librerías (Axios)

- RESTful APIs

- Manejo de estados de carga y error

- Custom hooks para fetching de datos

### 4. Herramientas de Desarrollo

- \*\*Conceptos clave\*\*:

- React Developer Tools

- Debugging en navegador

- ESLint y Prettier para calidad de código

- Validación de props con PropTypes o TypeScript

## Orden Recomendado de Aprendizaje

1. \*\*JavaScript ES6+\*\* (2-4 semanas)

- Domina los fundamentos antes de avanzar a React

2. \*\*React Fundamentals\*\* (3-5 semanas)

- Crea varios proyectos pequeños para practicar

3. \*\*Git y Control de Versiones\*\* (1 semana)

- Aprende mientras practicas React

4. \*\*Preprocesadores CSS\*\* (1 semana)

- Mejora tus habilidades de estilado

5. \*\*Peticiones HTTP y APIs\*\* (2 semanas)

- Conecta tus aplicaciones React con datos reales

6. \*\*React Intermedio\*\* (3-4 semanas)

- Profundiza en conceptos más avanzados

7. \*\*Manejo de Estado Avanzado\*\* (2-3 semanas)

- Implementa estado global en tus aplicaciones

8. \*\*Pruebas\*\* (2 semanas)

- Aprende a escribir tests para tus componentes

9. \*\*Build Tools y Deployment\*\* (1 semana)

- Publica tus proyectos en línea

## Proyectos Recomendados para Practicar

1. Lista de tareas (Todo App) con funcionalidad CRUD

2. Clima App que consuma una API pública

3. Buscador de películas/series

4. Carrito de compras

5. Blog personal con posts

6. Clon de una red social simple

Recuerda que la práctica constante es clave. Construye proyectos aunque sean pequeños, y gradualmente aumenta su complejidad. ¡Mucho éxito en tu camino para convertirte en programadora React!