

- 🖥️ Modalidad: Sincrónico / Asincrónico
- 📺 Curso grabado
- 📄 Incluye certificado digital
- 🕒 1 Lección - 3 horas c / semana (18 horas en total)

## Lección 1: Diseño Responsivo

Media Queries: Sintaxis y uso.  
Unidades Relativas: em, rem, vw, vh.  
Diseño Fluido: Uso de porcentajes y unidades flexibles.  
Frameworks y Herramientas: Uso de frameworks CSS como Bootstrap, fundamentos.

## Lección 2: Selectores Avanzados y Pseudoelementos

Selectores Avanzados: :nth-of-type(), :not(), :nth-last-child(), etc.  
Pseudoelementos: ::before, ::after, ::first-letter, ::first-line.  
Selectores Combinados: Usar múltiples selectores y pseudoelementos en combinación.  
Aplicaciones Prácticas: Ejemplos y ejercicios complejos.

## Lección 3: CSS Grid Layout

Definición de contenedores y elementos de grid.  
Propiedades del Contenedor: grid-template-columns, grid-template-rows, grid-gap.  
Ubicación de Elementos: grid-column, grid-row, grid-area.  
Diseño Complejo: Crear layouts responsivos y complejos utilizando CSS Grid.

-  Modalidad: Sincrónico / Asincrónico
-  Curso grabado
-  Incluye certificado digital
-  1 Lección - 3 horas c / semana (18 horas en total)

#### Lección 4: CSS Flexbox en Profundidad

Repaso de Conceptos Básicos: Contenedor flex, items flex.  
Propiedades flex-direction, flex-wrap, justify-content, align-items, align-content.  
Propiedades order, flex-grow, flex-shrink, flex-basis, align-self.  
Diseños Avanzados: Ejemplos y prácticas de layouts complejos usando Flexbox.

#### Lección 5: Animaciones y Transiciones Avanzadas

Transiciones: Propiedades avanzadas, múltiples propiedades, timing functions.  
Animaciones: @keyframes, propiedades de animación  
Interacción con JavaScript: Activar animaciones mediante eventos.  
Proyectos Prácticos: Crear animaciones complejas y transiciones fluidas.

#### Lección 6: Metodologías y Mejores Prácticas en CSS

BEM (Block Element Modifier): Principios y ejemplos.  
OOCSS (Object-Oriented CSS)  
SMACSS (Scalable and Modular Architecture for CSS)  
CSS Modular: Crear componentes reutilizables.  
Optimización y Performance: Minificación, uso eficiente de selectores, evitar redundancia.