



## IFCD65. Front end o web developer.

**Sku:** PS\_IFCD65

**Horas:** 240

**Formato:** HTML

### OBJETIVOS

- Desarrollar aplicaciones web en el ámbito Front End para trabajar en el sector digital.
- Desarrollar webs basadas en HTML y CSS utilizando un entorno profesional de desarrollo y herramientas de gestión de proyectos.
- Crear proyectos web interactivos con HTML, CSS y JavaScript (ES6) con el fin de asentar los fundamentos de la programación web.
- Utilizar un framework de desarrollo web basado en JavaScript, como Vue.js, y extrapolar el proceso a otros frameworks a partir de la experimentación.

### CONTENIDOS

- **Unidad 1. Desarrollo web y gestión de proyectos.**
  - Conocimientos / Capacidades cognitivas y prácticas
    - Identificación de los elementos de Internet
      - Historia de Internet y el desarrollo web.
      - Evolución de los navegadores.
      - Cliente y Servidor
      - Protocolos. HTTP
    - Aplicación del lenguaje de marcado HTML para maquetar web.
      - Historia del HTML
      - Elementos de HTML: html, head, body y metadatos.
      - Elementos de separación del contenido, de texto y de multimedia
      - Tablas y formularios
      - Block vs Inline
      - Atributos de los elementos de HTML
      - Document Object Model (DOM)
    - Interpretación de los estándares y las directrices de accesibilidad web
      - Elementos semánticos

- Ayudas técnicas y modos de interacción
  - Introducción a las WCAG 2.1
  - Introducción a a11y project y en inclusive componentes
- Aplicación del lenguaje de hojas de estilo en cascada (CSS) para maquetar web
  - Historia del CSS
  - Anatomía del CSS
  - Especificidad
  - Pseudo-selectores y pseudo-elementos básicos
  - Estilos para textos y formularios
  - Colores y contraste
  - Modelo Box
  - Unidades (absolutas y relativas)
  - Posicionamiento
  - Layouts
  - Responsive web design
  - Flexbox
  - CSS Grid
  - Uso de transiciones
  - Uso de animaciones
- Análisis de diseños realizados con herramientas de diseño gráfico.
  - Archivos de diseño (provistos por Zeplin, AdobeXD, etc)
  - Diseño de prototipos con herramientas de diseño gráfico reproduciéndolo a través de código.
- Caracterización de los entornos, las fases y herramientas necesarias para realizar proyectos
  - Entornos, fases y documentación de un proyecto.
  - Metodologías de elicitación de requerimientos de usuario.
  - Planificación de tareas en proyectos de desarrollo.
  - Entornos colaborativos de desarrollo de software.
- Relación de los conocimientos de control de versiones y repositorios de código, como GitHub.
  - Introducción al Control de Versiones con Git
  - Sintaxis básica
  - GitFlow
  - GitHub
- Habilidades de gestión, personales y sociales
  - Demostración de iniciativa y visión estratégica para entender los elementos y las relaciones entre las partes de un proyecto web.
  - Capacidad para analizar los elementos relevantes de un proyecto web basado en HTML y CSS.
  - Mejora de las habilidades comunicativas con los clientes y el equipo de trabajo, utilizando herramientas de control de tareas y control de versiones de código.
  - Disposición para desarrollar las fases individuales de un proyecto autónomo basado en HTML y CSS, gestionando y planificando el tiempo.

- **Unidad 2. Javascript (es6).**

- Conocimientos / Capacidades cognitivas y prácticas
  - Caracterización de los fundamentos de la programación Front End: JavaScript.
    - Historia y evolución de JavaScript.
    - Variables (var, let y const)
    - Tipo de datos: Boolean, Number, String, Date, undefined, BigInt y Symbol.
    - Objetos
    - Valores especiales: null, NaN y Infinity.
    - Operadores y precedencia de operadores.
    - Expresiones.
    - Bloque de control del flujo, y control / tratamiento de errores
    - Scopes (incluyendo hoisting)
    - Funciones (first-class functions, funciones anidadas y clausuras, funciones anónimas, callback, arrow functions y this).
    - Gestión de eventos (event bubbling y captura).
    - JavaScript y DOM: cómo se relacionan HTML, CSS y JavaScript.
  - Aplicación de los elementos más avanzados de JavaScript:
    - Herencia basada en prototipos.
    - Herencia en ES6.
    - Programación funcional: map, filter y reduce.
    - Módulos y elementos introducidos en ES6: desestructurador, spread, etc.
  - Interpretación y reescritura de llamadas, datos y código AJAX.
    - Callback.
    - Promises.
    - Programación asíncrona, HTTP, API REST y Fetch API.
  - Reproducción con código, de prototipos realizados con herramientas de diseño gráfico.
    - Planificación de tareas en proyectos de desarrollo.
    - Escritura y depuración de código
    - Entornos colaborativos de desarrollo de software.
  - Manipulación de código de terceros
    - Comparación de códigos y soluciones existentes
    - Documentación técnica de terceros
  - Elaboración de documentación técnica y de usuario de lenguaje Javascript.
    - Documentación destinada al usuario final de una aplicación.
    - Documentación técnica destinada a programadores.
- Habilidades de gestión, personales y sociales
  - Concienciación de la importancia de la identificación, análisis y diseño de proyectos que requieran el uso de JavaScript.
  - Adquisición de actitudes positivas hacia el análisis de la información extraída de libros, Internet, etc. para solventar dudas y problemas que se presentan en la utilización de Javascript.
  - Mejora de las habilidades comunicativas con los clientes y el equipo de trabajo, utilizando herramientas de control de tareas, control de versiones

de código y generación de documentación.

- **Unidad 3. Framework de front-end (vue.js).**

- Conocimientos / Capacidades cognitivas y prácticas
  - Distinción de los frameworks de desarrollo Front End web.
    - Historia y evolución de los frameworks de desarrollo front-end web.
    - Comparativa de Vue.js, React y Angular.
  - Identificación de los elementos comunes que aparecen en los diversos frameworks:
    - Data binding
    - Reactividad
    - Templates
    - Componentes
    - Routing
    - State management
  - Utilización de un framework de front-end, en particular, Vue.js.
    - Data reactivity / two-way binding
    - Templates
    - Interpolación
    - Directivas
    - Conditional rendering
    - Events
    - Computed properties, Watchers
    - Componentes
    - Registration
    - Props
    - Custom events
    - Slots
    - Routing
    - Flux-like state management
  - Reproducción con un framework, de prototipos realizados con herramientas de diseño gráfico.
    - Planificación de tareas en proyectos de desarrollo.
    - Escritura y depuración de código
    - Entornos colaborativos de desarrollo de software.
  - Manipulación de código de terceros:
    - Comparación de códigos y soluciones existentes
    - Documentación técnica de terceros
  - Elaboración de documentación técnica y de usuario de un framework
    - Documentación destinada al usuario final de una aplicación.
    - Documentación técnica destinada a programadores.
- Habilidades de gestión, personales y sociales
  - Efectividad en la selección del framework JavaScript más adecuado para un problema dado.
  - Concienciación de la importancia de analizar la información extraída de la documentación y de los foros oficiales de un framework.

- Uso de habilidades de comunicación con los clientes y el equipo de trabajo, utilizando herramientas y estándares propios del framework para la codificación de los proyectos, el control de tareas, el control de versiones de código (p.ej. GitHub) y la generación de documentación.