

BOOTCAMP
Desarrollo Web
Híbrido
(224 horas)

Objetivo:

- Adquirir competencias para el desarrollo y diseño de sitios web atractivos para el usuario final.

Dirigido a:

- Profesionales involucrados en áreas de marketing, diseñador web y público en general interesado en desarrollar competencias en corto tiempo para emplearse de manera rápida en áreas de tecnología.

Reconocimiento:

- Al finalizar tu programa recibirás:
 - **Diploma Digital UVM con validez curricular y tecnología Blockchain con código QR y de verificación.**
 - **Certificado Internacional de Embiz Foundation.**
 - **Certificado de competencias laborales DC-3 de la STPS.**

¿Por qué UVM?

60 años de experiencia académica, más de 150 programas educativos y más de 180 programas de excelencia a nivel nacional.

Adquieres conocimientos y habilidades esenciales aplicables de manera inmediata a tu actividad profesional.

Los profesores que imparten las Certificaciones y Diplomados siguen un modelo de enseñanza con ejemplos reales, pues cada uno de ellos es experto y reconocido en su campo.

Flexibilidad educativa que te permite estudiar a tu ritmo, a cualquier hora y en cualquier lugar.

Los Diplomados y Certificaciones de UVM enriquecen tu CV y te posicionan como el mejor candidato.

Al estudiar el programa podrás:

Aprender los fundamentos de programación y bases de datos.



Dominar el desarrollo front-end.



Diseñar una web funcional y atractiva para el usuario final.



MÓDULOS

01 Fundamentos de programación y Extracción de datos mediante API.

1. Introducción a la programación
 - a. Qué es la programación
 - b. Lenguajes de programación comunes en el desarrollo web
 - c. Sintaxis básica
 - d. Variables y tipos de datos básicos
 - e. Operaciones básicas
2. Variables, tipos de datos y operadores
 - a. Variables: declaración, asignación y uso
 - b. Tipos de datos: números, cadenas y booleanos
 - c. Operadores aritméticos, de comparación y lógicos
 - d. Conversión de tipos de datos
3. Estructuras de control de flujo (condicionales y bucles)
 - a. Condicionales: if/else
 - b. Bucles: for, while
 - c. Anidamiento de estructuras de control de flujo
 - e. Break y continue
4. Funciones y modularidad
 - a. Definición y llamado de funciones
 - b. Parámetros y argumentos
 - c. Variables locales y globales
 - d. Creación y uso de módulos
 - e. Importación y exportación de módulos
5. Estructuras de datos básicas (arrays, objetos)
 - a. Arrays: creación, acceso y manipulación
 - b. Métodos de arrays: push, pop, shift, unshift, splice
 - c. Objetos: creación y acceso a propiedades
 - d. Métodos de objetos: keys, values, hasOwnProperty
6. Debugging y manejo de errores
 - a. Uso de la consola de desarrollador
 - b. Interpretación de mensajes de error
 - c. Depuración de código
 - d. Bloques try/catch
 - e. Manejo de excepciones
7. Extraer datos mediante API
 - a. Introducción a las API: Qué es una API
 - b. Beneficios de utilizar API en el desarrollo web
 - c. Ejemplos de API comunes
8. Métodos HTTP: GET, POST, PUT, DELETE
 - a. Qué son los métodos HTTP
 - b. Cómo funcionan los métodos GET, POST, PUT y DELETE
 - c. Utilización de los métodos HTTP para interactuar con una API
9. Tipos de API: REST, SOAP
 - a. Qué es una API REST y cómo funciona
 - b. Qué es una API SOAP y cómo funciona
 - c. Comparación entre API REST y SOAP
10. Autenticación de API: claves de API, OAuth
 - a. Qué es la autenticación de API
 - b. Cómo utilizar claves de API para autenticarse en una API
 - c. Cómo utilizar OAuth para autenticarse en una API
11. Uso de bibliotecas de JavaScript para trabajar con API
 - a. Qué son las bibliotecas de JavaScript y cómo funcionan
 - b. Ejemplos de bibliotecas de JavaScript para trabajar con API
 - c. Cómo utilizar una biblioteca de JavaScript para extraer y manipular datos de API

MÓDULOS

02 Creación de aplicaciones.

1. Introducción al patrón de diseño MVC

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

2. Roles y responsabilidades de cada componente (modelo, vista, controlador):

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

3. Ventajas del patrón MVC

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

4. Implementación de MVC en diferentes frameworks

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

5. Patrones de diseño adicionales relacionados con MVC:

5. Patrones de diseño adicionales relacionados con MVC:

5. Patrones de diseño adicionales relacionados con MVC:

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

6. Diseño de la arquitectura de la aplicación web utilizando MVC

6. Diseño de la arquitectura de la aplicación web utilizando MVC

- a. Qué es el patrón de diseño MVC
- b. Historia del patrón MVC
- c. Por qué utilizar el patrón MVC
- d. Diferentes implementaciones del patrón MVC

MÓDULOS

03 Desarrollo FrontEnd

1. HTML:

- a. Introducción a HTML
 - i. Qué es HTML y cómo funciona
 - ii. Sintaxis básica de HTML
 - iii. Estructura básica de un documento

HTML

- b. Elementos HTML
 - i. Elementos básicos de HTML (encabezados, párrafos, listas, etc.)
 - ii. Etiquetas de formato de texto (negrita, cursiva, etc.)
 - iii. Enlaces e imágenes
- c. Implementación HTML
 - i. Cómo implementar HTML en una página web
 - ii. Buenas prácticas para la organización del código HTML

2. CSS:

- a. Introducción a CSS
- b. Qué es CSS y cómo funciona
- c. Sintaxis básica de CSS
- d. Reglas CSS y selectores
- e. Elementos a CSS
- f. Propiedades básicas de CSS (color, tamaño de fuente, etc.)

tamaño de fuente, etc.)

- g. Diseño de cajas (margen, borde, relleno)
- h. Diseño de tablas y formularios
- i. Posicionamiento de CSS
- j. Posicionamiento de elementos en una página web utilizando CSS
- k. Flexbox y CSS Grid

3. JavaScript (comportamiento dinámico)

- a. Programación orientada a objetos
- b. Introducción a la programación orientada a objetos
- c. Creación de objetos y clases en JavaScript
- d. Encapsulamiento, herencia y polimorfismo

4. JQuery

- a. Introducción a jQuery
- b. Qué es jQuery y cómo funciona
- c. Selección de elementos y manipulación de DOM con jQuery
- d. Eventos en jQuery
- e. Manejo de eventos con jQuery
- f. Eventos de ratón y teclado en jQuery

MÓDULOS

04 Desarrollo BackEnd.

1. Python:

- a. Introducción a Python
 - i. Instalación de Python y configuración del entorno de trabajo
 - ii. Sintaxis básica de Python
 - iii. Variables, tipos de datos y operadores
- b. Control de flujo Python
 - i. Estructuras de control de flujo en Python (condicionales, bucles, etc.)
 - ii. Funciones y argumentos en Python
- c. Tipos de datos Python
 - i. Listas, tuplas y diccionarios en Python
- ci. Programación funcional en Python

2. Django:

- a. Introducción a Django
- b. Qué es Django y cómo funciona
- c. Configuración de Django y creación de una aplicación web
- d. Teoría URL's
 - i. Creación de URLs y vistas en Django
- di. Configuración de rutas en Django
- dii. Modelos Django
 - i. Creación y configuración de modelos en Django

ii. Interacción con bases de datos utilizando Django ORM

g. Autenticación Django

- i. Creación de usuarios y autenticación en Django
- ii. Autenticación basada en tokens en Django

3. MySQL:

- a. Introducción a MySQL
 - i. Qué es MySQL y cómo funciona
 - ii. Configuración de MySQL y creación de una base de datos
- b. Implementación de MySQL
 - i. Creación de tablas y relaciones en MySQL
 - ii. Interacción con MySQL utilizando Python y Django

05 Configuración del entorno de trabajo.

1. Dependencias

- a. Instalación y configuración de dependencias para diferentes lenguajes de programación y frameworks
- b. Uso de administradores de paquetes como npm y pip
- c. Manejo de versiones de dependencias y resolución de conflictos

2. Línea de comandos

- a. Introducción a la línea de comandos
- b. Navegación y manipulación de archivos y directorios
- c. Comandos básicos de la línea de comandos para el desarrollo web

3. Git

- a. Introducción a Git
- b. Instalación y configuración de Git
- c. Creación y gestión de repositorios locales y remotos
- d. Trabajo en equipo y colaboración utilizando Git

4. GitHub

- a. Introducción a GitHub
- b. Creación y gestión de repositorios en GitHub
- c. Trabajo en equipo y colaboración utilizando GitHub
- d. Uso de herramientas de GitHub como pull requests, issues y proyectos

06 Desarrollo de aplicación web.

1. Uso de la herramienta de Figma

- a. Introducción a Figma
 - i. Interfaz de usuario y herramientas de Figma
 - ii. Creación y organización de proyectos en Figma
 - iii. Importación y exportación de diseños en Figma
- b. Diseño de interfaces de usuario y experiencia de usuario en Figma
 - i. Diseño de wireframes y mockups en Figma
 - ii. Diseño de layouts y estilos en Figma
 - iii. Creación de prototipos interactivos en Figma
- c. Colaboración en equipo utilizando Figma
 - i. Trabajo en equipo y colaboración en proyectos de Figma
 - ii. Comentarios y revisiones en Figma
 - iii. Uso de plugins y recursos en Figma

2. Uso de la herramienta Jira

- a. Introducción a Jira
 - i. Interfaz de usuario y herramientas de Jira
 - ii. Creación y configuración de proyectos en Jira
 - iii. Roles y permisos en Jira
- b. Creación y gestión de proyectos en Jira
 - i. Creación y asignación de tareas en Jira
 - ii. Gestión de sprints y entregas en Jira
 - iii. Seguimiento y reporte de progreso en Jira
- c. Asignación de tareas y seguimiento de proyectos en Jira
 - i. Priorización y asignación de tareas en Jira
 - ii. Comunicación y colaboración en equipo utilizando Jira
 - iii. Uso de herramientas de análisis y métricas en Jira

Beneficios de la modalidad

Clases en vivo, actividades interactivas y casos prácticos. Puedes interactuar con profesores y otros alumnos para tener una experiencia más enriquecedora.

Networking. Tienes la oportunidad de construir una red de contactos profesionales con otras personas que tienen intereses similares o se desempeñan en el mismo ámbito.

Estudia a tu ritmo. Consulta todas las sesiones grabadas en el horario que más te convenga.

Soporte técnico. Cuentas con atención técnica en todo momento para ayudarte a solucionar cualquier problema que se presente.

Nota: Si no asistes a las sesiones en vivo con el profesor en las fechas y horarios establecidos, tendrás 30 días naturales para ver completa la grabación de la clase en Teams® y realizar la actividad asignada para que acredites el módulo.

SÉ PARTE DE LA UVM



@uvmmx



uvm



@uvmmx



uvm.mx