

BOOTCAMP EN DESARROLLO
WEB Y UX/UI
EN LÍNEA
(9 meses)

DIPLOMADO EJECUTIVO

¡Transforma tu perfil profesional y destaca en el mercado laboral!

- Obtén tres insignias digitales que validan tus habilidades.
- Credenciales verificables con tecnología blockchain, que fortalecen tu perfil profesional.
- Accede a +20 programas en diversas áreas de estudio, diseñados para responder a las demandas actuales de los empleadores



Reconocimiento Diplomado Ejecutivo

Obtén un Constancia digital UVM del programa con tecnología blockchain y código de verificación



Reconocimiento Curso IA

Adquiere nuevos conocimientos y desarrolla tus habilidades para el uso práctico y estratégico de las herramientas de IA en tu entorno profesional.



Constancia DC-3 por STPS

Documento avalado por la Secretaría de Trabajo y Previsión Social (STPS) que acredita las habilidades y competencias adquiridas.

Objetivo:

- Conviértete en un profesional integral dominando el *front end*, *back end* y el diseño centrado en el usuario. El Bootcamp en Desarrollo Web y UX/UI te prepara con herramientas técnicas y creativas para diseñar y construir aplicaciones *web* innovadoras y funcionales. Aprende desde cero o fortalece tu perfil como *full stack developer* combinando lenguajes de programación, bases de datos, principios de interfaz de usuario y experiencia de usuario.

Dirigido a:

- Dirigido a profesionistas que desean incursionar, actualizarse o evolucionar dentro del sector tecnológico, independientemente de su formación académica original. Está especialmente diseñado para personas con al menos 6 meses de experiencia profesional en cualquier sector.

Reconocimiento:

- Al finalizar tu programa recibirás:
 - **Diploma Digital con validez curricular y tecnología Blockchain** con código QR y de verificación.
 - Dos certificados internacionales digitales **Blockchain de Embiz Foundation: Certificado Digital Internacional de UX/UI y Certificado Digital Internacional de Desarrollo Web.**
 - Dos **certificados Ibaktor DC3** de competencias laborales: **Certificado Ibaktor DC-3 de Competencias Laborales de UX/UI y Certificado Ibaktor DC-3 de Competencias Laborales de Desarrollo Web.**

¿Por qué UVM?

Tenemos **más de 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** que puedes **aplicar de inmediato** en tu **actividad profesional**.

Los **profesores** que imparten las **Certificaciones y Diplomados** son **expertos reconocidos** en sus campos.

Tienes **flexibilidad educativa** que te permite **estudiar a tu ritmo**, a **cualquier hora** y en **cualquier lugar**.

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

Al estudiar el programa podrás:

Crear interfaces *web* accesibles, funcionales y centradas en el usuario, aplicando los principios de diseño UX/UI en soluciones digitales reales. Dominarás tanto la programación *front-end* como *back-end*, permitiéndote desarrollar aplicaciones *web* completas con una sólida estructura técnica y visual.



BLOQUES DE APRENDIZAJE

Formar bases en programación y lógica para desarrollo *web*, aplicando ciencia de datos al diseño UX/UI mediante análisis estadístico, visualización y *machine learning* para optimizar la toma de decisiones.

Fundamentos de programación y dirección de proyectos

Aprender desarrollo *web front end*: HTML para estructura, CSS y *frameworks* (Bootstrap, TailwindCSS) para diseño y posicionamiento, y JavaScript con POO para implementar comportamiento dinámico en sitios *web*.

Desarrollo Front End

Usar herramientas para gestión de dependencias, control de versiones y colaboración *web*: línea de comandos para archivos y GitHub para crear, gestionar y trabajar en repositorios, optimizando el trabajo en equipo profesional.

Configuración del entorno de trabajo y GitHub

Entender el patrón MVC en desarrollo *web*, sus componentes, ventajas y aplicación en distintos *frameworks*, junto a otros patrones de diseño, para crear arquitecturas eficientes y estructuradas en aplicaciones modernas.

Creación de aplicaciones

Aprender fundamentos de APIs: definición, tipos y métodos HTTP, autenticación con claves y OAuth, uso de bibliotecas JS y Node.js para crear y gestionar APIs con *middleware*, fortaleciendo la integración y desarrollo *back end*.

Desarrollo Back End

BLOQUES DE APRENDIZAJE

Desarrollar habilidades en Python y Django, desde instalación y configuración hasta crear aplicaciones *web* completas, manejando estructuras de control, tipos de datos y programación funcional en Python.

Python

Desarrollar habilidades en investigación cualitativa y cuantitativa para entender necesidades y contextos de usuarios, identificando *insights* clave que definan objetivos, alcance y criterios de éxito en proyectos UX/UI.

Métodos UX/UI de investigación y definición del proyecto

Aprender a estructurar arquitectura de información UX/UI con mapas de sitio, flujos y visualizaciones, creando *wireframes* y prototipos funcionales según principios de diseño, usando herramientas como Figma o Sketch.

Arquitectura UX/UI de la información y prototipado

Aprender a crear soluciones UX/UI accesibles y atractivas con HTML, CSS y JavaScript, aplicando diseño *responsive* y *mobile*, generando contenido visual y escrito, documentando procesos y colaborando con equipos multidisciplinarios.

Implementación y Design Content UX/UI

BLOQUES DE APRENDIZAJE

Fundamentos de programación y dirección de proyectos

1. Fundamentos de programación
 - a. Introducción a la programación
 - b. Variables, tipos de datos y operadores
 - c. Estructuras de control de flujo (condicionales y bucles)
 - d. Funciones y modularidad
 - e. Estructuras de datos básicas (*arrays*, objetos)
 - f. *Debugging* y manejo de errores
2. Fundamentos de *data science* y UX/UI
 - a. Introducción a la *data science*
 - b. Estadística básica
 - c. Análisis exploratorio de datos
 - d. Aprendizaje automático y minería de datos
 - e. Visualización de datos
3. Fundamentos de ciberseguridad y UX/UI
 - a. Introducción a la ciberseguridad
 - b. Amenazas y vulnerabilidades de seguridad
 - c. Protección de datos y privacidad
 - d. Seguridad en el diseño de UX/UI
4. Fundamentos de *tech business*
 - a. Introducción al *tech business*
 - b. Modelos de negocio en la industria tecnológica
 - c. Estrategias de *marketing* y ventas
 - d. Análisis financiero y gestión de presupuesto
 - e. Innovación en *tech business*
5. Fundamentos de Diseño UX/UI
 - a. Introducción a la UX/UI
 - b. Diseño centrado en el usuario
 - c. Principios de diseño de UX/UI
 - d. Herramientas de diseño de UX/UI
 - e. Evaluación y mejora de la UX/UI

Desarrollo *Front End*

1. HTML
 - a. Introducción a HTML
 - b. Elementos HTML
 - c. Implementación 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.)
 - 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
4. Librerías de diseño
 - a. Bootstrap
 - b. TailwindCSS
5. *Frameworks Front End*
 - a. *React*

Configuración del entorno de trabajo y GitHub

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

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)
3. Ventajas del patrón MVC
4. Implementación de MVC en diferentes *frameworks*
5. Patrones de diseño adicionales relacionados con MVC
6. Diseño de la arquitectura de la aplicación *web* utilizando MVC

Desarrollo *Back End*

1. Introducción a las API
 - a. ¿Qué es una API?
 - b. Beneficios de utilizar API en el desarrollo *web*
 - c. Ejemplos de API comunes
2. 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
3. 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
4. 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?
5. 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. Comparación entre API REST y SOAP
6. Node.js
 - a. ¿Qué es una API REST y cómo funciona?
 - b. ¿Qué es una API SOAP y cómo funciona?
 - c. ¿Cómo utilizar una biblioteca de JavaScript para extraer y manipular datos de API?

Módulo: Python

1. Python
 - a. Introducción a Python
 - b. Control de flujo Python
 - c. Tipos de datos 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
 - e. Configuración de rutas en Django
 - f. Modelos Django
 - g. Autenticación Django
3. MySQL
 - a. Introducción a MySQL
 - b. Implementación de MySQL

Métodos UX/UI de investigación y definición del proyecto

1. Métodos de investigación cualitativos
 - a. Cómo utilizar técnicas de investigación cualitativas, como entrevistas individuales, grupos de discusión y observación, para comprender las necesidades y expectativas de los usuarios
 - b. Cómo analizar y presentar los datos recopilados a través de técnicas de investigación cualitativas
2. Métodos de investigación cuantitativos
 - a. Cómo utilizar técnicas de investigación cuantitativas, como encuestas y análisis de datos, para recopilar información sobre la experiencia del usuario
 - b. Cómo analizar y presentar los datos recopilados a través de técnicas de investigación cuantitativas
3. Descubrimiento de *insights*
 - a. Cómo utilizar técnicas de investigación para descubrir *insights* valiosos sobre los usuarios, sus necesidades y su comportamiento
 - b. Cómo transformar los *insights* en oportunidades de diseño de UX/UI
4. Entendimiento de usuario final
 - a. Cómo comprender a los usuarios finales y sus necesidades específicas
 - b. Cómo utilizar técnicas de investigación para comprender los contextos de los usuarios y sus necesidades en diferentes etapas del ciclo de vida del producto
5. Definición del proyecto, objetivos y alcance
 - a. Cómo definir los objetivos del proyecto y el alcance de la solución de UX/UI
 - b. Cómo identificar y priorizar los requisitos y funcionalidades clave de la solución de UX/UI
6. Identificación de *stakeholders*
 - a. Cómo identificar a los *stakeholders* clave y entender sus necesidades y expectativas
 - b. Cómo involucrar a los *stakeholders* en el proceso de diseño de UX/UI
7. Creación de *user personas* y *user scenarios*
 - a. Cómo crear *user personas* y *user scenarios* para entender a los usuarios y sus necesidades
 - b. Cómo utilizar los *user personas* y *user scenarios* para informar el diseño de UX/UI y la toma de decisiones
8. Identificación y definición de los puntos de dolor del usuario
 - a. Cómo identificar los puntos de dolor del usuario y las oportunidades de mejora en la solución de UX/UI
 - b. Cómo definir soluciones efectivas para abordar los puntos de dolor del usuario
9. Establecimiento de criterios de éxito
 - a. Cómo establecer criterios de éxito claros y medibles para la solución de UX/UI
 - b. Cómo utilizar los criterios de éxito para evaluar el rendimiento y la efectividad de la solución de UX/UI

Arquitectura UX/UI de la información y prototipado

1. Mapeo del sitio
 - a. ¿Cómo crear un mapa del sitio efectivo que muestre la estructura de la solución de UX/UI?
 - b. ¿Cómo utilizar el mapa del sitio para guiar el diseño de la navegación y la organización de la información?
2. Diagramas de flujo
 - a. ¿Cómo crear diagramas de flujo que muestren cómo los usuarios interactúan con la solución de UX/UI?
 - b. ¿Cómo utilizar los diagramas de flujo para guiar el diseño de la interacción y la navegación?
3. Visualización de estructura
 - a. ¿Cómo visualizar la estructura de la solución de UX/UI de manera efectiva para ayudar a los usuarios a comprender y navegar la información?
 - b. ¿Cómo utilizar la visualización de estructura para guiar el diseño de la navegación y la organización de la información?
4. Wireframes
 - a. ¿Cómo crear wireframes que muestren el diseño de la interfaz de usuario de la solución de UX/UI?
 - b. ¿Cómo utilizar los wireframes para guiar el diseño de la interacción y la navegación?
5. Tree testing
 - a. ¿Cómo realizar pruebas de tree testing para evaluar la efectividad de la navegación y la organización de la información de la solución de UX/UI?
6. Card sorting
 - a. ¿Cómo realizar card sorting para evaluar cómo los usuarios organizan y etiquetan la información de la solución de UX/UI?
 - b. ¿Cómo utilizar los resultados del card sorting para mejorar la arquitectura de la información de la solución de UX/UI?
7. Prototipado y principios de diseño
 - a. ¿Cómo utilizar principios de diseño efectivos, como la elección de colores y tipografías, para crear interfaces de usuario atractivas y efectivas?
 - b. ¿Cómo aplicar principios de diseño a través de patrones de diseño comunes en soluciones de UX/UI?
8. Sistemas de composición
 - a. ¿Cómo utilizar sistemas de composición, como grids y estructuras de diseño, para crear interfaces de usuario efectivas y coherentes?
 - b. ¿Cómo aplicar sistemas de composición a través de patrones de diseño comunes en soluciones de UX/UI?
9. Herramientas de diseño
 - a. ¿Cómo utilizar herramientas de diseño, como Figma o Sketch, para crear wireframes y prototipos efectivos de soluciones de UX/UI?
 - b. ¿Cómo utilizar herramientas de diseño para colaborar con otros miembros del equipo de diseño de UX/UI?
10. Patrones web/app
 - a. ¿Cómo utilizar patrones web/app comunes, como formularios de registro y carruseles de imágenes, para crear soluciones de UX/UI efectivas y satisfactorias para los usuarios?
 - b. ¿Cómo identificar y aplicar patrones web/app efectivos en soluciones de UX/UI?
11. Prototipo en uso
 - a. ¿Cómo crear prototipos de soluciones de UX/UI para evaluar la efectividad del diseño de la interfaz de usuario?
 - b. ¿Cómo utilizar prototipos para recibir retroalimentación de los usuarios y mejorar el diseño de la solución de UX/UI?
 - c. ¿Cómo utilizar los resultados de las pruebas de tree testing para mejorar la arquitectura de la información de la solución de UX/UI?

Módulo: Implementación y design content UX/UI

1. Fundamentos de HTML
 - a. ¿Cómo utilizar HTML para crear la estructura y el contenido de una solución de UX/UI?
 - b. ¿Cómo aplicar las mejores prácticas de HTML para mejorar la accesibilidad y la usabilidad de la solución de UX/UI?
2. Fundamentos de CSS
 - a. ¿Cómo utilizar CSS para dar estilo y diseñar la apariencia de una solución de UX/UI?
 - b. ¿Cómo aplicar las mejores prácticas de CSS para mejorar la accesibilidad y la usabilidad de la solución de UX/UI?
3. Fundamentos de JavaScript
 - a. ¿Cómo utilizar JavaScript para agregar interactividad y dinamismo a una solución de UX/UI?
 - b. ¿Cómo aplicar las mejores prácticas de JavaScript para mejorar la accesibilidad y la usabilidad de la solución de UX/UI?
4. Diseño responsive y mobile
 - a. ¿Cómo diseñar soluciones de UX/UI que sean efectivas y satisfactorias para los usuarios en dispositivos móviles y pantallas de diferentes tamaños?
 - b. ¿Cómo aplicar las mejores prácticas de diseño responsive y mobile en soluciones de UX/UI?
5. Introducción al Design Content
 - a. ¿Qué es Design Content y por qué es importante para el diseño de UX/UI?
 - b. ¿Cómo el contenido puede influir en la experiencia del usuario y en la percepción de la marca?
6. Creación de contenido visual
 - a. ¿Cómo crear contenido visual que sea atractivo y relevante para el usuario?
 - b. ¿Cómo diseñar gráficos, imágenes y videos que apoyen la experiencia de usuario y la marca?
 - c. ¿Cómo utilizar herramientas y *software* de diseño gráfico para crear contenido visual?
7. Creación de contenido escrito
 - a. ¿Cómo escribir contenido efectivo que sea relevante y atractivo para el usuario?
 - b. ¿Cómo adaptar el tono y el estilo del contenido a la marca y al público objetivo?
 - c. ¿Cómo utilizar herramientas y *software* de edición de texto para crear contenido escrito?
8. Documentación de procesos
 - a. ¿Cómo documentar el proceso de creación de contenido para garantizar transparencia y una comunicación efectiva dentro del equipo de UX/UI?
 - b. ¿Cómo utilizar herramientas y *software* de gestión de proyectos para mantener un seguimiento eficiente de la documentación de procesos?
9. Colaboración con otros profesionales de UX/UI
 - a. ¿Cómo trabajar con diseñadores, desarrolladores y otros miembros del equipo para crear una experiencia de usuario coherente y efectiva?
 - b. ¿Cómo proporcionar comentarios y retroalimentación efectiva sobre el contenido creado por otros miembros del equipo?

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.

Asesoría y acompañamiento. Tienes un tutor que te brindará apoyo a través de enlaces en vivo, chat o WhatsApp para resolver cualquier duda.

SÉ PARTE DE LA UVM



@uvmmx



uvm



@uvmmx



uvm.mx