

Bootcamp:

Desarrollo Web

Programa *online* de 224 horas
(27 semanas)

¿Qué es un Bootcamp?

Una experiencia *learning by doing* significativa de aprendizaje, en la cual los participantes adquieren conocimientos y desarrollan competencias tecnológicas de aplicación inmediata, de una manera práctica y efectiva a través de sesiones sincrónicas semanales interactivas en línea y actividades de aprendizaje en la plataforma, guiados por profesores expertos y mediante modelo educativo integral.

Características del programa:

Modalidad Bootcamp interactivo de 224 horas:

- 104 horas de clases de clases sincrónicas interactivas en línea (clases en vivo).
- 120 horas actividades asincrónicas.

Las 104 horas de clases sincrónicas interactivas en línea (clases en vivo) se imparten en un período de 27 semanas:

- Fase de Preparación
- Semana 1: 2 horas de clases
- Semana 2: 2 horas de clases
- Fase de Bootcamp
- Semanas de la 3 a la 27 (total 25 semanas): 4 horas de clase por semana

Las 120 horas de actividades asincrónicas consisten en:

- Clases pregrabadas
- Ejercicios
- Foros
- Casos prácticos
- Exámenes de certificación

Objetivo:

El participante aprenderá fundamentos de programación y bases de datos para dominar desarrollo *frontend*, con la finalidad de diseñar una web funcional y atractiva para el usuario final.

Los participantes se enfrentarán a retos en un concepto diferente, Bootcamps, en los cuales se llevará a cabo la participación de cada integrante con su profesor en una dinámica y experiencia única para mostrar sus conocimientos aplicados para ver distintas perspectivas.

Diferenciación:

- Incluye curso de nivelación/preparación.
- Certificación internacional con tecnología *Blockchain*, de *Embiz Foundation* (www.embizfoundation.org).
- Certificado de competencias laborales DC-3, de la Secretaría del Trabajo y Previsión Social (STPS).
- Experiencia de Bootcamps, en donde el aspirante se enfrentará a retos prácticos con su profesor y compañeros de grupo con la finalidad de conocer distintas perspectivas y métodos, utilizando herramientas valiosas.
- Valor más alto del mercado y en horarios adecuados para nuestros alumnos.

Modelo educativo:

El programa estará distribuido de la siguiente manera:

- Acceso a plataforma
- Profesores expertos
- Clases sincrónicas (en vivo)
- Explicaciones prácticas
- Ejercicios guiados por el profesor
- Retos en equipos e individuales
- Sesiones *live code*, programación (interactiva - en vivo)
- Clases pregrabadas
- Ejercicios
- Foros de interacción
- Casos prácticos
- Exámenes de certificación

Beneficios:

Al finalizar el programa, el participante obtendrá, además de su diploma de UVM, el Certificado Digital Internacional avalado por *Embiz Foundation*, así como el Certificado DC-3 de Competencias Laborales, avalado por la STPS.

¿Qué herramientas de trabajo necesito para cursar un BootCamp en UVM?

1. Contar con computadora de escritorio o laptop y acceso a internet a través de wifi.
2. La computadora debe tener microprocesador Ryzen o Intel i5 o superior, Gen 10 o superior, 8 GB de RAM y 100 GB libres de disco duro.
3. Abrir cuentas para acceso a software libre y en la nube, acceder a un repositorio de documentos, código y proyectos, que el profesor informará en la primera clase.

TEMARIO

01 Fundamentos de programación

a) Introducción a la programación

1. Qué es la programación.
2. Lenguajes de programación comunes en el desarrollo web.
3. Sintaxis básica.
4. Variables y tipos de datos básicos.
5. Operaciones básicas.

b) Variables, tipos de datos y operadores

1. Variables: declaración, asignación y uso.
2. Tipos de datos: números, cadenas y booleanos.
3. Operadores aritméticos, de comparación y lógicos.
4. Conversión de tipos de datos.

c) Estructuras de control de flujo (condicionales y bucles)

1. Condicionales: *if/else*.
2. Bucles: *for, while*.
3. Anidamiento de estructuras de control de flujo.
4. *Break* y *continue*.

d) Funciones y modularidad

1. Definición y llamado de funciones.
2. Parámetros y argumentos.
3. Variables locales y globales.
4. Creación y uso de módulos.
5. Importación y exportación de módulos.

e) Estructuras de datos básicas (arrays, objetos)

1. Arrays: creación, acceso y manipulación.
2. Métodos de arrays: *push, pop, shift, unshift, splice*.
3. Objetos: creación y acceso a propiedades.
4. Métodos de objetos: *keys, values, hasOwnProperty*.

d) Debugging y manejo de errores

1. Uso de la consola de desarrollador.
2. Interpretación de mensajes de error.
3. Depuración de código.
4. Bloques *try/catch*.
5. Manejo de excepciones.

02 Extraer datos mediante API

a) Introducción a las API

1. Qué es una API.
2. Beneficios de utilizar API en el desarrollo web.
3. Ejemplos de API comunes.

b) Métodos HTTP: GET, POST, PUT, DELETE

1. Qué son los métodos HTTP.
2. Cómo funcionan los métodos *GET*, *POST*, *PUT* y *DELETE*.
3. Utilización de los métodos HTTP para interactuar con una API.

c) Tipos de API: REST, SOAP

1. Qué es una API REST y cómo funciona.
2. Qué es una API SOAP y cómo funciona.
3. Comparación entre API REST y SOAP.

d) Autenticación de API: claves de API, OAuth

1. Qué es la autenticación de API.
2. Cómo utilizar claves de API para autenticarse en una API.
3. Cómo utilizar OAuth para autenticarse en una API.

e) Uso de bibliotecas de JavaScript para trabajar con API

1. Qué son las bibliotecas de JavaScript y cómo funcionan.
2. Ejemplos de bibliotecas de JavaScript para trabajar con API.
3. Cómo utilizar una biblioteca de JavaScript para extraer y manipular datos de API.

03 Creación de aplicaciones

a) Introducción al patrón de diseño MVC

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

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

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

c) Ventajas del patrón MVC

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

d) Implementación de MVC en diferentes frameworks

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

e) Patrones de diseño adicionales relacionados con MVC:

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

f) Diseño de la arquitectura de la aplicación web utilizando MVC

1. Qué es el patrón de diseño MVC.
2. Historia del patrón MVC.
3. Por qué utilizar el patrón MVC.
4. Diferentes implementaciones del patrón MVC.

04 Desarrollo FrontEnd

a) HTML:

1. Introducción a HTML
 - Qué es HTML y cómo funciona.
 - Sintaxis básica de HTML.
 - Estructura básica de un documento HTML.
2. Elementos HTML
 - Elementos básicos de HTML (encabezados, párrafos, listas, etcétera).
 - Etiquetas de formato de texto (negrita, cursiva, etcétera).
 - Enlaces e imágenes.
3. Implementación HTML
 - Cómo implementar HTML en una página web.
 - Buenas prácticas para la organización del código HTML.

b) CSS:

1. Introducción a CSS
2. Qué es CSS y cómo funciona.
3. Sintaxis básica de CSS.
4. Reglas CSS y selectores.
5. Elementos a CSS.
6. Propiedades básicas de CSS (color, tamaño de fuente, etcétera).
7. Diseño de cajas (margen, borde, relleno).
8. Diseño de tablas y formularios.
9. Posicionamiento de CSS.
10. Posicionamiento de elementos en una página web utilizando CSS.
11. Flexbox y CSS Grid.

c) JavaScript (comportamiento dinámico)

1. Programación orientada a objetos.
2. Introducción a la programación orientada a objetos.
3. Creación de objetos y clases en JavaScript.
4. Encapsulamiento, herencia y polimorfismo.

d) JQuery

1. Introducción a jQuery.
2. Qué es jQuery y cómo funciona.
3. Selección de elementos y manipulación de DOM con jQuery.
4. Eventos en jQuery.
5. Manejo de eventos con jQuery.
6. Eventos de ratón y teclado en jQuery.

05 Desarrollo BackEnd

a) Python:

1. Introducción a Python.
 - Instalación de Python y configuración del entorno de trabajo.
 - Sintaxis básica de Python.
 - Variables, tipos de datos y operadores.
2. Control de flujo Python.
 - Estructuras de control de flujo en Python (condicionales, bucles, etcétera).
 - Funciones y argumentos en Python.
3. Tipos de datos Python.
 - Listas, tuplas y diccionarios en Python.
4. Programación funcional en Python.

b) Django:

1. Introducción a Django.
2. Qué es Django y cómo funciona.
3. Configuración de Django y creación de una aplicación web.
4. Teoría URL's.
 - Creación de URLs y vistas en Django.
 - Configuración de rutas en Django.
5. Modelos Django.
 - Creación y configuración de modelos en Django.
 - Interacción con bases de datos utilizando Django ORM.
6. Autenticación Django.
 - Creación de usuarios y autenticación en Django.
 - Autenticación basada en tokens en Django.

c) MySQL:

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

06 Configuración del entorno de trabajo

a) Dependencias

1. Instalación y configuración de dependencias para diferentes lenguajes de programación y *frameworks*.
2. Uso de administradores de paquetes como *npm* y *pip*.
3. Manejo de versiones de dependencias y resolución de conflictos.

b) Línea de comandos

1. Introducción a la línea de comandos.
2. Navegación y manipulación de archivos y directorios.
3. Comandos básicos de la línea de comandos para el desarrollo web.

c) Git

1. Introducción a Git.
2. Instalación y configuración de Git.
3. Creación y gestión de repositorios locales y remotos.
4. Trabajo en equipo y colaboración utilizando Git.

d) GitHub

1. Introducción a GitHub.
2. Creación y gestión de repositorios en GitHub.
3. Trabajo en equipo y colaboración utilizando GitHub.
4. Uso de herramientas de GitHub como *pull requests*, *issues* y proyectos.

07 Desarrollo de aplicación web

a) Uso de la herramienta de Figma

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

b) Uso de la herramienta Jira

1. Introducción a Jira.
 - Interfaz de usuario y herramientas de Jira.
 - Creación y configuración de proyectos en Jira.
 - Roles y permisos en Jira.
2. Creación y gestión de proyectos en Jira.
 - Creación y asignación de tareas en Jira.
 - Gestión de *sprints* y entregas en Jira.
 - Seguimiento y reporte de progreso en Jira.
3. Asignación de tareas y seguimiento de proyectos en Jira.
 - Priorización y asignación de tareas en Jira.
 - Comunicación y colaboración en equipo utilizando Jira.
 - Uso de herramientas de análisis y métricas en Jira.

08 Proyecto real

a) Crear aplicación

1. Identificación y definición de requerimientos del proyecto.
2. Diseño y arquitectura de la aplicación.
3. Desarrollo de la aplicación utilizando tecnologías y herramientas aprendidas en el Bootcamp.

b) Hacer integraciones Front y BackEnd

1. Integración del FrontEnd y el BackEnd de la aplicación.
2. Diseño y desarrollo de APIs para la integración entre el FrontEnd y el BackEnd.
3. Pruebas y depuración de la integración entre el FrontEnd y el BackEnd.

c) Sacar a producción

1. Selección y configuración de un servidor de producción.
2. Configuración de la aplicación para la producción.
3. Despliegue de la aplicación en el servidor de producción.

d) Lanzamiento con dominio

1. Selección y registro de un nombre de dominio para la aplicación.
2. Configuración de DNS y apuntamiento del nombre de dominio al servidor de producción.
3. Pruebas finales y lanzamiento de la aplicación al público.

Primer CV corto

Profesor:

Alejandro Anaya

Experiencia Profesional:

Amplia experiencia de más de 30 años en diversas áreas de Tecnologías de Información, como: Infraestructura, Telecomunicaciones, Seguridad, Desarrollo de Sistemas, Administración, Planeación, Control de Proyectos, Investigación y Desarrollo de Tecnologías, ha trabajado en la formación de equipos multidisciplinarios con el objetivo de integrar grupos resilientes con una amplia autonomía para el trabajo remoto.

Tiene perfecto conocimiento de normas y estándares de las organizaciones que deben cumplir con prácticas de Gobierno Corporativo. Además, ha colaborado en la obtención de certificaciones internacionales (ISO 20000, ISO 9000 e ISO 27000) en organizaciones que deben operar bajo marcos normativos y regulaciones de terceros.

Acostumbrado a la adopción de estándares y la creación de metodologías basadas en Sistemas de Gestión. En el aspecto de inversiones, posee una fuerte disciplina en materia de control de costos sin poner en riesgo la operación; reducción de costos, mediante la implantación de tecnologías basadas en *software* libre; sobre tecnologías licenciadas, supervisión para mantener términos de lo justo necesario.

Experiencia en la construcción de sinergias entre diferentes áreas para llevar a cabo nuevas implantaciones de infraestructura tecnológica, priorizando las necesidades del negocio.

Formación Académica

- Más de 500 horas de formación en cursos de especialización práctica.
- Licenciatura en Sistemas de Computación Administrativa, en la Universidad del Valle de México.

Reconocimientos

Promociones y logros importantes en diferentes posiciones en el ámbito corporativo.

Segundo CV corto

Profesor:

Adrián Hernández del Valle

Experiencia Profesional:

Profesional con décadas de experiencia en Ciencia de Datos y Finanzas. Pionero en Latinoamérica en el tema de inteligencia artificial. Ha gestionado varios proyectos de ciencia y ha colaborado con corporaciones y agencias gubernamentales para impulsar decisiones de negocio basadas en analítica avanzada. Domina el uso de lenguajes de programación estadística para la Ciencia de Datos. Además, es conferencista, escritor e investigador.

Reconocido profesor, impartió clases de economía en: 1) la Universidad Carlos III de Madrid, en España; 2) The Elliott School of International Affairs, de George Washington University, en Estados Unidos. Actualmente es profesor en la Escuela Superior de Economía, del Instituto Politécnico Nacional (ESE-IPN); la Universidad Panamericana (UP); y el Instituto Tecnológico Autónomo de México (ITAM).

Formación Académica

- Máster en Inteligencia Artificial, OBS Business School y Universidad Politécnica de Cataluña, España. 2020-2021.
- Máster en Estadística, Columbia University, Estados Unidos. Mayo 2007.
- MEDEX, Instituto Panamericano de Alta Dirección de Empresa (IPADE), México 2001-2003.
- Maestría en Ciencias en Economía; Especialidad en Economía Financiera, ESE-IPN, México.

Reconocimientos

Publicó veinticuatro trabajos de investigación y un libro. Fue líder de varios proyectos de Ciencia de Datos. Dirigió cuatro tesis de doctorado y 10 tesis de maestría en Economía.

Beneficios de estudiar un diplomado con Modelo Educativo Ibaktor

Obtienes dos certificados:

- Certificado Internacional digital de alta seguridad y encriptación, con examen de certificación, incluido en el costo de tu diplomado.
- Certificado DC-3 de la STPS.

Temas actualizados y de vanguardia:

Con gran capacidad de actualización y reinención al ser de una duración más corta que otros posgrados, un diplomado te ofrece una capacitación enfocada en temas relevantes y de alta demanda para el mercado laboral.

Capitaliza lo aprendido:

El alto enfoque práctico y estratégico de un Diplomado hace que cada módulo sea aplicable desde el primer día 1 en tus actividades profesionales y desarrollo personal.

Mejora tus oportunidades laborales:

Enriquece tu CV especializándote y posíciónate como el mejor candidato.

Networking:

No solo compartirás salón de clases con buenos compañeros, también con excelentes profesionistas con los que podrás intercambiar puntos de vista, *tips* y oportunidades de negocio.

Profesores con más 15 años en experiencia profesional:

Toma clases de la mano de expertos en su disciplina con amplia experiencia compartiendo su conocimiento y trabajando en las mejores empresas nacionales e internacionales.

Duración:

La duración del diplomado es de 6 meses, así podrás aplicar lo aprendido muy rápidamente y seguir creciendo profesionalmente.

Diploma:

Todos nuestros Diplomados y Certificaciones tienen validez curricular.

Beneficios del Modelo Educativo Ibaktor

Clases pregrabadas y en vivo:

Estudia a tu ritmo, con material de gran calidad, puedes consultar todas las sesiones en el horario que más te convenga. Todas las clases en vivo se graban para tu comodidad.

Experiencias de aprendizaje *online*:

Foros.

Juegos.

Ejercicio y herramientas para aplicarlas en tu trabajo o proyectos.

Casos prácticos.

Acceso a materiales complementarios.

Contenido siempre disponible:

Podrás consultar y descargar el material desde la plataforma en cualquier momento del día, durante todo el tiempo que dure tu diplomado.

Además, nuestra plataforma es multidispositivo, así podrás estudiar en cualquier computadora de escritorio, *laptop*, tableta o *smartphone*.

Soporte técnico:

El equipo de soporte técnico estará a tu disposición en todo momento para ayudarte a resolver cualquier situación.

Chatbot:

Mediante el cual te podemos apoyar en todos los temas relacionados con tu experiencia en el diplomado y generamos *tickets* de servicio para tu comodidad, tranquilidad y seguridad.

Asesoría y acompañamiento:

Cuentas con Seguimiento Académico a través de *Whatsapp* y otras herramientas a distancia en tiempo real, para resolver tus dudas y dar retroalimentación.

Evaluación y seguimiento ágil:

Tendrás retroalimentación fluida y objetiva de tu progreso en el programa para el logro de tu certificado internacional.

UVM / EDUCACIÓN
CONTINUA