

GLOBAL MASTER IN AI & FULL STACK DEVELOPMENT

Horarios

Opción entre semana:

Lunes a Jueves*

Tardes
19.00 - 22.00 H.

Opción fin de semana:

Viernes
16.00 - 20.00 H.

Sábados
10.00 - 14.00 H.

*El módulo de **PYTHON** entre semana solo se imparte en el siguiente horarios:

Tardes
16.00 - 19.00 H.

Pide cita **aquí**
con un asesor de estudios

Consulta precio, ofertas y promociones actuales en

info@cei.es
www.cei.es

CEI.

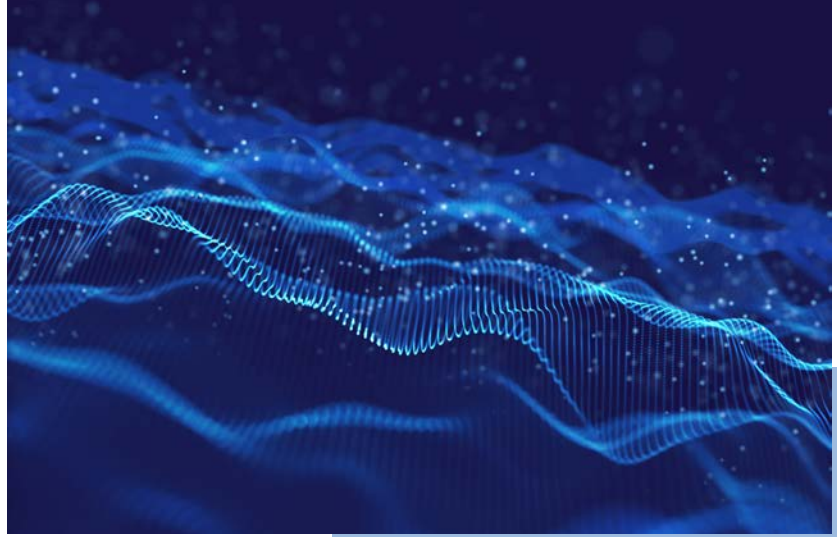


720 H. (12 meses)

560 H. Presencial+*Online* + 160 H. Proyecto final

* El contenido de este programa es orientativo, pudiendo variar el mismo por razones de actualización o modificación de los contenidos.

OBJETIVOS



El **Global Master in AI & Full Stack Development** está diseñado para formar profesionales capaces de desarrollar aplicaciones web completas integrando tecnologías de programación, diseño web y soluciones basadas en inteligencia artificial.

En un entorno digital en constante evolución, las empresas demandan perfiles capaces de diseñar, programar e implementar aplicaciones web de principio a fin, combinando conocimientos de desarrollo *frontend* y *backend* con herramientas avanzadas de programación y automatización. Este programa proporciona una formación práctica orientada al desarrollo de proyectos digitales reales.

A lo largo del **Global Master**, los estudiantes aprenderán a crear interfaces web modernas, programar aplicaciones dinámicas y desarrollar soluciones inteligentes utilizando Python, adquiriendo las competencias necesarias para participar en todo el ciclo de desarrollo de productos digitales.

Áreas de formación

Inteligencia Artificial: Fundamentos

Introducción a los principios de la inteligencia artificial, aprendizaje automático y análisis de datos. Los estudiantes comprenderán los conceptos clave que permiten desarrollar sistemas capaces de aprender, predecir y optimizar procesos.

Diseño Web con HTML5 y CSS

Fundamentos y técnicas de desarrollo *frontend* para la creación de interfaces web modernas, accesibles y adaptadas a diferentes dispositivos. Los estudiantes aprenderán a estructurar y diseñar páginas web utilizando estándares actuales de desarrollo web.

Programación en Python

Aprendizaje del lenguaje de programación Python aplicado al desarrollo de aplicaciones, automatización de procesos y creación de soluciones tecnológicas. Python es uno de los lenguajes más utilizados en desarrollo web, análisis de datos e inteligencia artificial.

Full Stack Development

Se aprenderá a crear aplicaciones web completas, desde la interfaz en hasta el backend, trabajando con bases de datos; a estructurar proyectos con JavaScript moderno, tipar las aplicaciones y a construir proyectos escalables y optimizados. Todo ello escribiendo el código de forma manual para dominarlo, utilizando la IA como un complemento para depurar, documentar y acelerar el flujo de trabajo.

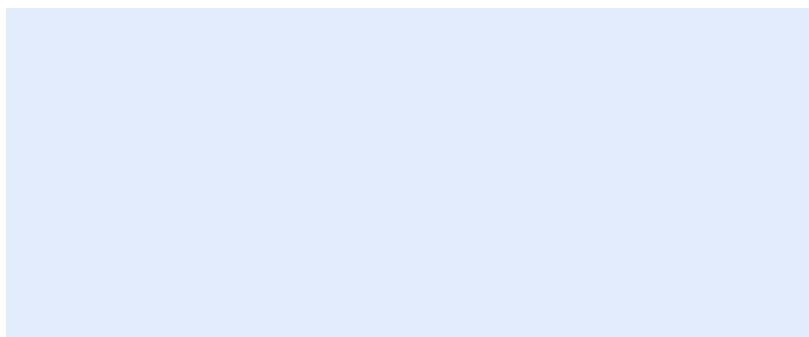
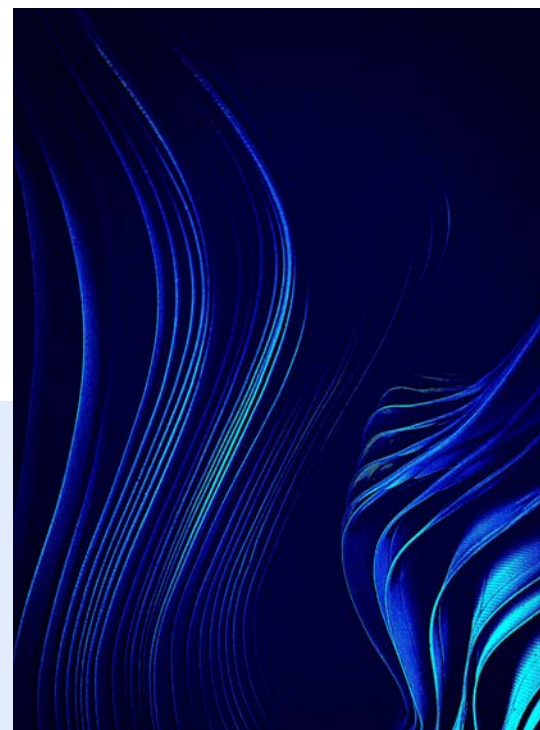
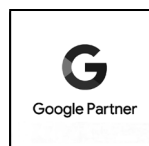
Objetivo del Global Master

El objetivo del programa es formar profesionales capaces de desarrollar aplicaciones web completas e integrar soluciones de inteligencia artificial en productos digitales, preparándolos para trabajar en entornos tecnológicos donde la programación, la innovación y la capacidad de adaptación son claves para la transformación digital.

OFICIAL

CEI –Centro de Estudios de Innovación, Diseño y Marketing– es un centro de formación reconocido por el Ministerio de Educación para impartir estudios oficiales de grado superior y homologado por las compañías que lideran el sector de las artes digitales y las nuevas tecnologías.

Ser el único centro de formación de España que reúne las siguientes certificaciones es una garantía indiscutible para nuestros alumnos: Adobe Certified Professional, Apple Authorized Training Center, Autodesk Authorized Training Center, Maxon Training Provider, Microsoft Certified Educator, Unity Official Training Center, Google Partner, IBM Education Partner, Cisco Networking Academy, AWS Academy... Además, nuestra Escuela ha sido galardonada por los Premios Excelencia Educativa como Mejor Programa Educativo para Inserción Profesional, Mejor Centro de Formación Presencial, Mejor Web de Centro de Formación Presencial y Mejor Centro de Formación Profesional.





LA ESCUELA

CEI –Centro de Estudios de Innovación, Diseño y Marketing– es una escuela de diseño digital fundada en el año 1991 especializada en el sector de las nuevas tecnologías, el diseño y el marketing digital, con sedes en Madrid, Sevilla y Valencia. Nuestros estudios se desarrollan en seis áreas de formación: **diseño gráfico y dirección de arte, diseño y programación web, marketing digital y business, producción, postproducción audiovisual (VFX) y motion graphics, diseño de interiores y arquitectura BIM, arte, diseño y programación de videojuegos, y big data y cloud.**

Ahora sabemos que acertamos cuando, en **1991**, apostamos por una enseñanza de calidad como **escuela de diseño, marketing y nuevas tecnologías**. Nuestra máxima fue, es y seguirá siendo que, el alumno, cuando acabe el periodo lectivo, maneje a la perfección las técnicas impartidas durante su formación. Por esto, la inmensa mayoría de los miles de alumnos que han pasado por nuestras aulas se han incorporado al mercado de trabajo o han mejorado sus condiciones laborales tras terminar sus estudios en CEI.

Con más de 30 años en el sector, más de 20.000 alumnos formados y un equipo de profesionales altamente cualificados, CEI pretende potenciar las habilidades de sus alumnos adaptándose constantemente a las fuertes exigencias del mercado laboral de todo el mundo. Es por todo ello que nuestra escuela ha sido galardonada, en los últimos años, por los **Premios Excelencia Educativa como Mejor Centro de Formación Presencial, Mejor Programa Educativo para Inserción Profesional, Mejor Web de Centro de Formación Presencial y Mejor Centro de Formación Profesional.**

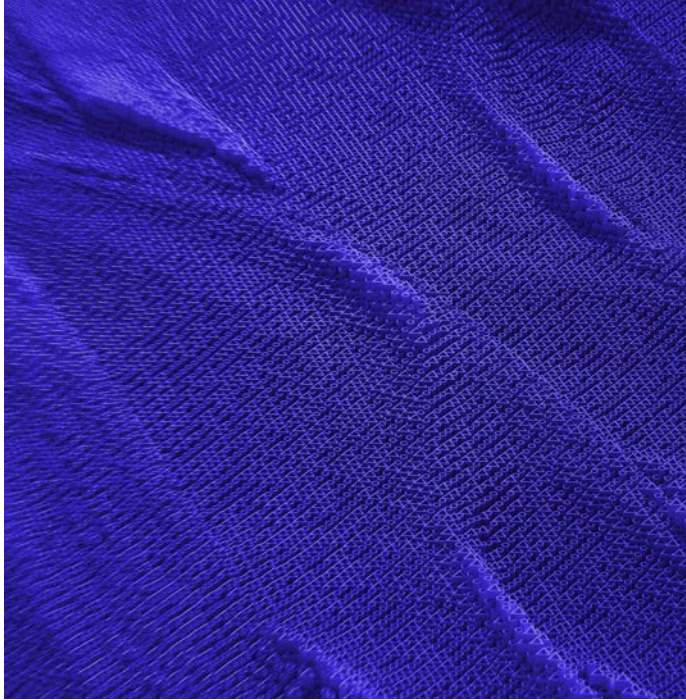
En 2022, CEI unió fuerzas con **Omnes Education Group**. Omnes Education es una institución privada de educación superior e investigación multidisciplinar que ofrece programas educativos en los campos de la gestión, la ingeniería, la comunicación y los medios digitales y las ciencias políticas. Sus campus se encuentran en Abiyán, Barcelona, Beaune, Burdeos, Chambéry, Ginebra, Londres, Lyon, Madrid, Mónaco, Múnich, París, Rennes, San Francisco, Sevilla y Valencia, repartidos entre 13 escuelas y universidades diferentes. Con más de 40.000 estudiantes cada año y más de 200.000 antiguos alumnos, Omnes ocupa un lugar único en el panorama educativo internacional.

Otro de los puntos fuertes de la escuela es el **equipo de profesores altamente cualificados**, que ponen a disposición de los estudiantes varios años de experiencia en sus respectivas disciplinas. Un centro de formación homologado ha de comunicar quiénes son sus profesores/as y someterles al control de la compañía propietaria del *software* sobre el que impartimos formación. Esto garantiza su permanente conocimiento de todas las funcionalidades de las aplicaciones, su contrastada valía y experiencia profesional, así como su indudable capacidad y calidad como docente.

PROGRAMA

- Fundamentos de la Inteligencia Artificial
- Python para análisis y modelado de datos
- Aprendizaje Automático (*Machine Learning*)
- *Deep Learning*
- Procesamiento del Lenguaje Natural (NLP)
- *Computer Vision*
- Figma
- HTML y SEO *onsite*
- CSS y SASS
- JavaScript
- Github y Github Copilot
- ChatGPT
- React + TypeScript
- NextJS
- Puesta en producción
- Fundamentos de Python
- Entorno de trabajo y reutilización del código
- Programación orientada a objetos
- Persistencia y manipulación de datos
- *Frameworks*





MÓDULO
**INTELIGENCIA
ARTIFICIAL (IA):
FUNDAMENTOS Y
APLICACIONES
PRÁCTICAS
180 H.**

01 FUNDAMENTOS DE
LA INTELIGENCIA
ARTIFICIAL

- Introducción a la IA: ¿qué es IA hoy?
Impacto de negocio. ML vs. GenAI
- Formulación correcta del problema
- Métricas de negocio
- Sesgos en los datos y modelos: *Data Leakage* a nivel conceptual
- Marco ético e impacto en la sociedad

02 PYTHON PARA
ANÁLISIS
Y MODELADO
DE DATOS

- Estructuras de Control aplicadas a datos
- Funciones y Módulos
- Manipulación de datos (Numpy y Pandas)
- EDA y *Feature Engineering*

03 APRENDIZAJE AUTOMÁTICO (MACHINE LEARNING)

- Tipos de aprendizaje: supervisado, no supervisado y por refuerzo
- Algoritmos de clasificación, regresión y *clustering*
- Evaluación de modelos y selección de características
- Uso de bibliotecas como Scikit-Learn

04 DEEP LEARNING

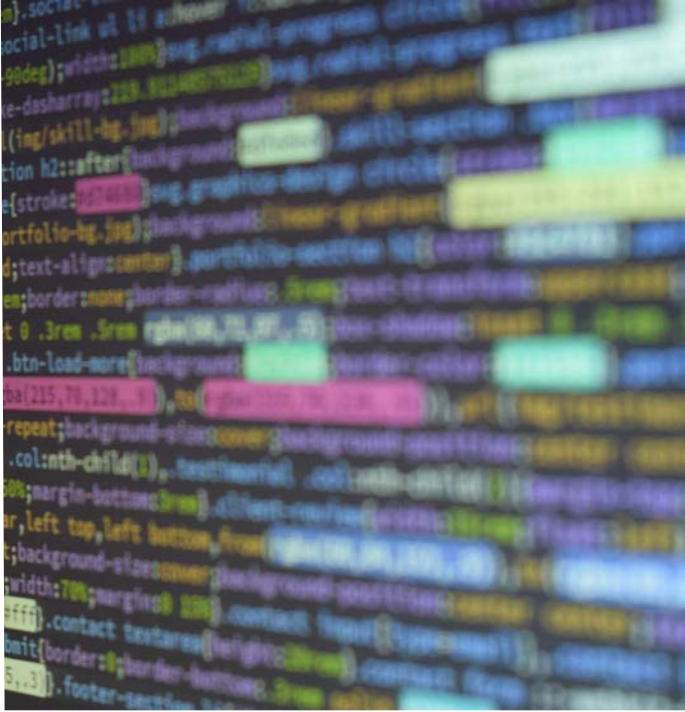
- Redes Neuronales: conceptos y fundamentos
- *Frameworks para Deep Learning*
- Regularización/*Overfitting*
- Arquitecturas de redes neuronales: MLP, redes convolucionales, recurrentes y *Transformers Transfer*
- *Learning y Fine Tuning*
- IA Generativa

05 PROCESAMIENTO DEL LENGUAJE NATURAL (NLP)

- Fundamentos del procesamiento del lenguaje natural
- *Text Mining*: TF-Idf, *Word Cloud* y Análisis del Sentimiento.
- Representaciones vectoriales: *Word Embeddings* y *Sentence Embeddings*
- LLM (*Large Language Models*) y *Transformers*
- *Prompt Engineering* y uso práctico de LLMs
- RAG (*Retrieval-Augmented Generation*)

06 COMPUTER VISION

- Introducción
- CNN y visión basada en *Transformers*
- *Transfer Learning*
- Detección y reconocimiento de objetos.
- Modelos Multimodales



MÓDULO

DISEÑO WEB CON HTML5, CSS3 Y JAVASCRIPT

180 H.

01 INTRODUCCIÓN AL DISEÑO WEB

- Perfiles en desarrollo web: *Frontend Developer, Backend Developer, Fullstack Developer, IA First Developer*
- Nuevos perfiles: *IA First Developer, Frontend IA Developer*
- Documentación oficial: cómo leerla y trabajar con ella
- Herramientas de apoyo: inspector de elementos y VSCode

Inteligencia Artificial

- Ecosistema: Empresas, Modelos...
- VSCode Copilot vs. Cursor
- IA Integrada en navegadores

02 ESTRUCTURA DE UNA WEB/APP

- Etiquetas y SEO
- Relación semántica entre etiquetas
- Elementos semánticos y elementos estructurales
- Títulos, párrafos, listas, tablas, enlaces e imágenes
- Formularios

Clean Code y Metodologías

- BEM (Bloque, Elemento y Modificador)
- SUITCSS (Metodología basada en componentes y utilidades)

Web IA Native

- Microdatos estructurados en JSON-LD
- Microdata en HTML mediante schema.org
- Técnicas IA Native

Accesibilidad con WCAG

- Accesibilidad en el HTML
- Adaptar la web con técnicas WCAG

Optimización/Performance

- Carga diferida en imágenes
- Carga diferencial para Javascript
- Optimización de imágenes
- Ofuscación (minificación de código)

Inteligencia Artificial

- Documentación de Metodologías
- Comparación de metodologías
- Mejoras en SEO para los textos con Microdata

03 APLICANDO ESTILO

- Flujo de CSS
- Propiedad Display: el comportamiento de las etiquetas
- Reglas, selectores, propiedades y declaraciones
- Tipos de selectores y especificidad

- Uso del EM y REM como medida responsable

Box Model

- Unidades de medida, absolutas y relativas
- *Margin, padding y border*
- *Removed/Natural flow* en el *position*
- Cómo centrar etiquetas de forma matemática
- Diseño responsive, @media-queries
- *Flex* y *Grid* en CSS

Tipografía

- Propiedades tipográficas
- Tipografía web: Google Fonts y tipografías externas
- Color: hexadecimal, RGB, RGBA
- *Variable fonts*: haciendo la tipografía responsive

Modern CSS

- *Nesting* o anidación de etiquetas
- @container: otra nueva forma de hacer responsive
- @layer: controlando la especificidad por capas en CSS
- CSS Snap para desplazamiento en móviles
- Técnicas para tipografía flexible y *responsive*

Inteligencia Artificial

- *Frameworks* de CSS: Bootstrap y Tailwind
- Depurando CSS con ChatGPT
- Depurando CSS con Copilot
- Depurando con IA en el Navegador

04 SASS: POTENCIA TU CSS

- Variables en SASS vs. *Custom Properties* en CSS
- *Mixins*: optimizando nuestro código
- Modularizando nuestro código
- *Nesting* de SASS vs. *Nesting* de CSS

Inteligencia Artificial

- Paradigma de los pre-procesadores con Copilot
- Depuración de SASS con Copilo

05 JAVASCRIPT: INTERACCION Y FUNCIONALIDAD

- Programación vs. Maquetación
- Variables, constantes y funciones
- API del DOM: editando el HTML desde JS
- Capturar elementos del DOM
- Gestión de clases, atributos y contenido de HTML

Eventos

- Eventos: *pointers, click, mouseover...*
- Parámetro e: detectando información sobre los eventos
- *Callbacks*
- *Intersection Observer*

Inteligencia Artificial

- Documentación con IA
- Depuración de Javascript con IA
- Optimización de código con Copilot

API REST

- JSON como forma de envío de información
- Qué son los API REST
- Promesas y asincronía en peticiones
- *Fetch* como sistema para enviar y recibir información entre API REST

06 GIT Y GITHUB

- Uso de GIT con terminal
- Usando Github Desktop
- *Commits*: guardando versiones de nuestra información
- Ramas: el sistema de gestión de *commits*
- Publicando nuestra web con Github Pages

Inteligencia Artificial

- Mejorando *commits* en Github Desktop
- Github Editor Web + Copilot
- Github Copilot en VSCode

07 INTELIGENCIA ARTIFICIAL (ESPECIALIZACIÓN)

- Ecosistema de la Inteligencia Artificial
- Cómo funciona una IA y la importancia del Contexto
- *Prompting* para Desarrolladores
- La importancia del contexto y MCP
- Deuda técnica: problemas habituales de la IA

Procesos de aprendizaje

- NotebookLM

IDE: Editores de código

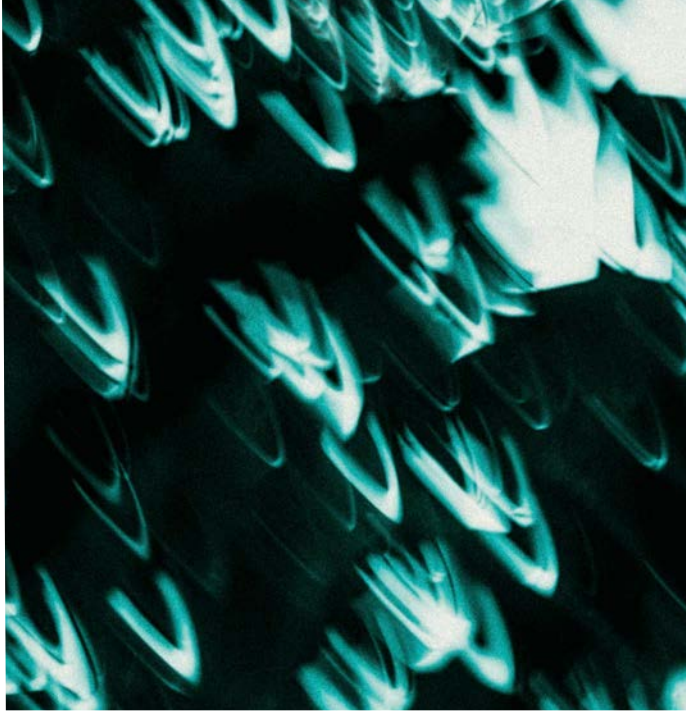
- Visual Studio Copilot
- Cursor y Windsurf
- IA Integrada en Google Chrome

Introducción a la terminal

- Terminales en macOS y Windows
- Gemini CLI

08 FIGMA PARA DEVS

- Integración de Figma en las prácticas de clase
- Interfaz y navegación
- Exportación de *assets* y código CSS
- Introducción a la integración con herramientas de desarrollo: Figma DevMode



MÓDULO
**DESARROLLO
WEB
*FULL STACK
DEVELOPMENT***
180 H.

01 INTRODUCCIÓN
AL DESARROLLO
FULL STACK

- Documentación oficial: cómo leerla y trabajar con ella
- Perfiles en el desarrollo de aplicaciones *Full stack*
- Tipos de herramientas en el ecosistema de JavaScript
- Por qué se estudian tres elementos, *Frameworks*, API-REST y bases de datos

Inteligencia Artificial

- Ecosistema de la IA
- Empresas y Modelos
- Herramientas de IA
- Agentes, Copilotos y Asistente

02 TERMINAL

- Qué son las terminales/*powershell*
- Estudiando la terminal desde cero
- Comandos de funcionamiento para carpetas
- SUDO vs. Administrador: ejecutando CLI dentro de la terminal

Inteligencia Artificial

- Github Copilot CLI
- Gemini CLI
- Terminales con IA: Warp

03 NPM: INSTALANDO PAQUETES

- Qué es y para qué sirve Node.js
- Funcionamiento de empaquetadores
- Usando NPM o YARN como empaquetadores de Node.js
- Comandos en la terminal de funcionamiento de NPM
- Empaquetadores: NPM (*Node Package Manager*)
- VITE como *build tool* para la estructura de aplicaciones

Inteligencia Artificial

- *Prompting* para la instalación de paquetes
- Documentación de la instalación

04 BASES DE DATOS

- La necesidad de una base de datos
- Bases de datos relacionales vs. documentos
- ORM vs. ODM en ExpressJS
- MongoDB: uso y características
- MongoDB: crear, leer, actualizar y borrar

05 API REST CON EXPRESSJS

- Estructura básica de una API REST de Express
- Tipos de métodos: get, post, put, patch y delete
- Peticiones asíncronas
- *Middlewares* para controlar la información
- *Routing* con Parámetros y Controllers
- Configuración del entorno: *Production vs. Developer*
- Mongoose: conectando Express con MongoDB
- Esquemas de datos con Prisma
- Generación de modelos con Prisma
- CRUD aplicado en Prisma y MongoDB

Inteligencia Artificial

- Documentación de APIs con IA
- Diagramas con Mermaid.js
- *Prompting* específico para código generado por IA

06 REACTJS

- Introducción al uso de librerías/*frameworks*
- La importancia de automatizar el HTML con datos
- Conceptos básicos de React: JSX
- Componentes y props
- *Hooks*: useState, useEffect...
- Eventos
- *React Router*
- Formularios

Inteligencia Artificial

- Documentación de Apps con IA
- *Prompting* específico para código generado por IA

07 PUESTA EN PRODUCCIÓN

- Plataforma como servicio (PaaS)
- Infraestructura como servicio (IaaS)
- Servidores como repositorios de ficheros estáticos vs. el modelo tradicional

Entornos

- Entornos de desarrollo, *testing* y producción
- Variables en archivos *.env*
- Variables de entorno vs. Plataforma

Implementación CI/CD

- CI: cada *push*/PR en GitHub ejecuta automáticamente tests, *linters* y *builds*
- CD: automatización del despliegue una vez superados los tests

GitHub Actions

- YAML *workflows* para automatizar tareas
- Configuración de *jobs* y *runners*
- Integración con Vercel/Netlify para despliegues automáticos

Inteligencia Artificial

- Creación de apps con Vercel VO
- Creación de apps con Lovable

08 GIT Y GITHUB

- La importancia de Git en empresas
- *Commits*: guardando nuestra información
- Ramas: creación y combinación de código
- Usando Github Desktop como control de Git
- La importancia de Github Copilot

Eventos

- Eventos: *pointers*, *click*, *mouseover*...
- Parámetro *e*: detectando información sobre los eventos

Inteligencia Artificial

- Mejorando *commits* en Github Desktop
- Github Editor Web + Copilot
- Github Copilot en VSCode

09 INTELIGENCIA ARTIFICIAL (ESPECIALIZACIÓN)

- Ecosistema de la Inteligencia Artificial
- Cómo funciona una IA y la importancia del Contexto
- *Prompting* para desarrolladores
- La importancia del contexto
- Github Copilot en Visual Studio Code y CLI
- Deuda técnica: problemas habituales de la IA
- IA en VSCode Copilot
- IA en Warp
- IA Integrada en navegadores
- Github Copilot a fondo

10 FIGMA PARA DEVS

- Integración de Figma en las prácticas de clase
- Interfaz y navegación
- Exportación de *assets* y código CSS
- Introducción a la integración con herramientas de desarrollo: Figma DevMode



MÓDULO PYTHON 180 H.

01 FUNDAMENTOS DE PYTHON

- Introducción y conceptos básicos de Python
- Tipos de datos, variables y operadores matemáticos
- Manipulación avanzada de cadenas de caracteres (*Strings*)
- Entrada y salida de datos
- Estructuras de control de flujo y operadores condicionales
- Estructuras de datos dinámicas (Listas, Diccionarios, Tuplas, Conjuntos)
- Manejo, creación y registro de excepciones.
- Ideación, diseño y planificación del desarrollo: priorización de requisitos (MVP), flujo de trabajo y diagramas básicos para la organización del proyecto final.
- Comprensión de listas*

02 ENTORNO DE TRABAJO Y REUTILIZACIÓN DEL CÓDIGO

- Configuración del entorno de desarrollo (IDE, Terminal y GIT)
- Gestión de dependencias y entornos virtuales (Pip / Venv)
- Funciones (Parámetros, retornos y modularización funcional)
- Módulos y paquetes (Organización profesional de archivos)
- Programación funcional avanzada*
- *Clean code* y buenas Prácticas*
- Arquitectura de *testing* y calidad de *software**

03 PROGRAMACIÓN ORIENTADA A OBJETOS

- Introducción a la programación orientada a objetos
- Clases, objetos, métodos y propiedades
- Herencia, polimorfismo y composición
- Métodos especiales y sobrecarga de operadores*

04 PERSISTENCIA Y MANIPULACIÓN DE DATOS

- Lectura y escritura de archivos (Texto Plano, CSV y JSON)
- Manipulación de archivos y directorios

- Introducción a Bases de Datos Relacionales (Modelado SQL y gestión con SQLite)
- Consumo de servicios externos y APIs de terceros (Librería *Requests*)
- Introducción a *pandas**
- Bases de *web scraping**

05 FRAMEWORKS

- Fundamentos de diseño web con HTML5 y CSS3
- Arquitectura cliente-servidor y utilidad de los *frameworks*
- Creación de APIs con FastAPI
 - Creación rápida de *Endpoints*
 - Validación de datos
 - Documentación automática,
 - Seguridad básica
- Software en la web con Django
 - El patrón MVT
 - El ORM para gestión de bases de datos
 - Creación de plantillas,
 - Uso del panel de administración
 - Usuarios y permisos