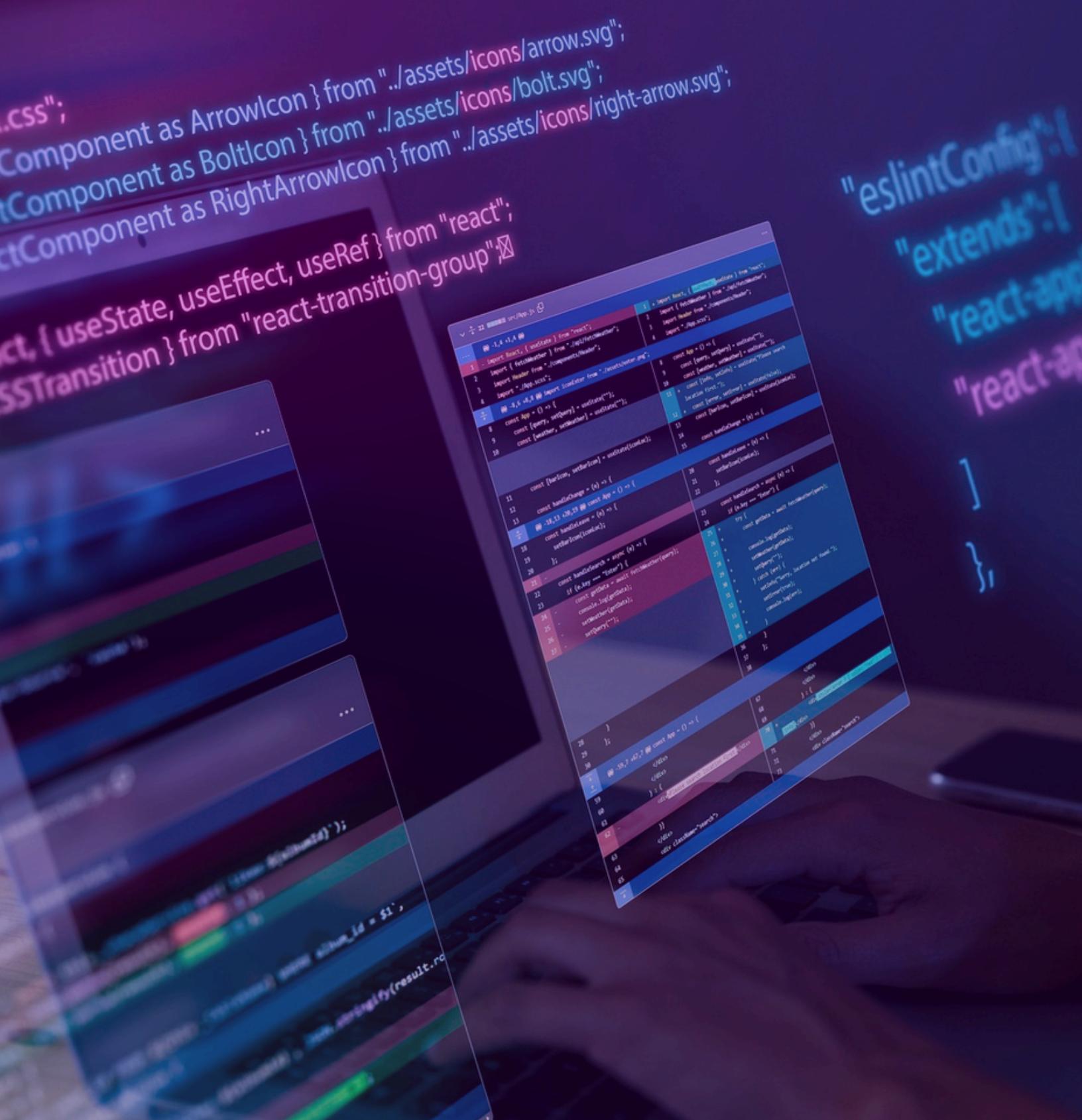


Python Senior AI

La Ruta Maestra del Desarrollo Real



Dev Senior Code

Python Senior AI

Dev Senior Code

El futuro lo creas tú...

Programa Completo

Python Senior AI: La Ruta Maestra del Desarrollo Real



Modalidad
Online - En Vivo



Nivel de dificultad
Inicial



Dedicación
Media - Alta



Proyecto Final.



Acerca del Programa.

"Python Senior AI: La Ruta Maestra del Desarrollo Real"

Este programa es mucho más que un curso. Es una experiencia formativa integral diseñada para llevarte de cero al nivel senior en desarrollo de software con Python, dominando tanto el backend como el frontend, integrando Inteligencia Artificial en todo el proceso, y creando proyectos reales para el mundo real.

A lo largo de 12 módulos, aprenderás desde los fundamentos del lenguaje hasta el despliegue completo de sistemas profesionales en producción, usando herramientas actuales como FastAPI, PostgreSQL, React, Tailwind, Docker, GitHub Actions, Vercel y Railway. Todo, con el poder de la IA a tu lado, ayudándote a escribir código limpio, validado, seguro y de alto impacto.

Cada modulo está estructurado:

- Clases técnicas semanales
- Clases tutorías de refuerzo
- Clases de inglés técnico con docentes nativos (Programa English Boost)

Además, trabajarás con:

- Inteligencia Artificial como entorno de desarrollo profesional y copiloto inteligente.
- Visual Studio Code como entorno de desarrollo profesional
- Proyectos reales desde el módulo 9 (empresas, universidades, clínicas)
- Pruebas automatizadas, despliegue en la nube, CI/CD y defensa técnica del proyecto final.

Este programa no solo te forma como programador, te prepara como profesional capaz de liderar, crear y transformar soluciones digitales reales con criterio técnico, visión global y dominio de herramientas modernas.

👉 Objetivo principal del programa.

El objetivo principal de este programa es capacitar a los estudiantes para convertirse en Desarrolladores Python Senior Full Stack con dominio de herramientas de Inteligencia Artificial, capaces de diseñar, desarrollar, probar, desplegar y mantener sistemas reales, seguros, conectados y escalables, con calidad profesional y listos para producción.

Esto incluye:

1. Dominio del Backend con FastAPI + IA

Aprenderás a construir APIs modernas, seguras y eficientes utilizando FastAPI como framework principal, integrando JWT para autenticación, validaciones profesionales con Pydantic y conexión a bases de datos reales con PostgreSQL. Usarás herramientas IA como copiloto para acelerar el desarrollo, detectar errores, optimizar la lógica de negocio y documentar tus endpoints automáticamente.

2. Gestión de Bases de Datos Relacionales con PostgreSQL

Dominarás el diseño, modelado e integración de bases de datos relacionales con PostgreSQL, aplicando relaciones entre entidades, claves primarias/foráneas, validaciones y consultas SQL seguras. Aprenderás a manejar persistencia de datos profesional y conectar tu API con consultas eficientes y escalables, asistido por IA.

3. Frontend Moderno con React + Consumo de APIs

Construirás interfaces de usuario dinámicas y profesionales utilizando React JS, Tailwind CSS y Vite. Implementarás rutas protegidas, consumo de APIs seguras, formularios validados, tokens JWT y navegación fluida. Con la ayuda de IA, generarás componentes reutilizables, refactorizarás vistas y mejorarás la experiencia del usuario de forma inteligente y eficaz.

4. Integración Full Stack, DevOps y Flujo Profesional de Desarrollo

Unificarás tus conocimientos backend y frontend en proyectos reales orientados a empresas, hospitales y universidades. Implementarás flujos de trabajo profesionales con CI/CD utilizando GitHub Actions, contenerización con Docker, despliegue en Railway (backend) y Vercel (frontend), y control de calidad mediante pruebas automatizadas.

Utilizarás la IA como asistente estratégico en cada etapa del ciclo de desarrollo, asegurando sistemas funcionales, mantenibles y listos para producción.

⚙️ Metodología del Programa

Este programa está estructurado en 4 módulos, cada uno abordando una etapa clave del desarrollo profesional Full Stack en Python, con integración total de Inteligencia Artificial como copiloto de productividad, desarrollo de proyectos reales, y acompañamiento académico intensivo.



Estructura semanal por módulo:

- 2 Clases (2 Horas - Cada una)
- 1 Tutoría de refuerzo técnico (2 horas)
- 1 Clase de inglés técnico con docentes nativos (2 horas) (Programa English Boost)

Cada módulo está diseñado para que el estudiante:

- Aplique los conceptos en un entorno profesional real.
- Utilice IA en todas las etapas del desarrollo.
- Trabaje con herramientas y flujos de trabajo modernos.
- Cree sistemas funcionales, conectados y listos para producción.



Metodología integral:

● Desarrollo Asistido por IA:

Cada clase incorpora el uso de herramientas inteligentes para escribir, revisar y optimizar código en tiempo real, acelerando el aprendizaje y garantizando calidad desde el inicio.

● Proyectos Reales por Dominio:

Cada módulo contiene un proyecto aplicado en un entorno profesional real (universidades, hospitales, empresas), que permite al estudiante aplicar conceptos en contextos de la industria.

● Tutorías Estratégicas:

Espacios de 2 horas dedicados exclusivamente al acompañamiento personalizado, resolución de dudas, revisión de código y fortalecimiento técnico.

● English Boost – Inglés con Propósito Profesional:

Cada semana los estudiantes asisten a una clase de inglés técnico con docentes nativos, enfocada en mejorar la comunicación para entornos tecnológicos, entrevistas y lectura de documentación.

● Preparación Profesional:

Desde el inicio se construye un portafolio sólido. Al finalizar, los estudiantes presentan su proyecto final ante docentes e invitados, y reciben entrenamiento en entrevistas técnicas y proyección laboral.



Beneficios del programa

Al completar este programa, los estudiantes estarán equipados con:

Habilidades Técnicas Senior + IA:

Dominio profundo en Python moderno, FastAPI, PostgreSQL, React JS, y herramientas profesionales como GitHub Actions, Docker, Railway y Vercel. Estarán preparados para construir sistemas reales, seguros, conectados y escalables, aplicando buenas prácticas y arquitecturas profesionales, con IA como copiloto de productividad y calidad.

Experiencia Profesional con Proyectos Reales:

Cada módulo está diseñado para aplicar el conocimiento en contextos reales. A partir del Módulo 9, los estudiantes desarrollan proyectos aplicados en dominios como universidades, hospitales, clínicas, tiendas e instituciones reales. Ganan experiencia completa desde el modelado de datos hasta el despliegue en producción, con CI/CD, pruebas automatizadas, documentación técnica, y portafolio en GitHub.

Preparación para el Mercado Global:

A través del programa English Boost, con docentes nativos, simulaciones de entrevistas técnicas, defensa de proyectos y mentoría directa, los estudiantes se preparan para enfrentar oportunidades como Desarrolladores Python Full Stack en el mercado laboral nacional e internacional. Además, aprenderán a comunicar sus proyectos de manera clara y profesional en entornos multiculturales.

Este programa no solo te prepara para enfrentar los desafíos técnicos más exigentes del desarrollo de software moderno, sino que también te entrena para integrarte directamente en equipos reales, aplicando herramientas actuales, flujos de trabajo productivos con IA y metodologías ágiles.

Nos emociona acompañarte en esta transformación profesional.

Prepárate para convertirte en un desarrollador completo, competitivo y listo para liderar proyectos de alto impacto.

¡Bienvenido a Python Senior AI: La Ruta Maestra del Desarrollo Real!

Módulos - Python Senior AI

Módulo 1: Python Junior Developer - Professional Coding Foundations

Incluye lo que originalmente era del Módulo 1 al Módulo 4 :

- Fundamentos modernos de programación con Python + IA
- Programación Orientada a Objetos (POO) y principios SOLID
- Estructuras de datos (listas, tuplas, diccionarios, sets)
- Manejo de archivos, validaciones y persistencia

Fundamentos Modernos de programación con Python e Inteligencia Artificial

En este módulo, el estudiante conocerá a fondo el ecosistema moderno de Python. Aprenderá a instalar, configurar y trabajar herramientas con IA como entorno de desarrollo principal y copiloto de productividad, además de dominar la sintaxis básica del lenguaje, estructuras fundamentales y su uso en contextos reales. Este módulo sienta las bases del pensamiento computacional profesional, con enfoque en legibilidad, lógica clara, y uso de la IA como asistente inteligente.

CLASE 1: Preparando el entorno profesional con herramientas IA y Visual Studio Code

Objetivos de aprendizaje:

- Instalar y configurar un entorno de desarrollo moderno y asistido con IA.
- Comprender el rol del programador profesional y el uso de asistentes inteligentes.
- Iniciar el uso de Python con buenas prácticas desde el primer día.

Contenidos

- Introducción al programa y visión del programador profesional.
- Instalación y configuración de:
 - Python (última versión estable)
 - Visual Studio Code (VS Code)
 - Git + GitHub para control de versiones básico
- Primera ejecución de scripts.
- Fundamentos de programación asistida:
 - ¿Qué puede hacer la IA por mí? ¿Y qué no debe hacer?
 - Ejercicios con autocompletado, sugerencias, refactorización inicial.
- Tipos de errores comunes y cómo interpretarlos con ayuda de la IA.

Ejercicio práctico:

- Crear el primer script con saludo personalizado, validación de datos y sugerencias de mejora asistidas por IA.

CLASE 2: Diagnóstico de instalación + primeros pasos en desarrollo con IA

Objetivo de la CLASE TUTORÍA:

- Asegurar que todos los estudiantes tengan el entorno profesional listo y operativo.
- Reforzar el pensamiento lógico inicial en Python, sin depender completamente de la IA.

Actividades:

- Verificación del entorno instalado: ejecución de scripts, acceso a terminal, integración de herramientas IA.
- Exploración asistida: probar qué tipo de código sugiere la IA y cuándo corregirla.
- Taller guiado: ejercicios de entrada, impresión, y uso de comentarios estructurados.
- Discusión técnica: cómo trabajar con IA de forma ética y productiva, evitando la dependencia.

CLASE 3: Variables, Tipos de Datos y Operadores en Python

Objetivos de aprendizaje:

- Comprender la lógica de variables, su declaración y asignación dinámica.
- Manipular tipos de datos primitivos en Python.
- Utilizar operadores correctamente y con sentido lógico.

Contenidos

- Declaración e inferencia de tipo.
- Tipos de datos: int, float, str, bool
- Conversión de tipos (casting)
- Operadores:
 - **Aritméticos:** +, -, *, /, //, %, **
 - **Relacionales:** ==, !=, >, <, >=, <=
 - **Lógicos:** and, or, not
- Herramientas IA como copiloto: sugerencias de mejoras en operaciones y formatos.
- Introducción a f-strings para impresión profesional de datos.

Ejercicio práctico:

- Mini script de calculadora básica que pida datos, procese operaciones y entregue resultados formateados profesionalmente.

CLASE 4: Buenas prácticas en el uso de variables, expresiones y validaciones

Objetivo de la CLASE TUTORÍA:

- Reforzar la comprensión profunda de tipos y operaciones.
- Promover el uso profesional de nombres, comentarios y estructuras.

Actividades:

- Ejercicios tipo reto: diseñar fórmulas, validar entrada de datos.
- Revisión grupal de código asistido por IA: ¿qué sugiere la IA? ¿es correcto?
- Introducción a la validación manual vs. automática.
- Taller: refactorización de código usando IA para hacerlo más limpio y expresivo.

CLASE 5: Estructuras de Control de Flujo (Condicionales y Bucles)

Objetivos de aprendizaje:

- Aplicar estructuras condicionales para control de decisiones en programas.
- Utilizar bucles para ejecutar procesos repetitivos de forma controlada.
- Aplicar validaciones y flujos de lógica que reflejan problemas reales.

Contenidos

- **Condicionales:** if, elif, else.
- **Bucles :** While, for
- Palabras clave break, continue y su uso estratégico.
- Construcción de flujos de lógica como menús interactivos o validaciones repetitivas.
- Comparación entre estructuras de control generadas por el estudiante y sugeridas por IA.

Ejercicio práctico:

- Programa de simulación de acceso: valida usuario y edad con mensajes personalizados, estructura modular y legibilidad profesional.
- Script de “control de intentos”: el usuario tiene 3 intentos para ingresar un dato correcto. Muestra mensajes personalizados y control de flujo automatizado .

CLASE 6: Resolución de problemas con lógica y estructuras de control

Objetivos de la CLASE TUTORÍA:

- Aplicar los conocimientos en escenarios prácticos donde la lógica se vuelve esencial.
- Reforzar el pensamiento algorítmico y la depuración con herramientas inteligentes.

Actividades:

- Retos de programación guiados: validación de contraseñas, sistema de intentos, menú de navegación.
- Diagnóstico de errores lógicos con ayuda de IA: por qué un bucle no se detiene, por qué un caso switch no funciona.
- Desarrollo colaborativo de una “calculadora de lógica”, que usa condicionales y bucles para resolver problemas simples.
- Feedback de buenas prácticas en la estructura y legibilidad del código.

CLASE 7: Funciones, Modularidad y Proyecto Práctico Integrador

Objetivos de aprendizaje:

- Aplicar el concepto de modularidad para dividir y organizar el código.
- Utilizar funciones con y sin retorno para resolver problemas.
- Consolidar los conocimientos del módulo en un proyecto estructurado.

Contenidos

- Sintaxis de funciones con def
- Parámetros, argumentos y valores por defecto
- Funciones puras vs funciones con efectos colaterales
- Uso profesional de return
- Introducción a *args y **kwargs para funciones flexibles
- Buenas prácticas: nombres claros, responsabilidades únicas
- Asistencia con IA:

- Refactorización de bloques repetidos
- Generación de funciones sugeridas
- Documentación automática de funciones

Ejercicio práctico:

- Aplicación de consola: Sistema de Registro de Estudiantes Funciones:
 - Registrar nuevo estudiante
 - Mostrar todos los registros
 - Buscar estudiante por nombre
 - Calcular promedio general
 - Validación de entradas y salidas con IA

CLASE 8: Desarrollo asistido y retroalimentación personalizada del proyecto

Objetivos de la CLASE TUTORÍA:

- Aplicar todo lo aprendido en un entorno de desarrollo simulado.
- Analizar, mejorar y defender el proyecto con acompañamiento profesional.

Actividades:

- Revisión en parejas del proyecto integrador con retroalimentación cruzada
- Simulación de presentación de código ante un líder técnico
- Diagnóstico automático del código: uso adecuado de funciones, modularidad, redundancia, claridad
- Refactorización conjunta del proyecto con sugerencias de IA
- Preparación de README y subida profesional a GitHub

Programación Orientada a Objetos

Este módulo introduce al estudiante en la Programación Orientada a Objetos (POO), uno de los pilares fundamentales del desarrollo en Python. Se aprenderá a modelar clases, utilizar herencia, aplicar interfaces. Se promueve el uso de IA para análisis y refactorización de código orientado a objetos.

CLASE 1: Clases, Objetos, Constructores.

Objetivos de aprendizaje:

- Comprender la estructura y el propósito de clases y objetos.
- Aplicar la encapsulación como base de la programación robusta.
- Construir y utilizar instancias de clases con propiedades y comportamientos definidos.

Contenidos

- ¿Qué es un objeto? ¿Qué es una clase? Analogías del mundo real.
- Atributos y métodos.
- Uso de constructores por defecto y parametrizados.
- Encapsulamiento
- Getters y setters con convenciones
- Creación de múltiples instancias y manipulación de objetos en memoria.
- Uso de IA para generar clases de manera automatizada y detectar errores de encapsulamiento.

Ejercicio práctico:

- Crear una clase Libro con atributos como título, autor, ISBN y precio.
- Crear una clase principal que permita crear objetos Libro y mostrar su información de forma segura.

CLASE 2: Refuerzo práctico de modelado de clases y encapsulamiento

Objetivos de la CLASE TUTORÍA:

- Reforzar la capacidad de crear clases con atributos y comportamientos claros.
- Evaluar el uso correcto del encapsulamiento y buenas prácticas.

Actividades:

- Taller de modelado orientado a objetos: crear clases Producto, Estudiante, Programa.
- Revisión de errores comunes: atributos públicos, falta de encapsulación, duplicación de código.
- IA como asistente de validación: ¿se pueden mejorar los getters/setters? ¿se están exponiendo datos innecesarios?
- Mini proyecto: Sistema de gestión de productos con operaciones básicas desde objetos.

CLASE 3: Herencia, Polimorfismo e Interfaces

Objetivos de aprendizaje:

- Aplicar herencia como mecanismo de reutilización de código.
- Comprender el polimorfismo como capacidad de adaptar comportamientos.
- Introducir el uso de interfaces como contrato de implementación.

Contenidos

- Herencia: uso de extends, reutilización y especialización de clases.
- Clases padre e hijas, superclase y subclase.
- Uso de super() para acceder a constructores y métodos superiores.
- Sobrescritura de métodos (@Override) y uso práctico del polimorfismo.
- Interfaces: declaración con interface, implementación con implements.
- Uso de Herramientas IA para sugerir métodos a sobrescribir y validar consistencia jerárquica.

Proyecto en clase:

- Crear una jerarquía de clases Empleado, Gerente, Desarrollador.
- Crear una interfaz Trabajable con método trabajar(), e implementarla en clases concretas.

CLASE 4: Reforzamiento de jerarquías, abstracción y reutilización

Objetivos de la CLASE TUTORÍA:

- Identificar jerarquías correctas e incorrectas.
- Aplicar polimorfismo en métodos que interactúan con clases abstractas o interfaces.

Actividades:

- Análisis de diagramas de clases reales: qué se hereda, qué se especializa.
- Ejercicio colaborativo: crear un modelo de zoológico con animales genéricos y específicos.
- Uso de IA para sugerencias de diseño orientado a objetos.
- Validación de objetos polimórficos: Empleado e = new Gerente();
- Casos de mal diseño (herencia innecesaria, duplicación de código).

CLASE 5: Modelado Avanzado de Clases en POO

Objetivos de aprendizaje:

- Diseñar soluciones utilizando técnicas avanzadas de modelado de clases.
- Modelar relaciones complejas entre clases, como composición y agregación.
- Implementar un sistema funcional utilizando buenas prácticas de diseño.

Contenidos

- Relaciones Avanzadas entre Clases
 - Composición y agregación en profundidad.
 - Relaciones bidireccionales y restricciones.
- Utilización de clases abstractas y métodos abstractos.
- Comparación entre clases abstractas e interfaces.

Ejercicio práctico:

- Se modelará un sistema para gestionar un restaurante, incorporando relaciones complejas entre clases.

CLASE 6: Modelado de Clases en POO

Objetivos de la CLASE TUTORÍA:

- Reconocer y aplicar conceptos avanzados de relaciones entre clases, como composición, agregación y herencia.
- Proponer mejoras con IA y aplicar refactorizaciones en el código.

Actividades:

- Realizar ejercicios de modelado y conversión de ellos a clases Java y relaciones manual.
- Herramientas IA como mentores automáticos: sugerencias para mejorar estructuras según OOP.
- Herramientas IA para Generación de modelo con el gráfico del modelo de datos

CLASE 7: Proyecto Integrador Orientado a Objetos con IA

Objetivos de aprendizaje:

- Integrar los conceptos de POO y principios SOLID en un proyecto funcional.
- Aplicar conocimientos adquiridos para diseñar una solución escalable y mantenible.
- Utilizar la IA para mejorar estructura, identificar errores y optimizar el diseño.

Contenidos

- Análisis de requerimientos y diseño de clases
- Definición de interfaces, herencias y comportamientos
- Separación de lógica en capas: main, servicio, modelo
- Validaciones básicas y manejo de errores
- Documentación de clases y métodos

Proyecto en clase:

- Sistema de gestión de clientes con roles, historial de acciones y reglas de negocio diferenciadas.

CLASE 8: Presentación, revisión estructural y retroalimentación

Objetivos de la CLASE TUTORÍA:

- Evaluar el diseño y la implementación del proyecto bajo criterios técnicos y de buenas prácticas.
- Aplicar revisión por pares y retroalimentación crítica.
- Mejorar el código con IA como copiloto de calidad.

Actividades:

- Presentación del diseño de clases (diagramas UML opcional)
- Validación de cumplimiento de principios OOP y SOLID
- Feedback técnico de parte del docente y compañeros
- Refactorizaciones finales con apoyo de IA
- Subida del proyecto a GitHub con README profesional y commits bien documentados

Estructuras de Datos y Manipulación Inteligente con IA

En este módulo, el estudiante dominará las principales estructuras de datos en Python: listas, tuplas, diccionarios y conjuntos. Aprenderá a manipular, transformar y recorrer estas estructuras de forma eficiente. Con el apoyo de herramientas IA podrá refactorizar estructuras complejas, depurar errores, optimizar recorridos y aplicar patrones reales en el tratamiento de datos empresariales.

CLASE 1: Listas y Tuplas - Organización secuencial de datos

Objetivos de aprendizaje:

- Comprender el uso de listas y tuplas para almacenar datos de forma ordenada.
- Aplicar operaciones de acceso, modificación, ordenamiento y segmentación.
- Usar IA para escribir código más claro y eficaz en recorridos y manipulaciones

Contenidos

- Introducción a listas: definición, acceso, modificación, recorrido
- Métodos comunes: append(), insert(), remove(), sort(), reverse(), pop()
- Introducción a tuplas: diferencias clave con listas
- Slicing y comprensión de listas (list comprehensions)
- Asistencia IA:
 - Optimización de bucles de listas
 - Refactorización de estructuras repetitivas
 - Generación de expresiones más limpias

Ejercicio práctico:

- Sistema de análisis de calificaciones de estudiantes: ingreso de notas, promedio, clasificación y retroalimentación estructurada con listas y tuplas.

CLASE 2: Taller de listas dinámicas y procesamiento profesional

Objetivos de la CLASE TUTORÍA:

- Consolidar operaciones con listas en flujos de datos más largos.
- Usar la IA para organizar estructuras y resolver errores comunes.

Actividades:

- Desafío de listas dinámicas: ingresar N productos y generar un reporte limpio.
- Uso de herramientas IA para identificar patrones, errores de índice y mejoras de rendimiento.
- Introducción a “listas de listas” para simular tablas de datos.

CLASE 3: Diccionarios y estructuras clave-valor

Objetivos de aprendizaje:

- Dominar los diccionarios como estructuras fundamentales para representar datos del mundo real.
- Realizar búsquedas, actualizaciones y recorridos eficientes.
- Refactorizar código desorganizado con IA para hacerlo más declarativo.

Contenidos

- Creación y acceso a diccionarios
- Métodos comunes: .get(), .keys(), .values(), .items(), .update(), .pop()
- Diccionarios anidados y estructuras mixtas
- Simulación de objetos con estructuras de datos
- Herramientas IA:
 - Limpieza de recorridos
 - Generación de diccionarios desde datos brutos
 - Optimización de impresión y validación

Ejercicio práctico:

- Simulador de registro de usuarios: ingreso de datos, actualización de perfiles, validaciones, y búsqueda eficiente en estructuras anidadas.

CLASE 4: Diccionarios complejos y diseño estructurado

Objetivos de la CLASE TUTORÍA:

- Trabajar con estructuras de datos más complejas.
- Validar integridad y consistencia de los datos ingresados.

Actividades:

- Proyecto mini-CRM: ingresar clientes con múltiples atributos.
- Análisis con IA: recomendaciones de estructura, separación de responsabilidades, y limpieza de impresión.
- Taller grupal: transformación de tablas a estructuras clave-valor.

CLASE 5: Conjuntos y estructuras mixtas

Objetivos de aprendizaje:

- Usar conjuntos (set) para manejar colecciones únicas y realizar operaciones matemáticas de conjuntos.
- Componer estructuras avanzadas a partir de listas, tuplas y diccionarios.

Contenidos

- Operaciones básicas: unión, intersección, diferencia
- Eliminación de duplicados

- Validaciones cruzadas con sets
- Composición de estructuras mixtas:
 - Listas de diccionarios
 - Diccionarios con valores en listas
- Herramientas IA:
 - Generación de expresiones más limpias
 - Reestructuración automática de estructuras para mejor acceso

Ejercicio práctico:

- Sistema de control de inventario: consolidación de productos, detección de duplicados, y validación de stock.

CLASE 6 : Composición inteligente de estructuras - práctica integral

Objetivos de la CLASE TUTORÍA:

- Aprender a elegir la estructura adecuada para cada tipo de dato.
- Usar IA para transformar, validar y limpiar estructuras mal diseñadas.

Actividades:

- Reto: construir el esquema de datos para una biblioteca (libros, autores, préstamos, fechas)
- Revisión cruzada: uso de Herramientas IA para sugerir alternativas de estructura
- Ejercicio de refactorización completa: eliminar redundancias, reestructurar datos.

CLASE 7: Buenas prácticas, limpieza de datos y manipulación aplicada

Objetivos de aprendizaje:

- Aplicar todas las estructuras aprendidas en un mini sistema real.
- Entender el uso de expresiones lambda y su poder cuando se combina con funciones como map, filter y reduce.

Contenidos

- Qué es programación funcional?
- Expresiones lambda
- Uso de map, filter y reduce con listas y estructuras complejas
- Validación de datos antes de ingresar
- Limpieza de datos no estructurados
- Separación de lógica de presentación y lógica de negocio
- Organización de código en bloques con IA
- Preparación para trabajar con estructuras más avanzadas como JSON

Ejercicio práctico:

- Sistema de control de inventario: consolidación de productos, detección de duplicados, y validación de stock.

CLASE 8: Evaluación final del módulo y optimización estructural con IA

Objetivos de la CLASE TUTORÍA:

- Demostrar dominio completo de estructuras básicas y complejas.
- Aplicar IA como herramienta profesional para revisión final del código.

Actividades:

- Análisis final de estructuras: ¿cómo se pueden mejorar?
- Taller de documentación y refactorización con IA
- Presentación final del sistema de gestión de eventos
- Cierre de módulo: reflexión sobre lo aprendido y conexión con el siguiente nivel (archivos, persistencia y más).

Manejo de Archivos, Persistencia y Validación con IA

En este módulo, los estudiantes aprenderán a leer, escribir y actualizar datos en archivos de texto, CSV y formatos estructurados. Se abordarán principios de persistencia, validaciones, estructuras organizadas y control de errores, todo asistido por IA para garantizar calidad, limpieza y seguridad en el flujo de datos.

CLASE 1: Archivos de texto plano - Escritura y lectura segura

Objetivos de aprendizaje:

- Comprender cómo funciona la persistencia de datos en archivos .txt.
- Aplicar métodos de apertura, lectura, escritura y cierre.
- Usar IA para organizar operaciones de entrada/salida eficientemente.

Contenidos

- Función open() y modos de apertura (r, w, a, r+)
- Escritura línea por línea con .write()
- Lectura con .read(), .readline(), .readlines()
- Uso del bloque with para manejo automático de cierre
- Asistencia IA:
 - Refactorización de operaciones repetidas
 - Creación automática de funciones para carga y guardado

Ejercicio práctico:

- Sistema de registro de tareas: guardar en archivo, agregar nuevas y mostrar listado en consola.

CLASE 2: Validación, organización y refactorización del flujo de archivos

Objetivos de la CLASE TUTORÍA:

- Reforzar la lógica de escritura/lectura segura.
- Organizar el código con IA para hacerlo más profesional.

Actividades:

- Refactorizar un sistema de archivo mal estructurado con IA.
- Validaciones básicas: archivo vacío, datos mal formateados.
- Automatizar la creación de backups usando lógica y sugerencias IA.

CLASE 3: Archivos CSV - Manejo estructurado y formateado de datos

Objetivos de aprendizaje:

- Trabajar con datos estructurados como listas y diccionarios en formato CSV.
- Usar el módulo csv para importar y exportar datos.
- Escribir código limpio y seguro para archivos con estructuras tabulares.

Contenidos

- Lectura de archivos CSV con csv.reader
- Escritura con csv.writer
- Uso de diccionarios con csv.DictReader y csv.DictWriter
- Separadores personalizados y codificación
- IA como asistente:
 - Generación de funciones para importar/exportar
 - Validación automatizada de filas

Ejercicio práctico:

- Crear un sistema de inventario que cargue, modifique y registre productos en archivo CSV.

CLASE 4: Depuración y verificación de integridad de datos CSV

Objetivos de la CLASE TUTORÍA:

- Asegurar que los datos CSV estén bien organizados y estructurados.
- Detectar problemas de codificación, duplicados o corrupción de líneas.

Actividades:

- Reto: detectar y corregir filas incompletas con IA.
- Comparación entre lectura manual y con DictReader.
- Optimización del flujo de validación usando código sugerido por Herramientas IA.

CLASE 5: Validaciones, errores y control de excepciones

Objetivos de aprendizaje:

- Aplicar estructuras profesionales de control de errores.
- Usar try/except/finally y validar datos antes de procesarlos.
- Usar IA como detector y corrector de errores comunes.

Contenidos

- Manejo de errores comunes: FileNotFoundError, ValueError, KeyError
- Validaciones de existencia, permisos y formato
- Uso de múltiples except y estructura else/finally
- Uso de IA para:
 - Sugerencia de validaciones faltantes
 - Refactor de bloques try/except mal escritos

Ejercicio práctico:

- Aplicación de carga de datos seguros: verifica antes de guardar, atrapa errores y genera mensajes limpios para el usuario.

CLASE 6: Taller de robustez - código preparado para fallos reales

Objetivos de la CLASE TUTORÍA:

- Fortalecer el control de errores en escenarios del mundo real.
- Usar IA para identificar riesgos y puntos débiles en el flujo.

Actividades:

- Casos simulados: archivo no existe, datos corruptos, permisos denegados.
- Taller de validación profunda de entrada/salida.
- Optimización asistida con IA: mejorar estructura de control sin sobrecargar el código.

CLASE 7: Proyecto Final – Sistema de Gestión con Persistencia en Archivos

Objetivos de aprendizaje:

- Integrar estructuras, archivos, validaciones y flujos de datos en un solo sistema profesional.
- Usar IA para optimizar, documentar y validar el sistema final.

Proyecto en clase

• Sistema de Gestión de Contactos:

- Añadir, buscar, editar y eliminar contactos
- Guardado persistente en CSV
- Validaciones y control de errores
- Refactorización completa con ayuda de IA

• Cierre de modulo

- Exportar backups
- Documentar funciones con docstrings asistidos por Herramientas IA
- Preparación para el próximo nivel: bases de datos reales

CLASE 8: Revisión profesional y checklist de calidad con IA

Objetivos de la CLASE TUTORÍA:

- Evaluar y mejorar el proyecto final del módulo.
- Garantizar que el código sea funcional, legible y mantenable.

Actividades:

- Auditoría de código con IA: estilo, eficiencia, seguridad.
- Reto de presentación: explicar la lógica, decisiones y estructura.
- Preparación de documentación técnica básica.

Módulo 2: Python Backend Developer – Enterprise API Engineering

- Bases de datos relacionales con PostgreSQL
- APIs profesionales con FastAPI + Pydantic
- Seguridad avanzada en APIs con JWT, roles y manejo de errores

Bases de Datos Relacionales con PostgreSQL e Integración Profesional con Python + IA

En este módulo, el estudiante aprenderá a trabajar con bases de datos reales usando PostgreSQL. Aprenderá a modelar datos, crear tablas, ejecutar consultas SQL, conectar su aplicación Python con la base de datos y aplicar principios de validación, integridad y consultas seguras. La IA será utilizada como asistente para optimizar sentencias, detectar errores comunes y sugerir mejoras de rendimiento.

CLASE 1: Fundamentos de bases de datos y diseño relacional

Objetivos de aprendizaje:

- Comprender los conceptos básicos de bases de datos relacionales y el modelo entidad-relación.
- Diseñar esquemas de tablas con llaves primarias, foráneas y tipos de datos correctos.
- Utilizar la IA para sugerencias de modelado y relaciones entre entidades.

Contenidos

- ¿Qué es una base de datos relacional?
- PostgreSQL: instalación local o conexión a base en la nube (Railway, Supabase)
- Tipos de datos: INT, VARCHAR, DATE, BOOLEAN
- Llaves primarias y foráneas
- Modelo entidad-relación y normalización básica
- Uso de IA:
 - Generación de modelos a partir de descripciones en lenguaje natural
 - Detección de campos redundantes

Ejercicio práctico:

- Diseño del modelo de una base de datos para un sistema de biblioteca: libros, autores, préstamos.

CLASE 2: Modelado lógico y físico de bases de datos

Objetivos de la CLASE TUTORÍA:

- Traducir requisitos funcionales a un modelo relacional correcto.
- Crear los scripts de base de datos y depurar errores comunes.

Actividades:

- Construcción guiada de tablas con CREATE TABLE.
- Asistencia con IA: sugerencias de relaciones, claves, índices.
- Revisión cruzada: ¿el modelo resuelve realmente el problema?

CLASE 3: CRUD SQL - Consultas reales para la gestión de datos

Objetivos de aprendizaje:

- Ejecutar operaciones básicas: insertar, consultar, actualizar y eliminar.
- Comprender y aplicar filtros, ordenamientos y consultas seguras.

Contenidos

- Operaciones INSERT, SELECT, UPDATE, DELETE
- Filtros con WHERE, operadores relacionales y lógicos
- Ordenamientos con ORDER BY, límites con LIMIT
- Buenas prácticas de escritura SQL con IA:
 - Detección de consultas ineficientes
 - Sugerencias de índices o claves

Ejercicio práctico:

- Crear una base de datos y ejecutar todas las operaciones CRUD para los registros de estudiantes y libros.

CLASE 4: Simulación de operación real - consultas SQL asistidas

Objetivos de la CLASE TUTORÍA:

- Fortalecer el dominio del lenguaje SQL en operaciones múltiples.
- Utilizar la IA como guía para optimizar consultas.

Actividades:

- Taller: escribir consultas en lenguaje natural y traducirlas a SQL con ayuda de herramientas IA.
- Desafío: encontrar errores en consultas mal redactadas y corregirlas con IA.
- Mini-reto de rendimiento: consultas más rápidas y organizadas.

CLASE 5: Conexión de Python con PostgreSQL - Proyecto real

Objetivos de aprendizaje:

- Conectar una aplicación Python a PostgreSQL.
- Ejecutar operaciones CRUD desde el código, validar datos y manejar errores.

Contenidos

- Introducción a la librería psycopg2 o asyncpg
- Conexión, cursor y ejecución de sentencias SQL
- Lectura de resultados y procesamiento de datos en Python
- Validaciones de entrada antes de ejecutar sentencias
- IA como asistente:
 - Generación automática de sentencias
 - Seguridad en las consultas (parámetros, inyección SQL)

Ejercicio práctico:

- Aplicación en consola: sistema de control de préstamos de libros conectado a PostgreSQL.

CLASE 6: Debug, validación y refactorización de consultas Python ↔ PostgreSQL

Objetivos de la CLASE TUTORÍA:

- Asegurar el correcto flujo de datos entre Python y la base de datos.
- Usar IA para evitar errores comunes de lógica y tipo.

Actividades:

- Pruebas de errores comunes: conexión fallida, claves duplicadas, campos nulos.
- Revisión asistida por IA: flujo completo CRUD desde Python.
- Introducción al uso de IA para generación de código de conexión más seguro.

CLASE 7: Proyecto de integración + mejores prácticas

Objetivos de aprendizaje:

- Aplicar todo lo aprendido en un sistema funcional con conexión real a base de datos.
- Validar, guardar y actualizar datos desde Python de forma profesional.

Proyecto en clase:

• Sistema real de gestión académica:

- Alumnos, Programa, notas
- Inserción, visualización y actualización de datos

- Validaciones, menú interactivo, conexión segura
- Refactorización con ayuda de IA

Asistencia IA:

- Refactorización de funciones
- Documentación automática
- Revisión de consultas mal formateadas

CLASE 8: Evaluación del proyecto - preparación profesional de backend conectado

Objetivos de la CLASE TUTORÍA:

- Evaluar funcionalidad completa y preparación para entornos reales.
- Ajustar errores, optimizar lógica y dejar el proyecto en condiciones profesionales.

Actividades:

- Evaluación del sistema completo: conexión, errores, control, legibilidad.
- Exportación de base de datos y documentación del proyecto.
- Última revisión con IA para sugerencias finales antes del módulo de backend con APIs.

Desarrollo de APIs Profesionales con FastAPI + Python + IA

En este módulo, el estudiante aprenderá a construir APIs modernas y profesionales utilizando FastAPI, un framework rápido, eficiente y ampliamente adoptado en la industria. Implementará endpoints seguros, CRUD completos, validaciones automáticas, documentación OpenAPI y pruebas básicas. Con Herramientas IA, podrá generar rutas, esquemas de datos, documentar funciones y depurar errores de forma ágil.

CLASE 1: Introducción a APIs RESTful con FastAPI y arquitectura básica

Objetivos de aprendizaje:

- Comprender qué es una API, sus principios REST y su rol en aplicaciones modernas.
- Crear su primera API real con FastAPI y probarla desde navegador y herramientas externas.

Contenidos

- ¿Qué es una API? ¿Qué significa REST?
- Introducción a FastAPI y ventajas frente a Flask y Django
- Instalación y configuración del entorno
- Estructura de una app FastAPI
- Definición de rutas GET, POST, PUT, DELETE
- Pruebas desde navegador y Postman
- Documentación automática (/docs)
- Asistencia con IA:
 - Generación de rutas
 - Documentación automática en Swagger
 - Refactorización de funciones

Ejercicio práctico:

- Crear una API para un sistema de tareas con rutas básicas funcionales y documentación Swagger disponible.

CLASE 2: Construcción de rutas completas con ayuda de IA

Objetivos de la CLASE TUTORÍA:

- Consolidar el dominio de los endpoints y estructuras de rutas.
- Usar IA para depurar rutas mal construidas, generar respuestas limpias y validar datos de entrada.

Actividades:

- Construcción de CRUD completo con IA asistiendo en generación y refactor.
- Simulación de errores comunes en endpoints: rutas incorrectas, respuestas no tipadas, errores 422.
- Revisión de documentación Swagger automática con retroalimentación del grupo.

CLASE 3: Validaciones, respuestas tipadas y estructuras con Pydantic

Objetivos de aprendizaje:

- Estructurar correctamente las peticiones y respuestas con Pydantic.
- Aplicar validaciones automáticas, respuestas personalizadas y estructura ordenada.

Contenidos

- Introducción a BaseModel de Pydantic
- Declaración de modelos para entrada (Request) y salida (Response)
- Validaciones automáticas: tipos, longitudes, rangos
- Respuestas personalizadas (status_code, JSONResponse)
- Herramientas IA como asistente:
 - Generación de esquemas de modelos
 - Sugerencias de validaciones y refactor

Ejercicio práctico:

- Mejorar la API de tareas integrando Pydantic para validar los datos de cada tarea y responder con estructuras limpias.

CLASE 4: Taller de validación completa y respuestas personalizadas

Objetivos de la CLASE TUTORÍA:

- Garantizar integridad de datos en cada operación de la API.
- Usar IA para enriquecer respuestas y mensajes para el cliente consumidor.

Actividades:

- Refactorizar manualmente rutas con múltiples validaciones
- Usar Herramientas IA para convertir estructuras sueltas en modelos con restricciones
- Simulación de errores: enviar datos incorrectos, revisar mensajes y status codes

Objetivos de aprendizaje:

- Conectar FastAPI a una base de datos PostgreSQL para persistencia real.
- Construir un sistema CRUD conectado y profesional.

Contenidos

- Integración de SQLAlchemy con FastAPI
- Declaración de modelos ORM (tablas)
- Sesiones de conexión y consultas
- Endpoints que interactúan con la base de datos
- Herramientas IA como asistente de código SQL embebido

Proyecto en clase:

- Ampliar la API de tareas para que guarde y consulte tareas desde una base de datos real conectada.

CLASE 6: Pruebas completas de API con conexión real

Objetivos de la CLASE TUTORÍA:

- Validar el flujo completo entre frontend, backend y base de datos.
- Verificar errores de conexión, duplicidad, claves inexistentes.

Actividades:

- Taller de pruebas con Postman: enviar datos válidos, inválidos y verificar respuestas.
- Auditoría con IA de las rutas, controladores y lógica SQL básica.
- Introducción a testing automático con pytest para endpoints.

CLASE 7: Proyecto Final – API profesional para un sistema real

Objetivos de aprendizaje:

- Integrar todo lo aprendido en una API profesional, segura, validada y documentada.
- Usar la IA para limpiar, ordenar y documentar automáticamente.

Proyecto en clase:

- Sistema de gestión de usuarios y tareas colaborativas
 - CRUD de usuarios y tareas
 - Validaciones, estados, fechas
 - Conexión a PostgreSQL
 - Documentación Swagger y /redoc
 - Refactor y documentación final con IA

Entregable final del módulo:

- Repositorio con endpoints funcionales
- Archivo .env con conexión
- Documentación generada automática + explicada

CLASE TUTORÍA 8: Evaluación completa de API y checklist de calidad profesional

Objetivos de la CLASE TUTORÍA:

- Validar que el estudiante pueda construir una API profesional desde cero.
- Mejorar su forma de escribir código backend robusto, limpio y validado.

Actividades:

- Revisión técnica de la API completa
- Auditoría asistida por IA: rutas sin documentar, duplicaciones, estructuras mejorables
- Exportación de la API con instrucciones claras
- Preparación para despliegue en la nube (proximidad a DevOps en módulo 7)

Seguridad en APIs con JWT, Autenticación y Manejo de Errores Profesionales

En este módulo, el estudiante aprenderá a proteger sus APIs utilizando tokens JWT, definir roles, Manejar sesiones seguras y construir controladores de errores globales. Además, aplicará prácticas modernas de validación, autenticación y autorización de usuarios. La IA será utilizada como guía técnica para refactorizar, detectar puntos inseguros y generar estructuras limpias y robustas.

CLASE 1: Fundamentos de autenticación y generación de JWT

Objetivos de aprendizaje:

- Comprender el flujo de autenticación en sistemas reales.
- Generar tokens JWT y aplicarlos en rutas protegidas.

Contenidos

- ¿Qué es un token JWT y cómo funciona?
- Instalación y uso de python-jose y passlib
- Generación y verificación de tokens
- Expiración y firma segura
- Simulación de login con usuario y contraseña
- Asistencia IA:
 - Sugerencias de estructura para login
 - Validación de lógica de expiración
 - Refactor de controladores

Ejercicio práctico:

- Implementar autenticación básica para la API de usuarios con generación de token JWT y expiración automática.

CLASE 2: Validación y debugging de autenticación con IA

Objetivos de la CLASE TUTORÍA:

- Garantizar el correcto flujo de autenticación y el uso seguro del token.
- Usar la IA para depurar errores comunes de seguridad.

Actividades:

- Probar login con credenciales inválidas y tokens vencidos
- Depuración con Herramientas IA: revisión de lógica condicional y flujos de error
- Simulación de ataques comunes y refuerzo de la lógica de protección

CLASE 3: Middleware, rutas protegidas y autorización basada en roles

Objetivos de aprendizaje:

- Proteger rutas con dependencias y validaciones JWT.
- Diferenciar permisos según el rol del usuario.

- Rutas protegidas con Depends en FastAPI
- Lectura y decodificación de token desde encabezados
- Middleware personalizado para validar el acceso
- Roles (admin, user) y autorización condicional
- Asistencia IA:
 - Detección de rutas expuestas sin protección
 - Refactorización de middleware

Ejercicio práctico

- Proteger las rutas de la API de tareas según el tipo de usuario con JWT y roles definidos.

CLASE 4: Diseño de flujo seguro y pruebas de rutas protegidas

Objetivos de la CLASE TUTORÍA:

- Comprobar que solo usuarios autorizados acceden a sus respectivos endpoints.
- Automatizar validaciones y respuestas ante accesos indebidos.

Actividades:

- Simulación de escenarios reales: intento de acceso sin token, con token inválido, con rol incorrecto
- Auditoría con IA: rutas mal protegidas o sin control de rol
- Taller de documentación del flujo seguro

CLASE 5: Manejo profesional de errores globales y respuestas claras

Objetivos de aprendizaje:

- Construir un sistema robusto de manejo de errores globales.
- Definir mensajes personalizados para mejorar la experiencia del cliente que consume la API.

Contenidos

- HTTPException, status_code, detail
- Respuestas personalizadas por error: 401, 403, 404, 422, 500
- Middleware de captura de errores globales
- Logger básico para registrar eventos importantes
- Asistencia IA:
 - Refactor de bloques try/except
 - Generación de mensajes más claros y descriptivos

Ejercicio práctico:

- Ampliar la API de usuarios y tareas con control de errores completo y mensajes claros, documentados automáticamente.

CLASE 6: Simulación de errores reales y respuesta estructurada

Objetivos de la CLASE TUTORÍA:

- Verificar que la API está preparada para producción en términos de seguridad y calidad.
- Utilizar IA como aliada para evaluar código, roles, y lógica de negocio.

Actividades:

- Validación completa: flujo de login, expiración de token, acceso restringido
- Auditoría con IA: flujos repetidos, controladores inseguros
- Revisión entre compañeros y documentación final

CLASE 7: Proyecto Final – API Segura, Validada y Robusta

Objetivos de aprendizaje:

- Integrar seguridad y control completo en una API funcional, moderna y profesional.
- Documentar el sistema con claridad para ser consumido por terceros.

Proyecto en clase:

- **Sistema real de gestión de cuentas y tareas seguras:**

- Login con JWT
- Registro y roles
- Rutas protegidas
- Validaciones estrictas
- Errores estructurados
- Logger de actividad
- Documentación completa y uso de Swagger

CLASE 8: Auditoría técnica del proyecto + checklist de seguridad profesional

Objetivos de la CLASE TUTORÍA:

- Verificar que la API está preparada para producción en términos de seguridad y calidad.
- Utilizar IA como aliada para evaluar código, roles, y lógica de negocio.

Actividades:

- Validación completa: flujo de login, expiración de token, acceso restringido
- Auditoría con Herramientas IA: flujos repetidos, controladores inseguros
- Revisión entre compañeros y documentación final

Módulo 3: Python Full Stack Developer - Advanced Systems Integration

- Frontend profesional con React + Tailwind
- Proyectos Full Stack reales (backend + frontend + DB)
- Testing automatizado (pytest, Vitest, Cypress)
- DevOps y despliegue con Docker, GitHub Actions, Railway y Vercel

Frontend Profesional con React JS, Consumo de APIs y UI Moderna con IA

En este módulo, los estudiantes construirán interfaces dinámicas y profesionales utilizando React JS, con diseño moderno, formularios validados y comunicación directa con APIs seguras construidas previamente. Se integrará el consumo de datos mediante fetch y axios, diseño modular con Tailwind CSS, enrutamiento con React Router y generación de componentes reutilizables. Herramientas IA será usada para autocompletar componentes, validar props, refactorizar lógica de estado y generar estructuras limpias.

CLASE 1: Fundamentos de React JS y entorno moderno con Vite + Tailwind

Objetivos de aprendizaje:

- Configurar el entorno moderno para desarrollar interfaces con React.
- Comprender los conceptos fundamentales de React y crear los primeros componentes funcionales.

Contenidos

- ¿Qué es React y por qué se usa en el frontend profesional?
- Vite: configuración rápida y óptima para proyectos modernos
- Estructura de carpetas y componentes
- Uso de useState, props y manejo de eventos
- Introducción a Tailwind CSS para estilos modernos
- Asistencia con IA:
 - Generación de componentes base
 - Refactorización de estructuras de JSX

Ejercicio práctico:

- Página inicial de sistema de tareas: vista principal, lista estática y formularios base.

CLASE 2: Modularización de componentes y refactor con IA

Objetivos de la CLASE TUTORÍA:

- Separar lógica y UI en componentes reutilizables.
- Mejorar el diseño y la estructura del frontend con ayuda de la IA.

Actividades:

- Separar en componentes: Header, TaskList, TaskItem, Footer
- Asistencia IA: identificar elementos repetidos y convertirlos en componentes limpios
- Aplicar diseño responsive básico con Tailwind CSS

CLASE 3: Manejo de formularios, estado y validaciones en React

Objetivos de aprendizaje:

- Configurar el entorno moderno para desarrollar interfaces con React.
- Comprender los conceptos fundamentales de React y crear los primeros componentes funcionales.

Contenidos

- useState y manejo de formularios controlados
- Validación manual de campos
- Envío de datos simulados al backend
- Feedback visual con clases dinámicas (Tailwind)
- Herramientas IA:
 - Generación de validaciones básicas
 - Refactorización de inputs y manejo de estado

Ejercicio práctico:

- Agregar tarea a una lista dinámica con validación de campos y alerta visual.

CLASE 4: Validación de formularios y conexión de lógica modular

Objetivos de la CLASE TUTORÍA:

- Organizar formularios grandes y manejar sus datos correctamente.
- Implementar validaciones limpias con ayuda de la IA.

Actividades:

- Crear un formulario de registro de usuario con validaciones
- Generar feedback visual con clases dinámicas
- Simulación de errores y revisión del comportamiento asistida por IA

CLASE 5: Consumo de APIs reales con Axios + React Router DOM

Objetivos de aprendizaje:

- Conectar el frontend a las APIs desarrolladas en módulos anteriores.
- Separar vistas con rutas protegidas, navegación y carga de datos.

Contenidos

- Uso de axios para GET, POST, PUT, DELETE
- Manejo de tokens JWT en encabezados
- Enrutamiento con React Router DOM: BrowserRouter, Routes, Link, useNavigate
- Renderizado condicional según estado
- Herramientas IA:
 - Generación de hooks personalizados para consumo
 - Sugerencias para control de errores y loading

Ejercicio práctico

- Panel de tareas dinámicas conectadas a API segura, con login y navegación de rutas.

CLASE 6: Integración completa React + API + Token

Objetivos de la CLASE TUTORÍA:

- Validar la comunicación completa entre frontend y backend.
- Proteger rutas y mantener estado de sesión en el navegador.

Actividades:

- Guardar token en localStorage
- Renderizar componentes según login
- Usar IA para refactorizar lógica de autenticación y sugerir mejoras

CLASE 7: Proyecto Final – Interfaz moderna de sistema real

Objetivos de aprendizaje:

- Construir una interfaz profesional conectada a una API con autenticación, consumo de datos y CRUD funcional.

Proyecto en clase:

• Sistema de Gestión de Tareas Colaborativo

- Login y registro de usuarios
- Lista de tareas por usuario
- Agregar, editar, eliminar tareas
- Rutas protegidas con React Router
- Interfaz moderna con Tailwind
- Token guardado, feedback visual
- Refactorización asistida por IA

CLASE TUTORÍA 8: Evaluación de interfaz profesional + presentación final

Objetivos de la CLASE TUTORÍA:

- Evaluar que el frontend cumple con requisitos visuales, funcionales y técnicos.
- Mejorar aspectos UI/UX con asistencia IA.

Actividades:

- Auditoría visual del sistema
- Revisión con IA: código repetido, componentes innecesarios, simplificación
- Documentación del flujo del usuario
- Preparación para proyecto Full Stack en el módulo 9

Proyecto Full Stack Real - Universidad Digital (Python + FastAPI + React + IA)

En este módulo, los estudiantes desarrollarán un sistema real de gestión universitaria, que incluye módulos para estudiantes, materias, notas, autenticación y reportes. El proyecto se construirá por capas, conectando backend seguro, base de datos relacional y frontend profesional.

El código será documentado, probado y versionado de forma profesional, con asistencia activa de IA para acelerar el desarrollo y garantizar calidad.

CLASE 1: Planificación, arquitectura y diseño del proyecto Full Stack

Objetivos de aprendizaje:

- Comprender la arquitectura de una aplicación real por capas.
- Planificar la base de datos, las rutas backend y las vistas frontend.
- Establecer flujo de trabajo y control de versiones.

Contenidos:

- Descripción del sistema: gestión académica universitaria
- Módulos: usuarios (roles), materias, inscripciones, calificaciones
- Diseño del modelo entidad-relación (IA como generadora de estructura)
- Configuración de monorepo: backend, frontend, base de datos
- Flujo técnico: API segura ↔ PostgreSQL ↔ React
- Herramientas: Git, GitHub, .env, documentación Swagger

Ejercicio práctico:

- Diagramar arquitectura del sistema, definir módulos y entidades.
- Crear repositorio e iniciar el scaffolding con IA asistiendo en generación de estructura base.

CLASE 2: Validación del diseño estructural y generación asistida del proyecto

Objetivos de la CLASE TUTORÍA:

- Asegurar que el diseño sea completo, escalable y bien estructurado.
- Apoyarse en IA para generar capas base, rutas y modelos.

Actividades:

- Revisión cruzada de modelos de datos
- Generación de clases y endpoints con IA
- Organización del backend en servicios, rutas y esquemas Pydantic

CLASE 3: Backend completo - Autenticación, entidades y relaciones en PostgreSQL

Objetivos de aprendizaje:

- Implementar el backend real del sistema, incluyendo autenticación, CRUD completo y relaciones.
- Conectar con PostgreSQL y proteger las rutas.

Contenidos:

- Rutas protegidas con JWT
- CRUD de estudiantes, materias e inscripciones
- Relaciones entre tablas: usuarios ↔ materias ↔ notas
- Serialización con Pydantic
- Validaciones y errores personalizados
- IA como copiloto:
 - Refactor de rutas
 - Documentación automática
 - Validación de estructura y duplicados

Ejercicio práctico:

- Backend funcional 100% conectado a PostgreSQL, seguro, validado y documentado.

CLASE 4: Integración completa y pruebas del backend profesional

Objetivos de la CLASE TUTORÍA:

- Validar el flujo completo del backend real.
- Probar todos los endpoints y documentar correctamente.

Actividades:

- Pruebas con Postman
- Validación de datos, errores y respuestas
- Auditoría con IA: refactor y detección de inconsistencias

CLASE 5: Frontend conectado a API real - vistas, rutas protegidas y formularios

Objetivos de aprendizaje:

- Crear el panel visual completo conectado al backend real.
- Construir formularios de inscripción, visualización de materias, notas y reportes.

Contenidos:

- Rutas públicas y protegidas con React Router
- Panel de estudiante: materias inscritas, notas
- Panel de administrador: CRUD de materias y notas
- Formularios con validación
- IA como asistente:
 - Componentes reutilizables
 - Refactorización y separación de lógica
 - Integración de hooks personalizados

Ejercicio práctico:

- Panel completo de usuarios con login, visualización de materias, y gestión académica conectada al backend.

CLASE TUTORÍA 6: Revisión de integración Full Stack y optimización profesional

Objetivos de la CLASE TUTORÍA:

- Validar conexión total entre frontend, backend y base de datos.
- Refactorizar, revisar y mejorar la experiencia general.

Actividades:

- Simulación de flujo de usuario: inicio de sesión → ver notas → agregar materia
- Revisión del código con IA: eficiencia, legibilidad, errores comunes
- Organización del repositorio y documentación técnica

CLASE 7: Presentación del sistema real + checklist profesional

Objetivos de aprendizaje:

- Consolidar el conocimiento desarrollando y presentando un sistema real.
- Defender las decisiones técnicas tomadas en la arquitectura.

Proyecto final del módulo:

• Sistema de Gestión Académica Universitaria

- Autenticación segura
- Panel de estudiante y administrador
- CRUD completo
- Persistencia y validación real
- Interfaz moderna y conectada
- Documentación técnica

Entregables:

- Repositorio con documentación
- Video corto de presentación (pitch técnico)
- Checklist de revisión final asistido por IA

CLASE 8: Defensa técnica del proyecto + retroalimentación entre pares

Objetivos de la CLASE TUTORÍA:

- Defender el sistema ante docentes e IA.
- Recibir retroalimentación profesional y recomendaciones.

Actividades:

- Presentación del proyecto ante jurado técnico
- Feedback técnico y refactor asistido con IA
- Revisión final de código y preparación para siguiente reto (testing + DevOps)

Testing, Cobertura y Calidad Profesional del Código con IA

En este módulo, los estudiantes implementarán pruebas automatizadas para backend y frontend, identificarán errores antes de que ocurran, y mejorarán la calidad general del software que construyen. Usarán frameworks como pytest, unittest, Vitest y Cypress, integrarán pruebas en sus proyectos reales, y utilizarán Herramientas IA como asistente para sugerencias, refactorización y cobertura de código.

CLASE 1: Introducción al testing automatizado en Python con pytest

Objetivos de aprendizaje:

- Comprender la importancia del testing en sistemas reales.
- Escribir pruebas unitarias y estructurarlas correctamente con pytest.

Contenidos

- ¿Qué es el testing? Tipos: unitario, integración, extremo a extremo
- Instalación y estructura básica de pytest
- assert, setup/teardown, manejo de excepciones
- Organización de tests por archivo y función
- Asistencia IA:
 - Generación automática de pruebas
 - Identificación de funciones no testeadas

Ejercicio práctico:

- Añadir pruebas a funciones del backend: autenticación, CRUD de tareas, validaciones.

CLASE 2: Cobertura de pruebas y refactorización con IA

Objetivos de la CLASE TUTORÍA:

- Validar la efectividad del testing implementado.
- Usar IA para sugerir nuevas pruebas o refactorizar tests repetitivos.

Actividades:

- Revisión de cobertura con pytest-cov
- Taller de refactor de pruebas duplicadas
- Generación de pruebas ausentes con IA: decisiones, edge cases

CLASE 3: Testing en el frontend con Vitest y React Testing Library

Objetivos de aprendizaje:

- Probar componentes de React y validar su comportamiento ante datos simulados.
- Generar pruebas unitarias y funcionales del frontend moderno.

Contenidos

- Instalación y uso de Vitest + Testing Library
- Simulación de eventos: clics, inputs, submits
- Renderización condicional y validación de contenido
- Mocking de props y funciones
- Herramientas IA como copiloto:
 - Generación automática de test cases para componentes
 - Refactor de pruebas largas

Ejercicio práctico:

- Probar componente TaskForm y TaskList, incluyendo comportamiento en diferentes condiciones.

CLASE 4: Pruebas de integración visual + flujo completo

Objetivos de la CLASE TUTORÍA:

- Validar el comportamiento general de vistas conectadas a datos reales.
- Refactorizar pruebas y encontrar errores ocultos en UI.

Actividades:

- Pruebas con datos de ejemplo
- Evaluación de cobertura y detección de fallos visuales
- Refactor guiado por IA

CLASE 5: Pruebas end-to-end con Cypress en el sistema real

Objetivos de aprendizaje:

- Automatizar escenarios completos de usuario.
- Detectar fallos en el flujo entre frontend y backend.

Contenidos

- Instalación y configuración de Cypress
- Escenarios: login, navegación, creación y edición de datos
- Captura de errores, tiempos de respuesta, validaciones
- IA como soporte:
 - Refactorización de escenarios
 - Mejora de descripciones, estructuras y tiempos

Proyecto en clase:

- Crear prueba E2E completa del flujo: login → crear tarea → ver confirmación.

CLASE 6: Auditoría del flujo de usuario y mejora UX con IA

Objetivos de la CLASE TUTORÍA:

- Evaluar el sistema desde la perspectiva del usuario final.
- Usar IA para mejorar respuesta visual, feedback y errores controlados.

Actividades:

- Simulación completa de uso
- Detección de errores visuales y mensajes poco claros
- Refactorización de feedback con IA: mensajes, alerts, validaciones

CLASE 7: Proyecto de Testing Completo + Checklist de Calidad de Código

Objetivos de aprendizaje:

- Implementar pruebas completas al sistema desarrollado en el módulo anterior.
- Validar cobertura, claridad y mantenimiento a largo plazo.

Proyecto en clase:

• Testing completo del Sistema de Gestión Universitaria

- Pruebas unitarias (backend)
- Pruebas de componentes (frontend)
- Pruebas end-to-end (Cypress)
- Cobertura > 90%
- Refactor de pruebas con IA

CLASE 8: Evaluación final de calidad + defensa de testing

Objetivos de la CLASE TUTORÍA:

- Validar que el sistema esté bien probado y documentado.
- Defender las decisiones de pruebas, cobertura y mejoras con ayuda de IA.

Actividades:

- Defensa del sistema y cobertura
- Auditoría cruzada entre pares
- Documentación final del testing + reportes

DevOps y Despliegue Profesional con Docker, GitHub Actions, Railway y Vercel

En este módulo, los estudiantes aprenderán a contenerizar sus aplicaciones con Docker, automatizar pruebas y despliegues con GitHub Actions, y publicar backend y frontend en plataformas reales. Este proceso completa el ciclo de vida del desarrollo profesional moderno. La IA será utilizada para crear archivos de configuración, detectar errores en despliegues, optimizar Dockerfiles y pipelines CI/CD.

CLASE 1: Fundamentos de DevOps, Docker y contenedores

Objetivos de aprendizaje:

- Comprender el ciclo DevOps, el concepto de contenedores y su rol en el despliegue moderno.
- Crear el primer Dockerfile para una aplicación Python.

Contenidos

- ¿Qué es DevOps y por qué es crucial?
- Contenedores vs Máquinas virtuales
- Instalación y configuración de Docker
- Creación de Dockerfile para FastAPI + PostgreSQL
- Comandos principales: build, run, compose, logs
- Asistencia IA:
 - Generación automática de Dockerfile
 - Optimización de imágenes y estructura

Ejercicio práctico:

- Dockerizar el backend del sistema de gestión universitaria con base de datos local.

CLASE 2: Debug y refactor de contenedores reales con IA

Objetivos de la CLASE TUTORÍA:

- Corregir errores comunes en Dockerfiles y contenedores mal configurados.
- Utilizar IA para sugerencias de optimización y rendimiento.

Actividades:

- Simulación de errores comunes de puerto, red y permisos
- Revisión con IA: limpieza de capas, caché, volúmenes
- Uso de Docker Compose para conectar servicios

CLASE 3: CI/CD moderno con GitHub Actions y despliegue en Railway

Objetivos de aprendizaje:

- Automatizar el flujo de pruebas, construcción y despliegue del backend.
- Desplegar una API real en Railway con base de datos remota.

Contenidos

- Introducción a GitHub Actions
- Workflows: test + build + deploy
- Variables de entorno seguras
- Conexión a Railway y PostgreSQL en la nube
- Herramientas IA como copiloto:
 - Generación de workflows YAML
 - Corrección de errores en pipelines

Ejercicio práctico:

- Automatizar despliegue continuo del backend a Railway desde GitHub.

CLASE 4: Validación de pipelines CI/CD reales**Objetivos de la CLASE TUTORÍA:**

- Validar el correcto funcionamiento del pipeline completo.
- Refactorizar workflows y conectar acciones condicionales.

Actividades:

- Simulación de errores y control de fallos
- Personalización de pasos y triggers
- Comparación entre workflows manuales vs generados por IA

CLASE 5: Despliegue del frontend en Vercel + conexión con API productiva**Objetivos de aprendizaje:**

- Publicar frontend moderno en Vercel y conectarlo al backend real.
- Administrar variables, seguridad y entornos correctamente.

Contenidos

- Conexión a GitHub desde Vercel
- Configuración de vite.config.js y .env.production
- Acceso seguro al backend mediante URL externa
- Herramientas IA:
 - Sugerencias de build scripts
 - Refactor de errores de conexión y carga

Ejercicio práctico:

- Desplegar el frontend React completo del sistema de gestión universitaria y conectarlo al backend de Railway.

CLASE 6: Revisión del sistema completo en producción**Objetivos de la CLASE TUTORÍA:**

- Validar todos los flujos del sistema en producción real.
- Simular usuarios reales, errores y recuperación.

Actividades:

- Auditoría: login, carga, navegación, edición, validaciones
- Comparación local vs producción
- Revisión final del sistema con IA

CLASE 7: Proyecto final DevOps + presentación del sistema publicado**Objetivos de aprendizaje:**

- Consolidar el ciclo profesional completo: construcción, pruebas, despliegue y presentación.

Proyecto final del módulo:**• Sistema completo desplegado:**

- Backend en Railway
- Frontend en Vercel
- Base de datos remota
- CI/CD con GitHub Actions
- Dockerizado y documentado

CLASE 8: Defensa técnica del sistema en producción + cierre de ciclo DevOps

Objetivos de la CLASE TUTORÍA:

- Validar dominio completo del flujo de desarrollo real.
- Recibir retroalimentación de docentes y pares sobre despliegue profesional.

Actividades:

- Presentación del sistema final en vivo (link público)
- Revisión técnica del código y CI/CD con IA
- Checklist de publicación, SEO, velocidad y errores

Módulo 4: Python Full Stack Developer – Advanced Systems Integration

- Proyecto final Full Stack profesional en un dominio real (universidad, hospital, empresa)
- Arquitectura avanzada con despliegue en producción
- Defensa técnica y construcción de portafolio profesional

Proyecto Final Profesional Full Stack + Defensa Técnica y Portafolio

Este módulo tiene como objetivo consolidar todo lo aprendido en los 11 módulos anteriores. Los estudiantes desarrollarán un proyecto Full Stack real, conectado, desplegado, probado y documentado, enfocado en un problema del mundo real. Al finalizar, presentarán su sistema, código, repositorio y documentación en una defensa técnica profesional, incluyendo checklist de calidad, funcionalidad, escalabilidad y buenas prácticas.

CLASE 1: Planeación del proyecto profesional y elección del dominio real

Objetivos de aprendizaje:

- Definir el alcance, objetivo y público objetivo de un sistema real.
- Diseñar la arquitectura técnica y el flujo general de desarrollo.

Contenidos

- Elección de dominio: hospital, universidad, clínica, ONG, inventario, tienda, sistema financiero
- Requisitos funcionales y técnicos
- Modelo entidad-relación
- Arquitectura Full Stack: Backend FastAPI + PostgreSQL | Frontend React + Tailwind
- Planificación por etapas
- Herramientas IA:
 - Asistencia en generación de modelo de datos
 - Sugerencias de diseño estructural y componentes

Entregables

- Documento de planificación con diagrama, entidades, funcionalidades
- Repositorio inicial y estructura base con readme

CLASE 2: Validación de planificación + generación de estructura con IA

Objetivos de la CLASE TUTORÍA:

- Revisar si el alcance del sistema es coherente y ejecutable en el tiempo.
- Generar la arquitectura base y carpetas con ayuda de IA.

Actividades:

- Revisión cruzada de alcances
- Generación de rutas, esquemas y modelos iniciales
- Planificación de tareas por equipo o persona

CLASE 3: Desarrollo del backend y base de datos conectada**Objetivos de aprendizaje:**

- Desarrollar todo el backend seguro, validado y documentado del sistema.
- Conectar con PostgreSQL real y preparar para despliegue.

Contenidos

- CRUD completos para entidades principales
- Relaciones entre tablas y validaciones con Pydantic
- Rutas protegidas con JWT
- Documentación Swagger y Redoc
- IA:
 - Refactor de rutas repetidas
 - Documentación automática
 - Validación de estructura y optimización

Entregables

- Backend funcional en producción (Railway o Render)
- Conexión verificada a base de datos externa

CLASE 4: Revisión del backend, pruebas, validaciones y documentación**Objetivos de la CLASE TUTORÍA:**

- Validar la seguridad, lógica y consistencia del backend.
- Aplicar pruebas unitarias a rutas clave y documentar endpoints.

Actividades:

- Testing manual y con Postman
- Generación de documentación final
- Verificación de errores, status codes y feedback limpio

CLASE 5: Desarrollo del frontend conectado a producción + control de roles**Objetivos de aprendizaje:**

- Construir una interfaz profesional conectada a APIs protegidas.
- Implementar roles, vistas privadas y formularios seguros.

Contenidos

- Login, dashboard, formularios, reportes
- Navegación protegida con React Router
- Integración con Vercel
- Tailwind para visuales modernos
- Herramientas IA:
 - Refactor de hooks, formularios, componentes
 - Mejora de UX/UI

Entregables

- Sistema visual completo conectado a producción.

CLASE 6: Pruebas de usuario, depuración visual y checklist visual

Objetivos de la CLASE TUTORÍA:

- Verificar experiencia del usuario final, errores de carga y comportamiento UI.
- Refactorizar visualmente con IA.

Actividades:

- Simulación de uso: admins, usuarios
- Revisión de responsive, carga, feedback visual
- Generación de documentación visual del sistema

CLASE 7: Defensa del proyecto, entrega final y publicación profesional

Objetivos de aprendizaje:

- Presentar el sistema de forma profesional, técnica y funcional.
- Preparar repositorio, presentación y checklist para entrevista laboral.

Contenidos

- Pitch técnico del sistema
- Presentación ante jurado o docente
- Checklist de revisión profesional:
 - Seguridad
 - Documentación
 - Código limpio
 - Testing
 - CI/CD
 - Responsive
 - Despliegue

Entregables Finales:

- Repositorio GitHub con readme profesional
- Video de defensa técnica (opcional)
- Sistema Full Stack desplegado y accesible
- Documentación + checklist de calidad

CLASE 8: Retroalimentación profesional + Portafolio y LinkedIn

Objetivos de la CLASE TUTORÍA:

- Recibir retroalimentación final sobre el proyecto.
- Publicar el sistema y posicionarlo en su portafolio digital.

Actividades:

- Revisión en vivo de proyectos
- Feedback profesional
- Publicación en LinkedIn + repositorio
- Cierre del Programa con reflexión final

Conclusiones.

Python Senior AI: La Ruta Maestra del Desarrollo Real; no es solo un Programa, es una experiencia de transformación profesional. A lo largo de 4 módulos intensivos, con acompañamiento experto, integración de inteligencia artificial y desarrollo de proyectos reales, te convertirás en un desarrollador capaz de crear soluciones técnicas sólidas, modernas y con impacto real.

Te formarás en backend con FastAPI, frontend con React, bases de datos con PostgreSQL, testing, DevOps, CI/CD, y despliegue profesional, pero también desarrollarás habilidades críticas como pensamiento estratégico, trabajo colaborativo, documentación técnica, comunicación global y autonomía con IA.

Al finalizar esta travesía:

- Habrás diseñado, programado y desplegado sistemas reales aplicados a universidades, empresas, hospitales y organizaciones del mundo real.
- Tendrás un portafolio profesional en GitHub, con código limpio, documentado, probado y publicado en producción.
- Dominarás herramientas IA, Docker, GitHub Actions, Railway y Vercel, utilizadas en equipos profesionales de desarrollo.
- Estarás listo para enfrentar entrevistas técnicas, retos laborales globales y procesos de selección en inglés gracias al programa English Boost.
- Y lo más importante: tendrás la confianza, el criterio técnico y la visión para trabajar como un verdadero Desarrollador Python Senior Full Stack en la industria tecnológica global.

Porque hoy el mundo no necesita más personas que solo sepan programar. Necesita mentes que piensen, construyan, lideren e innoven con propósito, herramientas modernas y visión profesional.

Dev Senior Code



“Tu ruta ya está trazada. El futuro empieza contigo.”
La industria te está esperando.