



ISEIE
ISEIE INNOVATION SCHOOL

BROCHURE
CURSO DE
PROGRAMACIÓN



 www.iseie.com

ÍNDICE DEL CURSO

- 03 **CURSO DE PROGRAMACIÓN**
- 04 **POR QUÉ REALIZAR EL CURSO**
- 05 **OBJETIVOS**
- 06 **PARA QUÉ TE PREPARA EL CURSO**
- 07 **DISEÑO Y CONTENIDO**
- 08 **REQUISITOS DE POSTULACIÓN**
- 09 **TITULACIÓN PROPIA**
- 10 **TRABAJO DE FIN DEL CURSO**
- 12 **CONTENIDO DEL CURSO**
- 15 **UBICACIÓN Y CONTACTO**

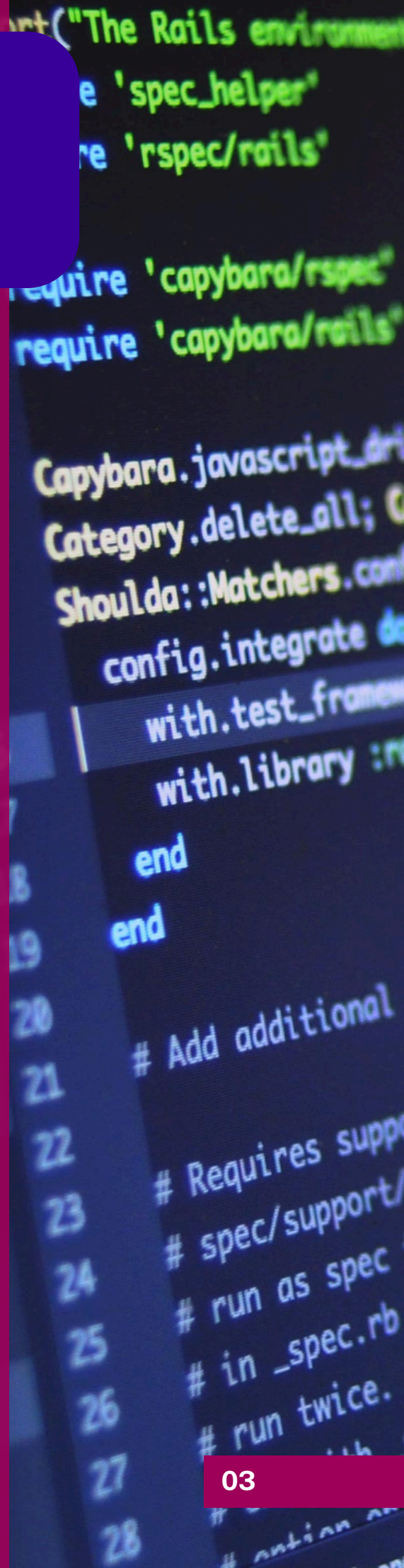


CURSO DE PROGRAMACIÓN

Bienvenido al curso de programación, una formación diseñada para llevarte paso a paso desde los conceptos básicos de la programación hasta habilidades prácticas y avanzadas. Este curso es el punto de partida ideal para todos aquellos que desean aprender a programar, independientemente de su experiencia previa, y construir una base sólida en el mundo del desarrollo de software.

A lo largo del curso, exploraremos una variedad de paradigmas de programación, plataformas y lenguajes como Python, JavaScript, Java, C#, y otros, a medida que adentramos en la estructura y funcionamiento de los lenguajes más utilizados en la industria actual. Te enseñaremos cómo pensar como un programador, cómo diseñar soluciones a problemas mediante algoritmos y cómo elegir el lenguaje adecuado según las necesidades del proyecto, ya sea para desarrollo web, móvil, data/IA, automatización, entre otros.

Este curso se basa en la resolución de problemas mediante ejercicios y mini-proyectos guiados que te ayudarán a dominar las técnicas fundamentales. Comenzarás con la configuración de tu entorno de desarrollo en VS Code, aprenderás a escribir tu primer código en Python, y progresivamente aplicarás tus conocimientos a través de técnicas de programación estructurada, modular y orientada a objetos. Además, conocerás las mejores prácticas en calidad de código, pruebas automatizadas, gestión de versiones con Git y cómo colaborar eficientemente en un equipo de desarrollo.



POR QUÉ REALIZAR EL CURSO



El realizar un curso eleva el conocimiento y nivel académico de la persona, convirtiéndola en un elemento fundamental dentro de un esquema de trabajo; su trascendencia radica en el desarrollo de competencias adicionales que adquiere, su proceso formativo se vuelve más sólido y por ende se convierte en un candidato más atractivo para cubrir un puesto preponderante.



Te brinda la oportunidad de adquirir conocimientos actualizados y estar al tanto de las últimas tendencias y avances en tu área de interés. Realizar un curso en un área que te apasiona puede brindarte una gran satisfacción personal. Te permite profundizar en un tema que te interesa y te da la oportunidad de contribuir de manera significativa en ese campo.



OBJETIVOS



Los estudios de postgrado consisten no solo en adquirir conocimientos por parte del participante.



Sino que estos queden supeditados al desarrollo de una serie de competencias en función de los perfiles académicos y los correspondientes perfiles profesionales.



Nuestra función es centrar los objetivos de este programa y los diferentes módulos que lo conforman no solamente en la simple acumulación de conocimientos.



Conocimientos sino también en las hard skills y soft skills que permitan a los profesionales desempeñar su labor de forma exitosa en este mundo laboral en constante evolución.

PARA QUÉ TE PREPARA EL CURSO

Este curso está diseñado para proporcionarte una base sólida en programación y desarrollo de software, preparándote para enfrentar los desafíos y tareas que los profesionales de la tecnología manejan a diario. Te preparará para:

- A Entender los conceptos fundamentales de la programación:** Aprenderás los pilares de la programación, como algoritmos, estructuras de datos, y las diferentes formas de organizar el código (paradigmas estructurados, modulares y orientados a objetos).
- B Desarrollar habilidades prácticas en varios lenguajes de programación:** Te familiarizarás con lenguajes populares como Python, JavaScript, Java, C# y SQL, y conocerás sus aplicaciones en diferentes áreas como desarrollo web, móvil, backend, automatización y más.
- C Crear proyectos reales:** Aplicarás lo aprendido en mini-proyectos prácticos, desde scripts sencillos hasta proyectos más complejos, utilizando herramientas como Git, GitHub, VS Code, y entornos virtuales para gestionar tu código y colaborar con otros.



PARA QUÉ TE PREPARA EL CURSO

- D Adquirir una comprensión profunda de la calidad de código:** Aprenderás a escribir código limpio, seguir buenas prácticas como PEP8, gestionar dependencias, y realizar pruebas automatizadas utilizando herramientas como pytest.
- E Dominar herramientas esenciales del desarrollo moderno:** Te familiarizarás con herramientas de desarrollo como VS Code, Git, y aprenderás a trabajar en equipo con metodologías ágiles, abordando tareas como la gestión de versiones y la documentación de tus proyectos.
- F Prepararte para un rol en equipos de desarrollo:** Comprenderás los diferentes roles en un equipo de desarrollo, como desarrolladores, QA y DevOps, y cómo colaborar efectivamente dentro de un flujo de trabajo ágil.
- G Integrar la Inteligencia Artificial en tu flujo de trabajo:** Aprenderás a trabajar con asistentes de código basados en IA, mejorando tu eficiencia en el desarrollo y depuración de proyectos.



DISEÑO Y CONTENIDO

01

Para el diseño del Plan de estudios de este curso, ISEIE Innovation School ha seguido las directrices del equipo docente, el cual ha sido el encargado de seleccionar la información con la que posteriormente se ha constituido el plan de estudio.



02

De esta forma, el profesional que acceda al programa encontrará el contenido más vanguardista y exhaustivo relacionado con el uso de procesos innovadores y altamente eficaces, conforme a las necesidades y problemáticas actuales.



Buscando la integración de conocimientos académicos y de formación profesional, en un ambiente competitivo y globalizado. Todo ello a través de cada uno de sus módulos de estudio presentado en un cómodo y accesible formato 100% online.



03



El empleo de la metodología Relearning en el desarrollo de este programa te permitirá fortalecer y enriquecer tus conocimientos y hacer que perduren en el tiempo a base de una reiteración de contenidos.

04

REQUISITOS DE POSTULACIÓN

Para postular a nuestro curso debes cumplir con los siguientes requisitos:



Documento de identidad



Correo electrónico

A QUIÉN ESTÁ DIRIGIDO

Principiantes en programación que deseen comprender los fundamentos de la programación y desarrollar sus habilidades desde cero.

Desarrolladores en formación que buscan una estructura organizada para profundizar en los principios de la programación y mejorar su capacidad de resolución de problemas.

Profesionales de áreas afines (como diseñadores, analistas o ingenieros) que desean adquirir habilidades técnicas en programación para ampliar sus capacidades y aportar a proyectos tecnológicos.



TITULACIÓN PROPIA



Al concluir el curso los participantes serán galardonados con una titulación propia otorgada por ISEIE Innovation School. Esta titulación se encuentra respaldada por una certificación que equivale a 4 créditos ECTS (European Credit Transfer and Accumulation System) y representa un total de 100 horas de dedicación al estudio.



Esta titulación no solo enriquecerá su imagen y credibilidad ante potenciales clientes, sino que reforzará significativamente su perfil profesional en el ámbito laboral. Al presentar esta certificación, podrá demostrar de manera concreta y verificable su nivel de conocimiento y competencia en el área temática del curso.



Esto resultará en un aumento de su empleabilidad, al hacerle destacar entre otros candidatos resaltando su compromiso con la mejora continua y el desarrollo profesional.



TRABAJO FINAL DEL CURSO

A

Una vez que haya completado satisfactoriamente todos los módulos del master, deberá llevar a cabo un trabajo final en el cual deberá aplicar y demostrar los conocimientos que ha adquirido a lo largo del programa.

B

Este trabajo final suele ser una oportunidad para poner en práctica lo que ha aprendido y mostrar su comprensión y habilidades en el tema.

C

Puede tomar la forma de un proyecto, un informe, una presentación u otra tarea específica, dependiendo del contenido del curso y sus objetivos. Recuerde seguir las instrucciones proporcionadas y consultar con su instructor o profesor si tiene alguna pregunta sobre cómo abordar el trabajo final.



CONTENIDO DEL CURSO DE PROGRAMACIÓN

MÓDULO 1: INTRODUCCIÓN A LA PROGRAMACIÓN (SIN CÓDIGO)

1.1 ¿Qué es programar? Problemas, algoritmos y modelos mentales

1.1.1 Paradigmas: estructurada, modular y POO (ventajas/limitaciones)

1.1.2 Plataformas y dominios: web, escritorio, móvil, data/IA, automatización

1.2 Lenguajes y usos comunes

1.2.1 Python: scripting, datos, backend, automatización

1.2.2 JavaScript: navegador, SPA, Node.js

1.2.3 Java/Kotlin: Android, backend Empresarial

1.2.4 C#: Windows, Unity, backend

1.2.5 SQL: persistencia de datos (transversal)

1.3 Arquitectura básica: cliente/servidor, API, base de datos (visión general)

1.4 Ruta del curso: por qué Python como eje

1.5 Roles en un equipo (dev, QA, DevOps) y flujo de trabajo

1.6 Ética, licencias y buenas prácticas desde el día 0

MÓDULO 2: ENTORNO Y EDITOR (PRÁCTICO, CERO LÓGICA AÚN)

2.1 Editor vs. IDE (recomendación: VS Code)

2.2 Instalación y primera configuración (tema, fuente, idioma, atajos)

2.3 Extensiones mínimas: Python, Pylance, Git, Jupyter, Black/Flake8, Live Server

2.4 Terminal integrada, venv y pip (crear/activar, requirements.txt)

2.5 Git y GitHub: init → commit → push (sin conflicto aún)

2.6 Depuración en VS Code: breakpoints y variables (demo guiada)

2.7 Variables de entorno y .env (no exponer secretos)

2.8 Estructura mínima de proyecto y README.md

2.9 Plantilla "Setup Pro":

.vscode/settings.json, linters, pytest habilitado

2.10 Mini-proyecto + Vibe clip: crear repo plantilla y explicar tu setup (2-3 min)

```
return (  
  <React.Fragment>  
    <div className="py-5">  
      <div className="container">  
        <Title name="our" title="product">  
          <div className="row">  
            <ProductConsumer>  
              {(value) => {  
                console.log(value)  
              }}  
            </ProductConsumer>  
          </div>  
        </div>  
      </div>  
    </React.Fragment>  
  )  
)
```

MÓDULO 3: PROGRAMACIÓN ESTRUCTURADA (EMPEZAMOS A CODEAR)

3.1 Secuencia, selección e iteración en Python

3.2 Variables, tipos y operadores (repaso práctico)

3.3 Entrada/salida y argumentos CLI

3.4 Funciones puras vs. con efectos; parámetros y retorno

3.5 Depurar: tracebacks y logging básico

3.6 Errores típicos y cómo leerlos

3.7 Práctica guiada: mini utilidades de consola

3.8 Lectura/escritura de archivos (txt/csv)

3.9 Documentar con docstrings

3.10 Mini-proyecto + Vibe clip: script CLI que procesa un archivo

MÓDULO 4: ESTRUCTURAS DE DATOS (EN PYTHON)

4.1 Listas y tuplas (slicing, mutabilidad)

4.2 Diccionarios y conjuntos (operaciones de conjunto)

4.3 Pilas/colas (list/deque)

4.4 Comprensiones y generadores

4.5 Búsqueda: lineal y binaria

4.6 Ordenación: burbuja, inserción y selección

4.7 Recursión: casos base

4.8 Complejidad (Big-O) con mediciones simples

4.9 Tests intro con pytest (primeros asserts)

4.10 Mini-proyecto + Vibe clip: analizador de datos con ranking/filtrado

MÓDULO 5: PROGRAMACIÓN MODULAR Y CALIDAD

- 5.1 Modularidad: cohesión y acoplamiento
- 5.2 Módulos y paquetes en Python; namespaces
- 5.3 Gestión de dependencias (requirements.txt)
- 5.4 Manejo de excepciones (try/except/else/finally)
- 5.5 Estilo: PEP8, black, flake8 (guardar = formatear)
- 5.6 Testing con pytest: estructura, fixtures básicos
- 5.7 Automatizar tareas frecuentes (scripts de proyecto)
- 5.8 Git práctico: ramas, merges y resolución de conflictos
- 5.9 Checklist de calidad antes de subir código
- 5.10 Mini-proyecto + Vibe clip: utilidades modulares con tests

MÓDULO 6: PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

- 6.1 Objetos y clases (responsabilidades)
- 6.2 Atributos, métodos, init , repr
- 6.3 Encapsulamiento y @property
- 6.4 Composición vs. herencia (criterios)
- 6.5 Polimorfismo y duck typing
- 6.6 Abstracción (abc)
- 6.7 dataclasses y organización de paquetes OO
- 6.8 Tests de clases y casos de uso
- 6.9 Refactor: de funciones a objetos
- 6.10 Mini-proyecto + Vibe clip: micro-librería OO con README



MÓDULO 7: IA PARA PROGRAMAR + VIBE CODING (BÁSICO)

- 7.1 Asistentes de código: alcance, límites y ética
- 7.2 Autocompletado vs. copilotos/chat (flujo en el editor)
- 7.3 Prompts efectivos: pedir ejemplos, refactors, docstrings y tests
- 7.4 Depurar con IA: describir error → propuesta de fix → test de regresión
- 7.5 "Vibe Coding": cápsulas 8-12 min con un objetivo único
- 7.6 Guion y buenas prácticas de grabación (1080p/30fps)
- 7.7 Publicar clips y organizar repos (tags por checkpoint)
- 7.8 Privacidad y datos sensibles en demos
- 7.9 Rubrica de evaluación de clips
- 7.10 Mini-proyecto: cápsula 5-7 min codificando una función + tests (repo público)



MÓDULO 8: TRABAJO FINAL DE CURSO (TFC)

- 8.1 Definición del problema y alcance
- 8.2 Diseño simple (flujo, datos, si aplica endpoints)
- 8.3 Tablero e issues; roadmap
- 8.4 Desarrollo iterativo (MVP + mejoras)
- 8.5 Pruebas mínimas y validación del flujo principal
- 8.6 README con instrucciones reproducibles
- 8.7 (Opcional) Endpoint sencillo con Flask/FastAPI si el proyecto lo requiere
- 8.8 Despliegue básico (Render/Railway) o demo local bien documentada
- 8.9 Demo Vibe 3-5 min
- 8.10 Presentación final y retro




Nota: El contenido del programa académico puede estar sometido a ligeras modificaciones, en función de las actualizaciones o de las mejoras efectuadas.



ISEIE
ISEIE INNOVATION SCHOOL

CONTÁCTANOS

 +34 960 25 47 46

 Av. Aragón 30, 5. 46021 Valencia.

 www.iseie.com