Aspectos generales

Título:	Programación y Modelado Computacional usando Python
Programas de posgrado o planes de estudio en donde se ofertará adicionalmente:	
Ciencias Biomédicas, Bioquímica y Biológicas	
Área del conocimiento:	Genética, genómica y bioinformática
Semestre:	2027-1
Modalidad:	Curso fundamental
Horario:	Martes: 9:00-11:00 hrs, Jueves: 9:00-11:00 hrs
No. sesiones:	20
Horas por sesión:	2.0
Total alumnos PDCB:	15
Total alumnos:	10
Videoconferencia:	Si
Lugar donde se imparte:	Unidad de Posgrado UNAM
Informes:	fersanp@gmail.com

Métodos de evaluación

MÉTODO	PORCENTAJE	NOTAS
Bitácoras (Evaluación Continua)	30%	Reportes semanales. La evaluación priorizará la capacidad de identificar errores (debugging), fundamentar las correcciones aplicadas y explicar la lógica utilizada en la resolución de problemas de sintaxis o estructuración de datos.
Proyecto Final	40%	Desarrollo de un pipeline de análisis, automatización o modelo de machine learning aplicado a los datos de investigación de posgrado del estudiante. Se evaluará mediante la defensa técnica de las decisiones algorítmicas, la limpieza de datos y el análisis
Revisión de Código (Revisiones Técnicas)	30%	Evaluaciones orales aleatorias durante las sesiones de taller. Se explicará detalladamente la lógica de control, las estructuras de datos o los modelos predictivos de sus scripts.

Contribución de este curso/tópico en la formación del alumnado del PDCB:

La programación es clave para desarrollar en los estudiantes un pensamiento computacional que les permita abordar problemas científicos de manera lógica y estructurada, independientemente del lenguaje de programación. Esta habilidad transversal fomenta la capacidad de descomponer problemas complejos, identificar patrones y crear algoritmos efectivos, Además, permite el análisis eficiente de grandes volúmenes de datos experimentales, automatizar procesos y optimizar la investigación de los estudiantes. Capacitando al alumnado en el uso de herramientas computacionales fundamentales.

Profesor (a) responsable

Nombre:	Sánchez Puig María Fernanda
Teléfono:	
Email:	fernanda@ciencias.unam.mx

Profesores (as) participantes

PARTICIPANTE	ENTIDAD O ADSCRIPCIÓN	SESIONES

SÁNCHEZ PUIG MARÍA FERNANDA Responsable	Otras entidades	Análisis exploratorio y agrupaciones Automatización y conexión con APIs Bucles e iteraciones Clasificación y predicción (Modelos Supervisados). Control de flujo (Condicionales) Diccionarios y mapeo de datos Funciones personalizadas Introducción a dataframes (Pandas) e importación de datos Introducción al entorno de trabajo y la lógica computacional. Introducción al Machine Learning Limpieza y estructuración de datos Listas y manipulación de secuencias. Presentación de Proyectos (Parte 1). Presentación de Proyectos (Parte 2) y Cierre Procesamiento de texto. Reproducibilidad y control de versiones Taller de Auditoría de Modelos Taller de Debugging Guiado Variables y tipos de datos. Visualización de datos (Matplotlib/Seaborn).

Introducción

En la actualidad, la investigación en ciencias biomédicas genera volúmenes masivos de datos procedentes de secuencias genómicas, registros clínicos y experimentación en laboratorio. Para procesar, analizar y extraer significado de esta información, la programación computacional se ha consolidado como una herramienta metodológica indispensable. Aunado a esto, la introducción de las herramientas de Inteligencia Artificial (IA) generativa ha transformado radicalmente el entorno de la programación. El paradigma ha cambiado: el reto principal para el investigador ya no es memorizar sintaxis, sino desarrollar un pensamiento lógico robusto que le permita dirigir, evaluar y corregir algoritmos.
Este curso está diseñado para introducir al alumnado en la programación computacional (utilizando Python) partiendo desde cero, con un enfoque dirigido a la resolución de problemas y al análisis de datos. A través de un formato de taller práctico, el curso aborda el aprendizaje de la programación integrando a la IA no como un atajo para resolver tareas, sino como un asistente de investigación colaborativo. Se enfatiza el desarrollo del pensamiento crítico computacional, enseñando al estudiante a analizar el código generado por IA, identificar sesgos de diseño, evitar la fuga de datos y asegurar la reproducibilidad de métodos exigida en la investigación científica.

Objetivo General: Capacitar al alumnado de posgrado en el desarrollo de competencias lógico-computacionales para el diseño y automatización de flujos de trabajo aplicados al análisis de datos en proyectos de investigación.

Objetivos Específicos:
1. Alfabetización Computacional: Comprender la lógica algorítmica y la sintaxis fundamental de programación para la gestión de variables, secuencias y estructuras de control, facilitando el acceso mediante entornos de desarrollo en la nube.
2. Manipulación de Datos a Gran Escala: Implementar técnicas de limpieza, estructuración, procesamiento y visualización de conjuntos de datos masivos.
3. Inteligencia Artificial: Desarrollar la capacidad analítica para el empleo de modelos de lenguaje (LLMs) como asistentes colaborativos, permitiendo la identificación de errores lógicos, alucinaciones y sesgos metodológicos en el código generado.
4. Automatización y Análisis Predictivo: Implementar modelos fundamentales de aprendizaje automático (Machine Learning) y procesos de automatización vinculados directamente a las bases de datos y objetivos de las tesis de maestría y doctorado de los estudiantes.
Temario:

Temario

Temario:
Módulo 1: Lenguajes de Programación (10 horas). Comprender la lógica de programación base.
? Sesión 1: Introducción al entorno de trabajo y la lógica computacional. Implementación de herramientas como Google Colab para facilitar el acceso sin instalaciones locales. Definición de algoritmos y primer acercamiento a la sintaxis del lenguaje.
? Sesión 2: Variables y tipos de datos. Manejo de valores numéricos, cadenas de caracteres y tipos booleanos.
? Sesión 3: Control de flujo (Condicionales). Uso de las sentencias if, elif y else para la toma de decisiones lógica.
? Sesión 4: Bucles e iteraciones (for y while). Estrategias para la automatización de procesos redundantes y el recorrido eficiente de secuencias de datos.
? Sesión 5: Funciones personalizadas. Implementación de la modularidad en el desarrollo de software..

Módulo 2: Estructuras de Datos (8 horas). Manejo de colecciones de datos.
? Sesión 6: Listas y manipulación de secuencias. Técnicas de indexación y segmentación (slicing) de arreglos de datos.
? Sesión 7: Diccionarios y mapeo de datos. Organización de información mediante pares llave-valor.
? Sesión 8: Procesamiento de texto. Análisis de frecuencia, cálculo de entropía y detección de patrones en cadenas.
? Sesión 9: Taller de Debugging Guiado. Identificación y resolución de errores programados en scripts.

Módulo 3: Manipulación de Datos Reales (10 horas). Integración de bibliotecas avanzadas como Pandas.
? Sesión 10: Introducción a dataframes (Pandas) e importación de datos. Procedimientos para la carga de archivos en formatos .csv y .xlsx provenientes de laboratorios o repositorios científicos.
? Sesión 11: Limpieza y estructuración de datos. Gestión de registros nulos y aplicación de filtros.
? Sesión 12: Análisis exploratorio y agrupaciones. Generación de estadísticas descriptivas aplicadas a conjuntos de datos masivos.
? Sesión 13: Visualización de datos (Matplotlib/Seaborn). Diseño de histogramas, diagramas de caja y dispersión.

? Sesión 14: Reproducibilidad y control de versiones. Documentación de flujos de trabajo híbridos.

Módulo 4: Análisis Predictivo y Automatización de Modelos (8 horas). Integración de herramientas de optimización de flujos de trabajo.
? Sesión 15: Introducción al Machine Learning.
? Fundamentos de Scikit-Learn y técnicas para la simplificación de variables.
? Implementación de Análisis de Componentes Principales (PCA) para la visualización y análisis de conjuntos de datos biomédicos de alta complejidad en entornos de baja dimensión.
? Sesión 16: Clasificación y predicción (Modelos Supervisados). Entrenamiento de algoritmos de regresión logística y árboles de decisión.
? Estrategias de validación mediante la segmentación de datos en conjuntos de entrenamiento y prueba (Train/Test split).
? Sesión 17: Automatización y conexión con APIs. Desarrollo de procesos para la gestión masiva de información.
? Construcción de scripts para la descarga automatizada de registros desde repositorios públicos y optimización del procesamiento de archivos.
? Sesión 18: Taller de Auditoría de Modelos. Evaluación técnica de arquitecturas.

Módulo 5: Proyecto Final (4 horas). Fase de evaluación centrada en la sustentación del código desarrollado y su relevancia para la investigación de posgrado.
? Sesión 19: Presentación de Proyectos (Parte 1). Exposición del pipeline computacional y defensa técnica de la lógica implementada.
? Sesión 20: Presentación de Proyectos (Parte 2) y Cierre. Evaluación.

Bibliografía

- Learning Python 5th Edition. Mark Lutz, O'Reilly. 2013
- Python programming: A step-by-step guide to learning the language. C. K. Dhaliwal, Poona Rana and T.P.S. Brar. CRC Press. 2024.
- Practical Programming, Paul Gries, 2013, The Pragmatic Programmers
- Python Programming for Biology: Bioinformatics and Beyond. Stevens, T. J., & Boucher, W. (2015). Cambridge University Press.
- Clean Code: A Handbook of Agile Software Craftsmanship. Martin, R. C. (2008). Prentice Hall
- Bioinformatics with Python Cookbook: Learn how to use modern Python bioinformatics libraries and applications (3ra ed.). Antao, T. (2022). Packt Publishing.

Observaciones

- No se requiere que el estudiante tenga experiencia avanzada en programación.
- Se requiere que el estudiante cuente con una computadora personal, de preferencia laptop, para su uso en las sesiones y para la realización de tareas y proyectos.