



Clases
Virtuales
En vivo

OTI  **UNI**

PROGRAMA DE ESPECIALIZACIÓN

CIENCIA DE DATOS CON PYTHON

EDICIÓN: IV

DESCRIPCIÓN

El programa de Ciencia de Datos con Python tiene como objetivo proporcionar a los participantes las habilidades y conocimientos necesarios para realizar un análisis exhaustivo de conjuntos de datos, utilizando técnicas de exploración visual y estadísticas descriptivas, así como para aplicar algoritmos de aprendizaje automático para la construcción y evaluación de modelos predictivos y clasificadores.

Al final del programa, los estudiantes estarán capacitados para comprender, explorar y analizar datos de manera efectiva, así como para implementar soluciones de aprendizaje automático utilizando Python en diversos contextos y aplicaciones.



PÚBLICO OBJETIVO

- Estudiantes de ciencias de la computación o campos relacionados: Que deseen complementar su formación académica con habilidades prácticas en análisis de datos.
- Analistas de datos: Aquellos que deseen mejorar sus habilidades en el análisis exploratorio de datos y en la construcción de modelos predictivos.
- Científicos de datos: Profesionales que buscan ampliar sus conocimientos en el procesamiento y análisis de datos.
- Desarrolladores de software: Interesados en incorporar capacidades de análisis de datos y aprendizaje automático en sus aplicaciones.
- Profesionales de marketing y negocios: Que buscan utilizar técnicas de análisis de datos y aprendizaje automático para tomar decisiones informadas.



¿QUÉ APRENDERÁS EN ESTE PROGRAMA?

- Fundamentos de la visualización de datos y su importancia.
- Técnicas de análisis exploratorio de datos y estadísticas descriptivas.
- Uso de bibliotecas como Matplotlib, Seaborn y Plotly para crear visualizaciones efectivas.
- Implementación de algoritmos de aprendizaje automático como regresión lineal, regresión logística y árboles de decisión.
- Manejo y transformación de datos en formatos como CSV, JSON, XML y Excel.



REQUISITOS ACADÉMICOS

- Conocimiento básico de informática y acceso a una computadora con internet.
- No se requiere experiencia previa en programación, ya que el curso está diseñado para principiantes y proporcionará los conocimientos y habilidades necesarias para comenzar a programar y analizar datos utilizando Python.



CERTIFICACIÓN

1. Certificado Digital

Al haber aprobado el programa con un **promedio ponderado mayor ó igual a 14**, se le otorga al participante un Certificado de aprobación a nombre de la Universidad Nacional de Ingeniería.

2. Constancia de Asistencia

Al participante que no cumpla con los requisitos de certificación, se le otorgará una Constancia de Asistencia del Curso, **para lo cual el alumno deberá contar con una asistencia a clase mínima del 75%**. En el caso de no cumplir con dicho requerimiento no se emitirá dicha Constancia.

EVALUACIÓN

La nota del programa se obtendrá de la siguiente manera:

El programa está estructurado de la siguiente manera: Cada módulo será evaluado con un nota. El promedio final se calculará sumando las notas de los cuatro módulos ($N1 + N2 + N3 + N4$) y dividiendo el total entre 4.

La asistencia del curso se obtendrá de la siguiente manera:

La asistencia a cada sesión se apertura automáticamente **en la plataforma Virtual durante el horario de la clase.**



PLAN DE ESTUDIOS

MÓDULO 1 - FUNDAMENTOS DE PROGRAMACIÓN CON PYTHON

ENFOQUE: En este módulo, se enseñan los principios básicos de la programación utilizando Python como lenguaje principal. Al finalizar, los estudiantes podrán resolver problemas simples y continuar su formación en programación.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
1	Introducción a Python y fundamentos de programación.	¿Qué es Python y sus características principales. - Instalación de Python, Visual Studio Code y uso de Google Colab. - Concepto de algoritmo y pensamiento lógico. - Buenas prácticas de programación y sintaxis básica. - Funciones integradas y módulos estándar de Python. - Usos del comando print. - Ejercicios de aplicación.
2	Sentencias de Control, cadenas y tuplas	- Introducción a las sentencias de control. - Sentencias condicionales y estructuras if, elif, else. - Sentencias de repetición: for y while. - Sentencias de control anidadas. - Ejercicios prácticos de bucles y control de iteración. - Introducción a las cadenas. - Creación y manipulación de cadenas. - Métodos útiles para cadenas. - Iteración sobre cadenas. - Introducción a las tuplas. - Creación y manipulación de tuplas. - Ejercicios prácticos con cadenas y tuplas.
3	Listas.	- Introducción a las listas. - Creación y manipulación de listas. - Métodos útiles para listas. - Listas anidadas. - Uso de funciones con listas. - Ejercicios prácticos con listas.
4	Diccionarios.	- Introducción a los diccionarios. - Creación y manipulación de diccionarios. - Acceso y modificación de elementos. - Iteración sobre diccionarios. - Uso de funciones con listas. - Ejercicios prácticos con diccionarios.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
5	▪ Funciones en Python.	▪ Introducción a las funciones en Python. ▪ Definición y sintaxis básica (def, return). ▪ Parámetros y argumentos. ▪ Variables locales y globales. ▪ Documentación y comentarios dentro de funciones. ▪ Ejercicios prácticos aplicados a listas y diccionarios.
6	▪ Archivos de texto.	▪ Manipulación avanzada de archivos de texto. ▪ Uso de la sentencia with para manejar archivos de manera segura. ▪ Procesamiento de texto: búsqueda, reemplazo y manipulación de cadenas. ▪ Ejercicios prácticos de manipulación avanzada de archivos de texto.
7	▪ Manejo de Errores y Excepciones.	▪ Introducción al manejo de errores en Python. ▪ Tipos comunes de errores y excepciones. ▪ Uso de las estructuras try, except, else y finally. ▪ Generación de excepciones personalizadas con raise. ▪ Buenas prácticas en el manejo de errores. ▪ Ejercicios prácticos de control de errores con archivos.
TOTAL DE HORAS		21

MÓDULO 2 - TRANSFORMACIÓN DE DATOS CON PYTHON

ENFOQUE: En este módulo se enseñan habilidades para transformar datos eficientemente usando Numpy y Pandas. Los estudiantes aprenderán a manipular, limpiar y analizar datos para extraer información valiosa.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
1	▪ Introducción a ETL y Fundamentos de Python para Datos.	▪ Introducción a ETL: Concepto, fases y su importancia. ▪ Casos de uso y arquitectura básica de un pipeline ETL. ▪ Ventajas de Python en procesos ETL. ▪ Repaso de Python esencial: Tipos de datos, variables y operadores. ▪ Estructuras de control: if, for, while. ▪ Funciones y manejo básico de módulos. ▪ Manejo de archivos (CSV, TXT) con Python estándar. ▪ Configuración del Entorno: • Instalación de Anaconda/Miniconda. • Uso de Jupyter Notebooks/JupyterLab. • Ejercicios prácticos de repaso.
2	▪ Programación Orientada a Objetos Aplicada a Datos (POO Avanzada).	▪ Introducción a la Programación Orientada a Objetos (POO): clases, atributos y métodos ▪ Principios básicos: encapsulamiento, herencia y reutilización. ▪ Creación y uso de clases personalizadas. ▪ Métodos especiales (<code>__init__</code>, <code>__str__</code>). ▪ Aplicación práctica: clase para representar un conjunto de datos (Dataset o Archivo). ▪ Integración con lectura y escritura de archivos. ▪ Ejercicios prácticos: mini proyecto de lectura de datos usando clases.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
3	▪ NumPy para Manipulación Numérica Eficiente	▪ Introducción a NumPy y ventajas del uso de arrays. ▪ Creación de arrays: <code>array()</code>, <code>zeros()</code>, <code>ones()</code>, <code>arange()</code>, <code>linspace()</code>. ▪ Atributos de arrays: <code>shape</code>, <code>dtype</code>, <code>ndim</code>, <code>size</code>. ▪ Indexación y slicing de arrays. ▪ Acceso a elementos, filas y columnas. ▪ Indexación booleana y fancy indexing. ▪ Operaciones aritméticas elemento a elemento. ▪ Funciones universales (<code>sqrt()</code>, <code>exp()</code>, <code>sin()</code>, etc.). ▪ Agregaciones: <code>sum()</code>, <code>mean()</code>, <code>std()</code>, <code>min()</code>, <code>max()</code>. ▪ Concepto de broadcasting y aplicaciones. ▪ Reestructuración de arrays: <code>reshape()</code>, <code>transpose()</code>, <code>flatten()</code>, <code>ravel()</code>. ▪ Apilamiento de arrays: <code>vstack()</code>, <code>hstack()</code>, <code>concatenate()</code>.
4	▪ Pandas – Series y DataFrames (Extracción y Carga).	▪ Introducción a Pandas: concepto de Series y DataFrame. ▪ Creación de Series y DataFrames desde listas, diccionarios o arrays de NumPy. ▪ Atributos de un DataFrame: <code>index</code>, <code>columns</code>, <code>values</code>, <code>dtypes</code>, <code>shape</code>. ▪ Carga de datos (Extracción): <code>read_csv()</code>, <code>read_excel()</code>, <code>read_json()</code>. ▪ Consideraciones al cargar datos: separadores, encabezados, tipos de datos. ▪ Exploración inicial: <code>head()</code>, <code>tail()</code>, <code>sample()</code>, <code>info()</code>, <code>describe()</code>, <code>value_counts()</code>. ▪ Guardado de datos: <code>to_csv()</code>, <code>to_excel()</code>, <code>to_json()</code>.
5	▪ Limpieza y Preprocesamiento de Datos (Transformación I).	▪ Detección de valores faltantes: <code>isnull()</code>, <code>notnull()</code>. ▪ Eliminación y conteo de valores nulos: <code>dropna()</code>, <code>sum()</code>, <code>any()</code>, <code>all()</code>. ▪ Imputación de valores faltantes con <code>fillna()</code>. ▪ Detección y eliminación de duplicados. ▪ Conversión de tipos: <code>astype()</code>, <code>to_numeric()</code>, <code>to_datetime()</code>. ▪ Manejo de errores en conversión (<code>errors='coerce'</code>). ▪ Renombrado de columnas con <code>rename()</code> o <code>df.columns</code>.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
6	Programación Funcional y Transformaciones Avanzadas (Transformación II).	- Introducción a la programación funcional en Python. - Listas por comprensión. - Funciones <code>map()</code>, <code>filter()</code> y <code>reduce()</code> aplicadas a colecciones. - Uso de <code>lambda</code> y funciones anónimas. - Aplicación práctica: limpieza y transformación de datos con funciones. - Transformaciones en Pandas: <code>apply()</code>, <code>map()</code>, <code>applymap()</code>. - Creación de nuevas columnas a partir de operaciones vectorizadas. - Ejercicios prácticos de transformación.
7	Operaciones Avanzadas con Cadenas, Fechas y Tiempos.	- Operaciones con cadenas (<code>.str</code>): <code>lower()</code>, <code>upper()</code>, <code>strip()</code>, <code>replace()</code>, <code>contains()</code>, <code>split()</code>. - Expresiones regulares: <code>.str.extract()</code>, <code>.str.match()</code>, <code>.str.findall()</code>. - Manejo de fechas y tiempos en Pandas: - Acceso a componentes (<code>year</code>, <code>month</code>, <code>day</code>). - Cálculo de diferencias de tiempo (<code>Timedelta</code>). - Creación de rangos de fechas (<code>date_range()</code>). - Ejercicios de integración de texto y fechas en DataFrames.
8	Agregación, Uniones y Reestructuración de Datos (Transformación III).	- Agrupación y agregación con <code>groupby()</code>. - Funciones de agregación: <code>sum()</code>, <code>mean()</code>, <code>median()</code>, <code>min()</code>, <code>max()</code>, <code>count()</code>, <code>agg()</code>. - Agregaciones múltiples y renombrado de columnas. - Uniones de DataFrames con <code>merge()</code> y <code>join()</code>. - Reestructuración de DataFrames: <code>pivot_table()</code>, <code>melt()</code>, <code>stack()</code>, <code>unstack()</code>. - Manejo de datos jerárquicos con <code>MultiIndex</code>.
TOTAL DE HORAS		24

MÓDULO 3 - VISUALIZACIONES DE DATOS CON PYTHON

ENFOQUE: Este módulo está diseñado para enseñar el uso eficiente de Numpy y Pandas en la transformación de datos. Los participantes aprenderán a manipular, limpiar y analizar datos para obtener información útil.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
1	▪ Introducción a la Visualización de Datos con Matplotlib.	▪ Introducción a la Visualización. ▪ Componentes de Matplotlib. ▪ Creación de Gráficos Básicos. ▪ Primeros Pasos con Figuras.
2	▪ Gráficos Fundamentales y Personalización Básica.	▪ Histogramas y Gráficos de Barras. ▪ Gráficos de Pastel. ▪ Ajustes de Color y Estilo. ▪ Etiquetas y Títulos Efectivos.
3	▪ Diseños de Múltiples Gráficos y Subplots.	▪ Uso de Subplots. ▪ Organización de Múltiples Gráficos. ▪ Técnicas de Comparación Visual. ▪ Narrativa Visual Rica.
4	▪ Personalización Avanzada y Anotaciones.	▪ Anotaciones de Texto. ▪ Personalización de Marcas y Leyendas. ▪ Estilos y Temas Avanzados. ▪ Resaltar Puntos de Datos Clave.
5	▪ Tipos de Gráficos Avanzados.	▪ Diagramas de Caja (Boxplots). ▪ Gráficos de Violín. ▪ Mapas de Calor. ▪ Análisis de Distribuciones.
6	▪ Visualización de Datos Geoespaciales.	▪ Mapas Coropléticos. ▪ Gráficos de Dispersión Geoespacial. ▪ Análisis de Patrones Geográficos. ▪ Uso de Datos Geográficos en Matplotlib.
7	▪ Visualizaciones en Tres Dimensiones (3D).	▪ Gráficos de Dispersión 3D. ▪ Superficies y Mallas. ▪ Uso de mplot3d. ▪ Interactividad en 3D.
8	▪ Creación de Visualizaciones Interactivas.	▪ Funciones de Zoom y Paneo. ▪ Eventos del Usuario. ▪ Widgets Interactivos. ▪ Exploración Dinámica de Datos.
9	▪ Animaciones con Matplotlib.	▪ Creación de Animaciones Básicas. ▪ Visualización de Cambios Temporales. ▪ Técnicas de Animación Diversas. ▪ Narrativa Dinámica de Datos.

MÓDULO 4 - MACHINE LEARNING

ENFOQUE: Este módulo ofrece una visión general del Machine Learning, sus aplicaciones y su relevancia en el mundo actual. Se introducen los diferentes tipos de aprendizaje y el flujo de trabajo típico en un proyecto de Machine Learning.

TEMA	NOMBRE DEL TEMA	DESCRIPCIÓN
1	▪ Introducción al Machine Learning.	▪ Conceptos Básicos y Terminología. ▪ Tipos de Aprendizaje. ▪ Aplicaciones Prácticas. ▪ Ciclo de Vida del ML
2	▪ Preprocesamiento y Preparación de Datos I.	▪ Limpieza de Datos. ▪ Manejo de Valores Faltantes. ▪ Transformaciones de Datos. ▪ Codificación de Variables Categóricas
3	▪ Preprocesamiento y Preparación de Datos II.	▪ Normalización y Escalado. ▪ División de Conjuntos de Datos. ▪ Generación de Características. ▪ Detección de Outliers.
4	▪ Algoritmos de Aprendizaje Supervisado.	▪ Regresión Lineal y Logística. ▪ Árboles de Decisión. ▪ Máquinas de Soporte Vectorial. ▪ Modelos de Ensemble.
5	▪ Algoritmos de Aprendizaje No Supervisado.	▪ Clustering: K-means y Jerárquico. ▪ Reducción de Dimensionalidad. ▪ Análisis de Componentes Principales. ▪ Modelado de Distribuciones.
6	▪ Modelos Avanzados y Aprendizaje Profundo.	▪ Redes Neuronales Básicas. ▪ Técnicas de Regularización. ▪ Redes Neuronales Convolucionales. ▪ Modelos de Aprendizaje por Transferencia.
TOTAL DE HORAS		18

DOCENTE



DANIEL PAZ ZUÑIGA

Ingeniero Mecatrónico de la Universidad Nacional de Ingeniería, Desarrollador Python Senior y Analista de Datos con más de 6 años de experiencia liderando el desarrollo de soluciones tecnológicas, automatización de procesos y análisis de información estratégica. Especialista en el ecosistema Python (desarrollo backend, automatización y docencia), Análisis de Datos (Pandas, ETL, optimización SQL) y Visión Computacional (YOLO, OpenCV). Cuenta con amplia trayectoria en desarrollo Web Full Stack (SQL/NoSQL), procesamiento de datos e ingesta en tiempo real para sistemas IoT / Edge Computing, así como en la creación de reportes institucionales automatizados e integración de APIs.



GIANCARLO CESPEDES MAS

Docente y asesor con más de 5 años de experiencia en programación, informática, ciencias y robótica, trabajando en instituciones públicas y privadas. Desde el 2019 es docente en la Oficina de Tecnologías de la Información de la Universidad Nacional de Ingeniería (OTI-UNI), donde ha dictado cursos de especialización en programación e informática orientados a distintos públicos académicos y profesionales.

Su experiencia abarca la formación académica en ciencias e ingeniería, el diseño e implementación de cursos de formación profesional y la realización de asesorías empresariales en automatización de procesos y análisis de datos.

() La Universidad se reserva el derecho de cambiar algún docente por contingencias inesperadas.*



INFORMACIÓN GENERAL



Horario

Lunes, Miércoles y Viernes de
19:00 a 22:00hrs.



Modalidad

Virtual
Clases en vivo



Duración

90 horas cronológicas
120 horas pedagógicas
30 sesiones



INVERSIÓN

PRECIO
REGULAR

S/1,800



PREVENTA

A SOLO

S/499,00*

Promoción Valida hasta el 31 de agosto de 2026 o hasta agotar cupos*



PROCESO DE INSCRIPCIÓN

Los siguientes documentos deberán ser enviado al correo electrónico:

diplomas.oti@uni.edu.pe

Asunto del correo: Inscripción - [Nombre del Programa]

Mensaje del correo: [Nombre y Apellido] [DNI]

1. Completar la Ficha de Inscripción virtual y tomar captura al finalizar el llenado.
2. Aceptar el Reglamento de Términos y Condiciones de Cursos/Programas
3. Copia simple del DNI (Legible)
4. Voucher de pago

Nota: Una vez enviado los documentos solicitados vía correo electrónico, el participante deberá esperar la confirmación de su matrícula.



MODALIDADES DE PAGO



PASO 1: Solicita a un asesor de ventas de la Unidad de Capacitación activar el ID personal. Indicando los siguientes datos: nombre y apellidos, número de documento de identidad (DNI o pasaporte), correo electrónico, número de celular y monto a pagar.

(*) En el caso de requerir factura, se solicitará los siguientes adicionales: R.U.C, Razón Social, Domicilio Fiscal y correo electrónico donde se enviará dicha factura.

PASO 2: Procede a realizar el pago a través de los siguientes canales de pagos autorizados.



Agente y Ventanilla

Indicar el **código 15226**
Universidad Nacional de Ingeniería
+ DNI, Pasaporte o RUC del alumno,
Concepto:
PAGO DE ESTUDIANTES



Banca móvil - BCP

Selecciona la opción: "PAGAR SERVICIO"
Escribe en el buscador por Empresa o Servicio:
"Universidad Nacional de Ingeniería"
Elige la opción de Universidad Nacional de
Ingeniería "PAGO ESTUDIANTES"
Coloca tus datos personales: DNI / pasaporte /
RUC
y ¡Listo, pago realizado!



Pago en Niubiz

Recibirá **automáticamente un**
correo electrónico con el enlace
para realizar el pago en línea.



COMUNÍCATE CON

diplomas.oti@uni.edu.pe

Horario de atención Lun. a Vie. de 09:00 a 17:00hrs.

Oficina de Tecnologías de la Información



www.ctic.uni.edu.pe