

**Programas de Estudio del  
Centro de Formación en Analítica Aplicada y  
Reconversión Profesional**

CFAARP

Ciudad de México, México

Carlos Ernesto Martínez Rodríguez

7 de mayo de 2026

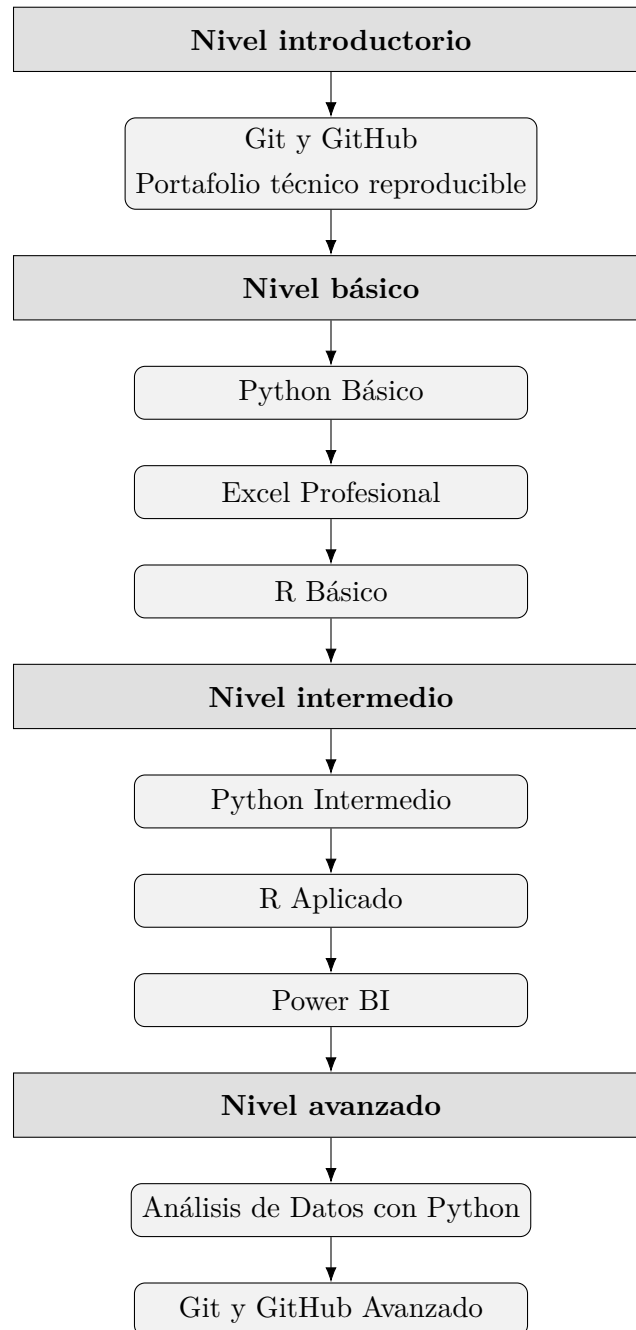
## Presentación

El presente documento reúne los programas de estudio correspondientes a la oferta académica inicial del **Centro de Formación en Analítica Aplicada y Reconversión Profesional**, diseñada para facilitar procesos de transición laboral hacia áreas de programación y análisis de datos mediante un modelo educativo práctico, modular y progresivo.

Los cursos aquí descritos se estructuran con base en competencias técnicas aplicadas, desarrollo de portafolio profesional reproducible y uso de herramientas alineadas con el entorno laboral actual.

Este conjunto de programas constituye la base curricular del modelo académico institucional en su fase piloto y se organiza como una trayectoria progresiva de formación en analítica aplicada estructurada en niveles introductorio, básico e intermedio, orientados al desarrollo de competencias técnicas transferibles a contextos académicos y profesionales contemporáneos.

## Mapa curricular del programa



El mapa curricular presenta la secuencia formativa sugerida del programa, organizada por niveles progresivos. El curso introductorio de Git y GitHub establece la base metodológica para la construcción de portafolio técnico reproducible, mientras que los niveles básico, intermedio y avanzado desarrollan competencias acumulativas en programación, análisis estadístico, visualización de datos y documentación profesional.

## 2. Nivel básico

### 2.1. Python Básico

#### Descripción general

Este curso introduce al estudiante en los fundamentos de la programación utilizando Python como lenguaje principal de trabajo, empleando Visual Studio Code como entorno de desarrollo estándar para la construcción de scripts reproducibles orientados al análisis de datos, automatización de tareas y desarrollo de portafolio técnico profesional.

El curso tiene un enfoque completamente práctico y constituye el primer módulo formal de programación dentro del programa de formación en analítica aplicada.

#### Datos generales del curso

- Duración: 24 horas
- Modalidad: Taller práctico con proyecto guiado
- Nivel: Básico
- Requisito: Curso propedéutico Git y GitHub (recomendado)
- Entorno de trabajo: Visual Studio Code
- Recomendado para: Estudiantes sin experiencia previa en programación

#### Objetivo general

Desarrollar en el estudiante las competencias fundamentales de programación estructurada en Python para la construcción de scripts reproducibles orientados a la manipulación de datos, automatización básica y resolución de problemas computacionales.

#### Objetivos específicos

Al finalizar el curso, el estudiante será capaz de:

- Comprender la estructura básica de un programa en Python
- Utilizar variables y tipos de datos fundamentales
- Implementar estructuras de control condicionales
- Construir ciclos iterativos con `for` y `while`
- Definir funciones reutilizables
- Manipular listas, diccionarios y estructuras básicas de datos
- Leer y escribir archivos de texto

- Resolver problemas computacionales básicos mediante programación
- Documentar scripts con buenas prácticas iniciales
- Publicar código en repositorios reproducibles en GitHub

### Palabras clave

Python, programación estructurada, variables, tipos de datos, condicionales, ciclos, funciones, listas, diccionarios, archivos de texto, scripting reproducible, automatización básica, Visual Studio Code, portafolio técnico.

### Contenido temático

#### Módulo 1: Entorno de trabajo con Python

- 1.1 Instalación de Python
- 1.2 Instalación y configuración de Visual Studio Code
- 1.3 Extensión oficial de Python para VS Code
- 1.4 Uso de la terminal integrada
- 1.5 Ejecución de scripts desde VS Code
- 1.6 Introducción a notebooks en VS Code
- 1.7 Extensiones recomendadas: Python, Jupyter, GitLens

#### Módulo 2: Variables y tipos de datos

- 2.1 Variables y asignación
- 2.2 Tipos numéricos
- 2.3 Cadenas de texto
- 2.4 Conversión de tipos
- 2.5 Operaciones básicas

#### Módulo 3: Estructuras de control

- 3.1 Operadores relacionales
- 3.2 Operadores lógicos
- 3.3 Estructuras condicionales
- 3.4 Uso de `if`, `elif`, `else`

**Módulo 4: Ciclos iterativos**

- 4.1 Ciclo `for`
- 4.2 Ciclo `while`
- 4.3 Control de iteraciones
- 4.4 Uso de `break` y `continue`

**Módulo 5: Estructuras de datos básicas**

- 5.1 Listas
- 5.2 Tuplas
- 5.3 Diccionarios
- 5.4 Conjuntos
- 5.5 Iteración sobre estructuras

**Módulo 6: Funciones**

- 6.1 Definición de funciones
- 6.2 Parámetros y argumentos
- 6.3 Valores de retorno
- 6.4 Organización modular del código

**Módulo 7: Manejo de archivos**

- 7.1 Lectura de archivos de texto
- 7.2 Escritura de archivos
- 7.3 Procesamiento básico de datos desde archivos

**Módulo 8: Proyecto integrador**

- 8.1 Diseño de script reproducible
- 8.2 Automatización de tarea sencilla
- 8.3 Documentación del código
- 8.4 Publicación en GitHub

**Proyecto integrador del curso**

El proyecto integrador consistirá en el desarrollo de un script en Python capaz de leer, procesar y analizar información contenida en un archivo de datos estructurado, generando

resultados interpretables y reproducibles.

El estudiante deberá implementar un programa que incluya:

- Lectura de datos desde un archivo de texto o archivo CSV
- Uso de variables, estructuras condicionales y ciclos iterativos
- Definición de al menos dos funciones reutilizables
- Procesamiento de información mediante estructuras de datos (listas o diccionarios)
- Generación de resultados en pantalla o en archivo de salida
- Documentación básica del código mediante comentarios

El proyecto deberá organizarse en un repositorio público en GitHub que incluya:

- Archivo principal del script
- Archivo de datos utilizado
- Archivo README.md con descripción del problema, metodología empleada y resultados obtenidos

Este proyecto constituirá la primera evidencia formal de programación reproducible dentro del portafolio técnico del estudiante. El estudiante deberá incluir instrucciones de ejecución del script dentro del archivo README.md para garantizar la reproducibilidad del proyecto.

### **Plantilla institucional de README.md para el proyecto integrador**

Como parte del proyecto integrador del curso, el estudiante deberá documentar su repositorio utilizando el siguiente esquema base de archivo README.md, con el propósito de garantizar claridad técnica y reproducibilidad del trabajo desarrollado.

#### **Estructura sugerida del archivo README.md**

- Título del proyecto
- Descripción general del problema abordado
- Objetivo del programa desarrollado
- Descripción del archivo de entrada utilizado (por ejemplo, archivo CSV)
- Funcionalidades implementadas en el script
- Instrucciones de ejecución del programa desde terminal
- Ejemplo de salida esperada
- Estructura del repositorio
- Tecnologías utilizadas
- Autor del proyecto

**Ejemplo de contenido mínimo esperado** El archivo deberá incluir una descripción similar a la siguiente:

- Lectura de archivo de datos estructurado
- Procesamiento mediante listas o diccionarios
- Cálculo de estadísticas básicas (promedio, máximo y mínimo)
- Conteo de registros que cumplen condiciones específicas
- Presentación de resultados en pantalla

Asimismo, el estudiante deberá incluir instrucciones claras para ejecutar el script, por ejemplo:

```
python analisis.py
```

La correcta documentación del proyecto mediante el archivo `README.md` constituye un elemento obligatorio para la acreditación del curso y forma parte de la construcción del portafolio técnico reproducible del estudiante.

### Producto final del curso

Al finalizar el curso, el estudiante contará con:

- Un script funcional en Python
- Lectura y procesamiento de datos desde archivo
- Uso de funciones definidas por el usuario
- Documentación básica del proyecto
- Repositorio publicado en GitHub como primera evidencia formal de programación reproducible dentro del portafolio técnico del estudiante

### Criterios de evaluación

La evaluación del curso se realizará mediante la revisión del proyecto final publicado por el estudiante conforme a la siguiente ponderación:

- Uso correcto de variables y estructuras de control (20 %)
- Implementación adecuada de funciones (20 %)
- Manipulación básica de archivos (20 %)
- Organización y claridad del código (20 %)
- Documentación del proyecto en GitHub (20 %)

La acreditación del curso requiere cumplir satisfactoriamente al menos el 80% de los criterios establecidos.

### **Importancia dentro del programa formativo**

Este curso establece las bases de programación necesarias para los cursos posteriores del programa, especialmente:

- Python Intermedio
- Análisis de Datos con Python
- Machine Learning aplicado