



Curso

# Especialización en Python

Desde la documentación oficial

# Descripción

Este curso integral de Python abarca desde los fundamentos del lenguaje hasta estructuras avanzadas. Incluye módulos especializados en PyQt5, MySQL, NumPy, Matplotlib, y OpenCV. Además, proporciona una introducción a Anaconda, machine learning y deep learning, equipándote con las herramientas necesarias para el desarrollo de aplicaciones, análisis de datos y aprendizaje automático.

El curso contiene 2 bloques.

**BLOQUE I** Estudio del Lenguaje de Programación según la Documentación Oficial.

**BLOQUE II** Aplicaciones de Python para base de datos, diseño GUI, PDI e IA.



## Requisitos

De preferencia tener nociones en algún programación, igualmente el curso iniciará desde cero.



## Dirigido a

Estudiantes y profesionales con interés en aprender y aplicar el Lenguaje de Programación Python.



## Duración

El curso tiene una duración de 52 Horas.

```
os.path.join(datadir, "feature.bin")
binary: %s" % self.featurefile
txt: %s" % idfile

def read(self, requested, isname=True):
    if isname:
        index_name_array = [(self.name2index[x], x) for x in requested if x in
            str.split(open(idfile).read()) if x.strip()
            in self.names, range(len(self.names)))]
```

## BLOQUE I

### Clase 1 Introducción

- Qué es un lenguaje de programación?.
- Tipos de lenguajes.
- Niveles en los lenguajes.
- Introducción a Python.
- Instalando Python.
- El proceso de la implementación.
- La implementación CPython.



### Clase 2 Objetos

- El contexto del Lenguaje C con Python.
- Qué es un objeto en Python?.
- Heap vs Stack.
- Variables y objetos.
- Tiempo de vida de un objeto.
- Garbage Collector.
- Objetos numéricos.



### Clase 3 Estructuras I

- Modelo de datos de Python.
- Objetos secuencia: Listas, Tuplas y String.
- Cabecera de listas vs tuplas.
- Index y Slicing.
- Pruebas de mutabilidad e inmutabilidad.
- Interpretación del Heap/Stack Memory en RAM.

## Clase 4

### Métodos y String

- Métodos en los objetos.
- Métodos en Listas.
- Métodos en tuplas.
- Métodos de los String.

## Clase 5

### Estructuras II

- Objetos Mappings: Diccionarios.
- Index en los diccionarios.
- Heap/Stack memory.
- Objetos invocables: Funciones, métodos y clases.

## Clase 6

### Jerarquías entre objetos

- Clases estándar en Python.
- Las clases raíz principales.
- Metaclasses.
- Type sobre las clases estándar.
- Type sobre type
- Diagrama de herencia

## Clase 7

### Operadores

- Funciones de entrada y salida.
- Operadores aritméticos.
- Ejercicios diversos.
- Operadores relacionales.
- Ejercicios diversos.
- Operadores lógicos.
- Ejercicios diversos.

## Clase 8

### Estructuras de control

- Sentencias condicionales: if, elif y else.
- Ejercicios diversos.
- Sentencias while.
- Sentencia for.
- For anidados.
- Ejercicios prácticos con bucles.

## Clase 9

### Estructuras por compresión

- Objetos tipo secuencia.
- Objetos tipo conjunto.
- Objeto tipo mappings.
- Ejercicios sobre listas por compresión.

## Clase 10

### Funciones I

- Definición de función.
- Análisis desde el stack y heap en RAM.
- Parámetros y argumentos.
- Por orden y nombre.
- Limitados e ilimitados: \*args , \*\*kwargs.

## Clase 11

### Funciones II

- Decoradores.
- Creación de funciones decoradoras.
- Recursividad: Caso base y recursivo.
- Desbordamiento el stack.
- Ejercicios prácticos.
- Introducción a los fractales.

## Clase 12

### Programación funcional (offline)

- Análisis de funciones Lambda.
- Función Map.
- Función Filter.
- Lambda/Map/Filter vs Listas por compresión.



## Clase 13

### Objetos tipo module

- Qué son los módulos?.
- Creación de módulos.
- CPython, análisis de implementación.
- Métodos de importación y sus diferencias.
- Python compilado pyc.
- Acerca de `__pycache__`.

## Clase 14

### Paquetes, Errores y excepciones (offline)

- Los paquetes en Python.
- Aserciones.
- Qué son las aserciones?.
- La palabra clase Assert.
- Errores y Excepciones.
- Errores de sintaxis.
- Excepciones.

## Clase 15

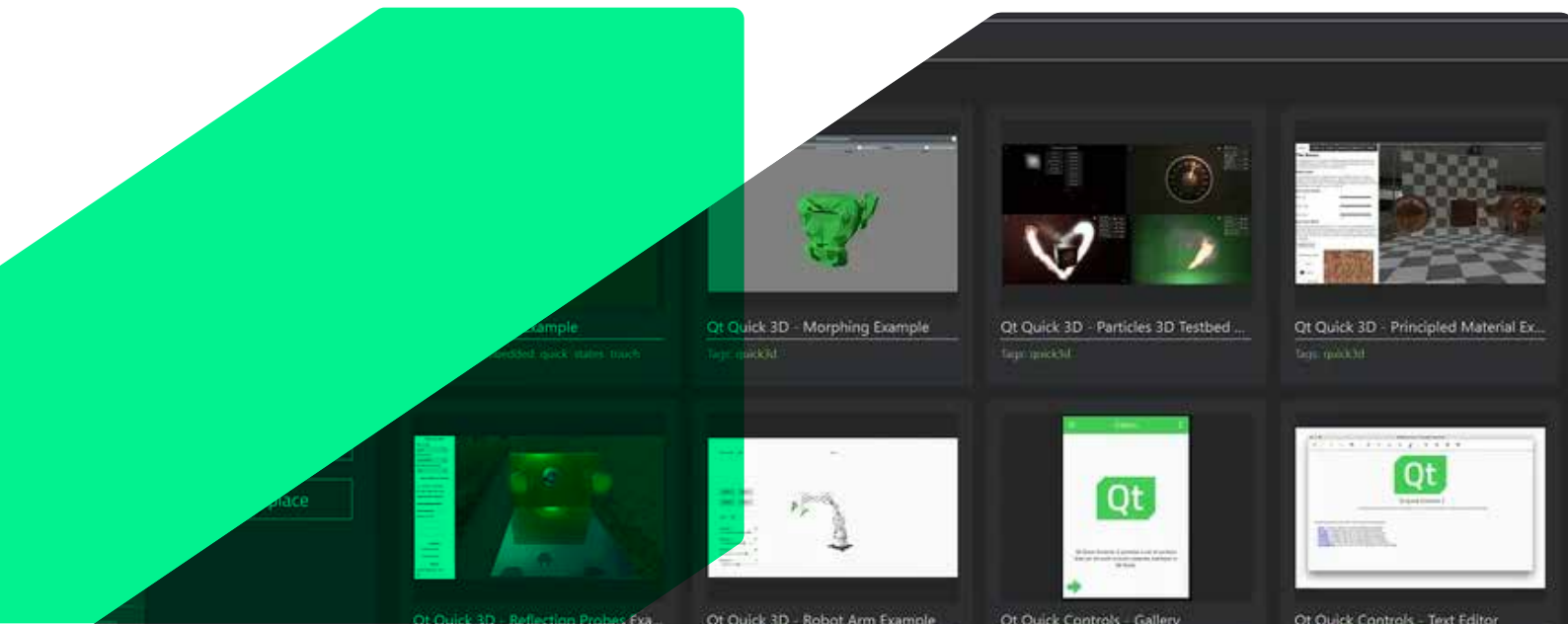
### POO I

- Qué es POO?.
- Clases, objetos, métodos y atributos.
- Definición de paradigma.
- Como abstraer un caso real a POO.
- Construyendo una clase desde cero.
- Las clases estándar de Python.

## Clase 16

### POO II

- Atributos de clase vs objetos.
- Métodos de clase.
- Métodos de clase vs de instancia.
- Métodos estáticos de clase.





## BLOQUE II

### Clase 17

#### Interfaz Gráficas

- Herramientas para diseño GUI en Python.
- Descripción de PyQt5.
- La clase QApplication, QWidget, QPushButton, QLabel.
- Diseño de botones, displays, sliders.
- Introducción a QT Designer.

### Clase 18

#### Base de datos

- Introducción a las bases de datos.
- Descripción de SQL.
- Instalación de MYSQL.
- Lectura, escritura y creación de tablas.
- Integración PyQt5 con MYSQL.
- Tablas en PyQt5 y consultas a la Base de datos.

### Clase 19

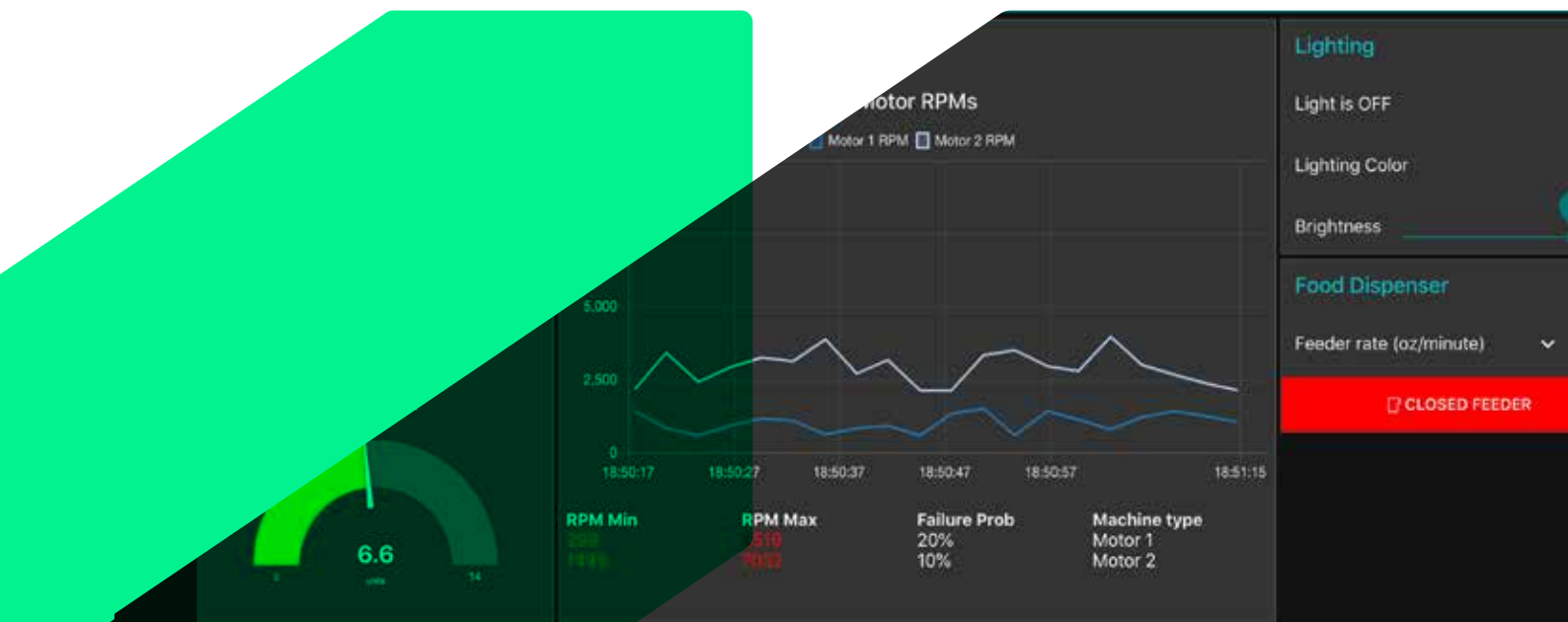
#### Numpy

- Instalación de numpy.
- Introducción al módulo de Numpy.
- Operaciones con objetos tipos Numpy.array.
- Slicing, matrices especiales.
- Métodos y atributos.

### Clase 20

#### Matplotlib (offline)

- Dibujando gráficos 2D.
- Gráficos lineales.
- Ploteo simple y múltiple.
- Personalización de las ventanas.
- Histogramas.



## Clase 21

### OpenCV

- Introducción al procesamiento digital de imágenes.
- Lectura de imágenes.
- Guardado de imágenes.
- Vídeos, uso de cámara.
- Operaciones bitwise y máscaras.
- Modelos y espacios de color.
- Detección de objetos por color.

## Clase 22

### OpenCV y PyQt5

- Integración de PyQt con OpenCV.
- Integración de cámara con la GUI de PyQt5.
- Botones para captura de imágenes.
- Apertura de imágenes en la GUI.
- Integración del programa de detección de objetos con la GUI de PyQt5.

## Clase 23

### Anaconda y Pandas

- Instalación de anaconda.
- Uso de enviroments.
- Instalación de módulos: Numpy y matplotlib.
- Pandas y el dataframe.
- Lectura de archivos.
- Manipulación de datos con Pandas.

## Clase 24

### Machine Learning

- Qué es el machine learning?.
- Clasificación de los algoritmos.
- Aprendizaje supervisado y no supervisado.
- Regresión lineal.
- Construcción del modelo.
- Función de costey de optimización.
- Caso práctico: Predicción de costo por incidentes en una empresa.
- Próximos pasos.





## Clase 25

### Deep Learning I

- Introducción al deep Learning.
- End-to-End Learning.
- Aplicaciones y clasificación de las técnicas de Deep Learning.
- Introducción a las Redes neuronales Artificiales.
- Introducción a la neurona de Mcculloch y Pits.
- Caso práctico: Implementación de la neurona M-P.

## Clase 26

### Deep Learning II

- En esta sección se realizará un ejemplo orientado al diagnóstico de cáncer de mama con la neurona M-P.
- Próximos pasos.



```
bin(datadir, "feature.bin")
features, %d dimensions" % (len(self.names), self.ndim)
ary: %s" % self.featurefile
txt: %s" % idfile
self, requested, isname=True):
index_name_array = [(self.name2index[x], x) for x in requested if x
else:
assert(min(requested)>=0)
assert(max(requested)<len(self.names))
array = [(x, self.names[x]) for x in requested]
for x in index_
```

# Beneficios



## Certificado

A nombre de la institución Umaker SAC, validando los conocimientos adquiridos.



## Aula virtual

Acceso al portal exclusivo de alumnos desde donde podrá ver de nuevo las clases y recursos adicionales.



## Asesorías

Asesorías personalizadas gratuitas de reforzamiento de los temas de curso