

## Guía 2 Python

Objetivo General: Realizar programas de mediana/alta complejidad usando Python como lenguaje de programación.

Objetivos Específicos:

1. Completar ejercicios utilizando la programación orientada a objetos.
2. Construir un web crawler usando python.

## Parte I : POO (II)

### Teoría.

En la mayoría de los ejemplos y ejercicios anteriores se diseñó los programas sobre bloques de sentencias que manipulan datos, conocidos como funciones. A este estilo se le llama Programación orientada a procedimientos. Existe otra forma de organizar los programas, que consiste en combinar los datos y su funcionalidad y juntarlo en elementos llamados objetos. A esta forma se le llama Programación orientada a objetos. Este paradigma facilita la escritura de programas mas grandes.

Los Objetos pueden almacenar datos usando variables ordinarias que pertenecen al objeto. Estas variables se conocen como atributos. Los objetos pueden tener funcionalidades que pertenecen a la clase, a estos funcionalidades se le conocen como métodos de clase.

Los atributos pueden ser de dos tipos: de instancia y de clase.

Una clase se crea usando la palabra reservada «class». Los atributos y metodos de la clase se listan en un bloque indentado.

## El «self»

Los metodos de una clase, solo pueden ser diferenciados de las funciones por que estos llevan un argumento extra al inicio de la funcion llamado por estándar como «self», a este parámetro no se le da un valor cuando se llama al método. Cuando se hace la llamada al método de una instancia Python provee del valor. Esta variable particular se refiera al objeto mismo y por convención se le llama «self».

*«self» es el equivalente al puntero «this» en java y C/C++*

## El método «\_\_init\_\_»

Existen muchos nombres de metodos que tienen un significado especial en las clases de python. Tal es el caso del método \_\_init\_\_.

Este método corre luego de que el objeto de una clase es instanciado. Este método es muy útil para inicializar objetos de la forma en que se desee.

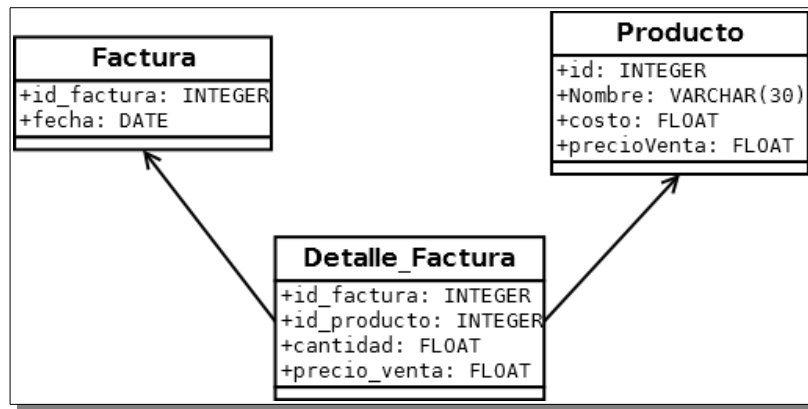
El nombre de este método siempre debe ser \_\_init\_\_.

## Ejemplo

1. Ejecutar y Analizar el archivo MainVista.py en el directorio de semana\_2/ejemplos/vista.

## Ejercicios

1. Investigar el uso de sqlite en python.
2. Completar el código en el archivo ejercicios/modelos/modelo.py .
3. Crear las clases correspondientes para el siguiente diagrama en sus respectivos archivos dentro de la carpeta ejercicios/modelos/:



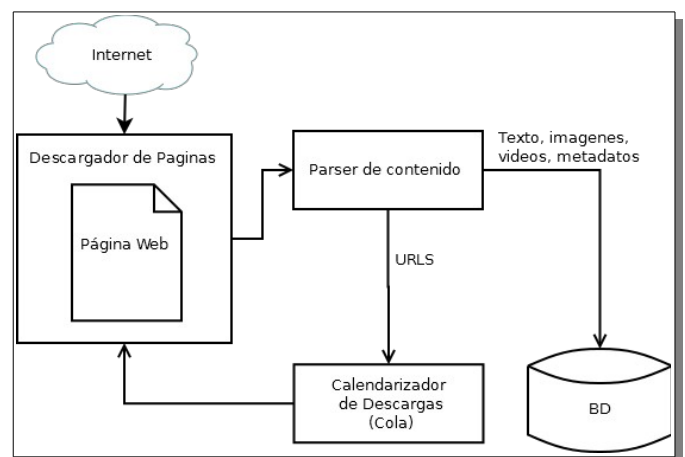
## Parte II: Construyendo un web crawler.

Un Web Crawler es un bot que sistemáticamente explora por el internet, con el fin de indexar. Muchos motores de búsqueda y otros tipos de páginas web utilizan el web crawling para actualizar su contenido. Estos copian el contenido de todas las paginas que visitan para su procesamiento.

### Arquitectura de un web crawler.

Un web crawler suele estar compuesto por los siguientes elementos:

1. Módulo de descarga de contenido.
2. Módulo que se encargue de parsear el contenido y que divida el contenido según sea el objetivo, las actividades mas comunes son:
  1. Urls
  2. Texto
  3. Imágenes
3. Módulo de almacenamiento de contenido.
4. Módulo de calendarización de descargas.



4. Videos
5. etc

## Ejercicio

1. En la carpeta ejercicios/webcrawler se ha dejado el archivo instrucciones.txt.