# CRONOGRAMA - BOOTCAMP PYTHON

## 18/10/2021

## 19/10/2021

## 20/10/2021

## 21/10/2021

## 25/10/2021

## 26/10/2021

## 27/10/2021

## 03/11/2021

## 04/11/2021

## 10/11/2021

## 11/11/2021

## 15/11/2021

## 16/11/2021

## 17/11/2021

## 18/11/2021

## 22/11/2021

## 23/11/2021

## 24/11/2021

## 25/11/2021

## 29/11/2021

## 30/11/2021

## 01/12/2021

## 02/12/2021

## 13/12/2021

## 14/12/2021

## 15/12/2021

## 16/12/2021

## 20/12/2021

## 21/12/2021

## 22/12/2021

## 10/01/2021

Aplicación de escritorio con GUI desarrollado con Tkinter.

Introducción a la orientación a objetos mediante el diseño del juego Space Invaders: atributos, constructores, instanciación, invocación a métodos.

Ejercicio modelado y programación de una transferencia bancaria.

## 11/01/2022

Continuación ejercicio transferencia bancaria.

Presentación de alternativas a la resolución de un problema (generación de PDF) mediante método de la clase o delegación en clase externa.

Orientación a objetos. Ejemplo de la cadena hotelera y la relación de asociación entre clases.

## 12/01/2022

Orientación a objetos. Ejemplo de la cadena hotelera y la relación de asociación entre clases.

Orientación a objetos. Atributos y métodos de clases y de instancia. Decorador @classmethod.

Ejercicio relacionado con los métodos de clase y de instancia.

## 13/01/2022

Aplicación de la orientación a objetos a un esquema de videojuego realizado con Pygame.

1:45 (aprox.). Herencia.

## 17/01/2022

Herencia.

Herencia múltiple.

## 18/01/2022

Excepciones.

## 19/01/2022

Encapsulación.

Visibilidad: público vs \_\_privado.

Solución Ejercicio 043\_ValidacionOO.py.

## 20/01/2022

Decoradores.

Visibilidad: \_protected.

Getter y setter realizados con decoradores aplicados a atributos privados y protegidos.

## 24/01/2022

Métodos *isinstance* e *issubclass*.

Clases y métodos abstractos.

Patrón Factory.

Ejercicio de generador de mensajes multiidioma con clases abstractas.

## 25/01/2022

Repaso del ejercicio de multiidioma al que se le incorpora un control de clase (isinstance) y se corrige un error ya que el Factory proporcionaba siempre la misma instancia.

Clases internas (*inner class*). La clase “de fuera” se denomina *outer class*.

Patrón MVC (*Model – View – Controller*).