Logo

Description automatically generatedLogo

Description automatically generated with medium confidence

*Practica: 04*

*Nombre de la Practica: Ejercicios de Pypass*

Instituto Politécnico Nacional.

Escuela Superior de Cómputo.

Licenciatura en ciencia de datos.

Nombre de la materia: Desarrollo de Aplicaciones para el Análisis de Datos

Grupo: 4AV1

Profesora: Sandra Luz Morales Guitron

Lopez Mendez Emiliano

INDICE

[INTRODUCCION 3](#_Toc180661222)

[DESAROLLO 3](#_Toc180661223)

[FUNCIONES 3](#_Toc180661224)

[OOP 10](#_Toc180661225)

[EXCEPTIONS 25](#_Toc180661226)

[CONCLUCIONES 27](#_Toc180661227)

# INTRODUCCION

En esta practica se llevo la instalación de un Editor de programación para poder programar con la versión mas actualizada de Python. Escogiendo la computadora que vamos a usar, el sistema operativa y el Editor que se usaran.

# DESAROLLO

Se llevo acabo el análisis y configuración de varios códigos ya adentro de pypass. Se modificaron para poder imprimir los resultados deseados y también para poder imprimir dichos programas se hizo un archivo aparte llamado args.py con las variables que se piden del main.

## FUNCIONES

1. num-in-interval **Este código define una función llamada `in\_range` que verifica si un número `n` se encuentra dentro del intervalo cerrado definido por `start` y `end`. Devuelve `True` si el número está dentro del rango, y `False` en caso contrario. En la ejecución, verifica si el número 5 está dentro del intervalo [1, 10], imprimiendo el resultado de la evaluación.**

A screen shot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

1. extract-evens

A computer screen with colorful text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. split-case

A screen shot of a computer screen

Description automatically generatedA screen shot of a computer

Description automatically generated

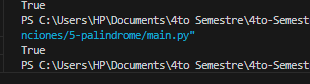
1. Perfect

A screen shot of a computer program

Description automatically generated

1. Palindrome

A computer screen shot of a program code

Description automatically generated

1. count-vowels-recursive **Este código define una función recursiva llamada `count\_vowels` que cuenta el número de vocales (incluyendo tildadas) en una cadena de texto dada. La función revisa cada carácter de la cadena: si es una vocal, suma 1 y continúa con el resto del texto; si no, simplemente avanza. Cuando la cadena está vacía, la recursión termina y devuelve el total de vocales encontradas.**

A computer screen shot of a program

Description automatically generatedA screenshot of a computer program

Description automatically generated

1. Pangram **Este código verifica si un texto es un pangrama, es decir, si contiene todas las letras del alfabeto al menos una vez. Primero convierte el texto a minúsculas y elimina los caracteres no alfabéticos. Luego compara las letras presentes en el texto con un conjunto del alfabeto. Si todas las letras del alfabeto están en el texto, devuelve `True`; de lo contrario, `False`.**

A computer screen shot of text

Description automatically generatedA screen shot of a computer screen

Description automatically generated

1. cycle-alphabet **Este código genera caracteres del alfabeto de forma cíclica. Utiliza un generador para iterar sobre el alfabeto en minúsculas repetidamente. Para cada carácter, utiliza el operador módulo (`%`) para asegurarse de que la posición se reinicie al alcanzar el final del alfabeto. En los ejemplos de ejecución, genera secuencias de 10 y 43 caracteres en ciclo.**

A computer screen shot of a program code

Description automatically generatedA black screen with white text

Description automatically generated

1. bubble-sort **Este código implementa el \*\*algoritmo de ordenamiento por burbuja\*\*. Crea una copia de la lista original y recorre repetidamente la lista, comparando elementos adyacentes e intercambiándolos si están en el orden incorrecto. De esta forma, los elementos más grandes "burbujearán" hacia el final en cada pasada. Al finalizar, retorna la lista ordenada. En la ejecución, muestra una lista desordenada y luego la lista ordenada con este método.**

A computer screen shot of a program code

Description automatically generatedA screenshot of a computer screen

Description automatically generated

1. consecutive-seq **Este código implementa una función recursiva que busca un número que aparezca consecutivamente una cantidad específica de veces en una lista. Si encuentra el número que cumple con las repeticiones requeridas, lo devuelve; de lo contrario, continúa recorriendo la lista. Si no se encuentra ninguna coincidencia al llegar al final, devuelve `None`. En la ejecución, busca un número que se repita tres veces consecutivas en la lista dada.**

A computer screen shot of a code

Description automatically generatedA screen shot of a computer

Description automatically generated

1. magic-square

A screenshot of a computer program

Description automatically generatedA screen shot of a computer

Description automatically generated

1. nested-add

A screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

1. fibonacci-recursive

A screen shot of a computer program

Description automatically generatedA screen shot of a computer

Description automatically generated

1. Hyperfactorial

A screen shot of a computer program

Description automatically generatedA screen shot of a computer screen

Description automatically generated

1. fibonacci-generator

A screen shot of a computer program

Description automatically generatedA screenshot of a computer

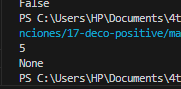
Description automatically generated

1. palindrome-recursiveA computer screen shot of text

   Description automatically generatedA screen shot of a computer

   Description automatically generated
2. deco-positive

A screen shot of a computer program

Description automatically generated

1. slice-recursive

A computer screen shot of a code

Description automatically generatedA screen shot of a computer

Description automatically generated

## OOP

1. Dna **Este código define una clase `DNA` que representa secuencias de ADN, permitiendo varias operaciones: suma y multiplicación de secuencias, cálculo de estadísticas de bases (A, C, G, T), modificación de bases en posiciones específicas y gestión de archivos. Además, proporciona funcionalidades para combinar secuencias, identificar coincidencias y guardar/cargar secuencias desde archivos.**

A screen shot of a computer program

Description automatically generated

A computer screen shot of a code

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

1. Istack **Este código implementa una clase `IntegerStack` que representa una pila de enteros con operaciones básicas como \*push\*, \*pop\* y \*peek\*. Además, se hace iterable mediante un iterador personalizado `IntegerStackIterator`, que permite recorrer los elementos de la pila desde el tope hacia la base usando un bucle `for`.**

A screen shot of a computer code

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen with white and green text

Description automatically generatedA screen shot of a computer

Description automatically generated

1. Iqueue **Este código define una clase `IntegerQueue` que implementa una cola de enteros con un tamaño máximo ajustable. Permite agregar (enqueue) y remover (dequeue) elementos, expandir el tamaño de la cola, guardar y cargar su contenido desde un archivo, y sumar colas. Además, proporciona un iterador personalizado para recorrer sus elementos.**

A screen shot of a computer program

Description automatically generated

A computer screen with many colorful text

Description automatically generated with medium confidence

A computer screen shot of a code

Description automatically generated

A screen shot of a computer program

Description automatically generatedA computer screen shot of a computer code

Description automatically generated

1. Date **Este código define una clase `Date` que representa una fecha personalizada con varias funcionalidades adicionales. Permite operaciones como comparar fechas, sumar o restar días, calcular el día de la semana utilizando el algoritmo de Zeller, verificar si es fin de semana y obtener la fecha en diferentes formatos. Además, permite convertir fechas a un formato basado en días acumulados y viceversa. También se pueden realizar operaciones aritméticas entre dos fechas y manipularlas fácilmente.**

A screen shot of a computer

Description automatically generatedA computer screen shot of a program code

Description automatically generatedA screenshot of a computer program

Description automatically generatedA computer screen shot of a program

Description automatically generatedA computer screen shot of a program code

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

## EXCEPTIONS

1. Poker-Card **Este código define una clase `Card` que representa cartas de poker, permitiendo validarlas, compararlas y sumarlas. También incluye una excepción personalizada `InvalidCardError` para manejar entradas inválidas. Provee métodos para crear cartas, obtener los palos disponibles, y generar todas las cartas de un palo específico. Finalmente, permite sumar dos cartas, ajustando su valor si supera 13.**

A computer screen with colorful text

Description automatically generatedA screen shot of a computer program

Description automatically generatedA computer screen shot of a program code

Description automatically generatedA screenshot of a computer program

Description automatically generated

# CONCLUCIONES