

22/03/2023

**Actividad. -Animal Shelter**

**Estructura de Datos**

**Chávez Rodríguez Mauricio Yosef**



## Programa:

Un refugio de animales que tiene perros (nombre, edad, raza) opera un FIFO. Las personas deben adoptar al mayor (según la fecha de llegada). Cree la estructura de datos para mantener el sistema e implementar operaciones como enqueue\_dogs o dequeue\_dogs.

## Explicación de Código:

```
import datetime
class Dog:
    def __init__(self, name, age, breed, date):
        self.name = name
        self.age = age
        self.breed = breed
        self.date = date
        self.next = None
```

Comenzamos importando una librería que trabaja con fechas la cual no ayudara para conocer la fecha y hora de cuando un animal es ingresado al refugio, para posteriormente definir nuestra clase perro la cual contará con algunas de las principales características del animal, nombre, edad, raza, la fecha en la cual llegó y un next el cual apuntará a el siguiente perro, de esta forma implementaríamos la estructura FIFO.

```
class Queue:
    def __init__(self):
        self.first = None
        self.length = 0
        self.last = None
```

Seguido de lo antes mencionado se encuentra la clase queue la cual tendrá los atributos propios de la estructura y los métodos encargados de los enlaces de un nodo y otro, entre sus atributos encontramos el length para saber cuántos nodos están enlazados o en este caso perritos, también encontramos el first atributo que apunta al primer dato de la fila y tenemos el last el cual apunta al último elemento de la agrupación de datos.

```
def dog_enqueue(self, name, age, breed):
    date = datetime.datetime.now()
    new_node = Dog(name, age, breed, date)
    if self.first == None:
        self.first = new_node
        self.last = new_node
    else:
        self.last.next = new_node
        self.last = new_node

    self.length += 1
    return
```

Dentro de nuestra clase Queue encontramos el primer método, el cual es el encargado de generar el nodo que almacenará los datos del perrito para ello se utiliza la clase Dog mencionada anteriormente a la cual solo se le mandan los parámetros o datos del perrito y dicha clase crea el nodo, es necesario decir que al entrar a dicho método de se registra la fecha y hora de entrada de los datos para llevar un mejor control, tras crear el nodo el método lo pondrá al final de la fila, pues la función almacena los datos según el orden de llegada de cada animal, para ello se utilizan 2 condiciones, una para cuando existe un nodo inicial y uno para cuando no existe tal nodo, y al agregar los datos de un animal nuevo se incrementa el tamaño de nuestra fila. es decir, nuestro length crece en 1.

```
def print_queue(self):
    tem=self.first
    while tem is not None:
        print("-----")
        print(f"Name: {tem.name} Age: {tem.age} Breed: {tem.breed} Date: {tem.date}")
        print("-----")
        tem = tem.next
```

Este método es solo para imprimir todos los animales disponibles con sus respectivos datos, es decir, imprime todos los datos almacenados en cada nodo de nuestra fila recorriendo de 1 en 1 dicha lista.

```
def dog_dequeue(self):
    tem = self.first
    if self.first==None:
        return False
    else:
        if self.first==self.last:
            self.first=None
            self.last=None
        else:
            self.first=tem.next
            tem.next=None

    self.length-=1
    return tem
```

Por último, el método dequeue se encarga de sacar el primero en la fila, y retornarnos el elemento sacado, disminuyendo el tamaño de la fila en 1.

```
my_queue=Queue()
selection=0
while True:
    print("\n")
    print("1.-Ingresar un Animalito")
    print("2.-Imprimir los animalitos disponibles")
    print("3.-Adoptar un Animalito")
    print("4.-Salir")
    selection=int(input())
    match selection:
        case 1:
            name=str(input("Ingresa el nombre del Animalito: \n"))
            age=int(input("Ingresa la edad del Animalito: \n"))
            breed=str(input("Ingresa la raza del Animalito: \n"))
            my_queue.dog_enqueue(name,age,breed)
        case 2:
            my_queue.print_queue()
        case 3:
            r = my_queue.dog_dequeue()
            if r==False:
                print("De momento ya no tenemos mas animalitos en adopcion")
            else:
                print("-----")
                print(f"Name: {r.name} Age: {r.age} Breed: {r.breed} Date: {r.date}")
                print("-----")
        case 4:
            print("Saliendo...")
            exit()
```

Esta última sección fuera de las clases se encarga de ser un menú repetitivo el cual permite acceder a cada función de manera específica, consta de un switch dentro de un ciclo el cual dependiendo el case seleccionado por el usuario llamara a uno u otro método de la clase.

### Prueba de Escritorio:

```
Animal_Shelter.py x
Run: Animal_Shelter x
1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
1
Ingresa el nombre del Animalito:
Fred
Ingresa la edad del Animalito:
2
Ingresa la raza del Animalito:
Chihuahua

1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
1
Ingresa el nombre del Animalito:
Fred 2
Ingresa la edad del Animalito:
1
Ingresa la raza del Animalito:
Pastor Aleman

1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
2
-----
Name: Fred Age: 2 Breed: Chihuahua Date: 2023-03-22 18:11:06.890143
-----
-----
Name: Fred 2 Age: 1 Breed: Pastor Aleman Date: 2023-03-22 18:11:44.714680
-----
```

Durante esta breve sección de código, ingresamos 2 animalitos para probar nuestro método enqueue, podemos notar que nos pide el nombre, la edad y la raza de manera individual en cada línea.

Para comprobar que realmente se formó de manera efectiva la lista y los datos se almacenaron imprimimos la lista completa, para observar que los 2 datos de los 2 animalitos se guardaron de manera efectiva.

```
1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
```

3

```
-----
Name: Fred Age: 2 Breed: Chihuahua Date: 2023-03-22 18:11:06.890143
-----
```

Para comprobar el funcionamiento del método dequeue seleccionamos la opción 3 del menú y cómo podemos observar nos retorna el primer animalito ingresado con su información tal y como se trabaja en la estructura

```
1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
```

2

```
-----
Name: Fred 2 Age: 1 Breed: Pastor Aleman Date: 2023-03-22 18:11:44.714680
-----
```

```
1.-Ingresar un Animalito
2.-Imprimir los animalitos disponibles
3.-Adoptar un Animalito
4.-Salir
```

Con el fin de comprobar la salida de el animalito por el método dequeue imprimimos una vez mas la lista y cómo podemos observar el primero en ser ingresado ya no aparece, dando a entender que nuestro animalito ya no forma parte de la fila y ha sido eliminado con éxito.