

Tecnicatura Superior en Telecomunicaciones

Módulo Programador

Evidencia N°2: Gestión de Rodeos

El programa se orienta a la gestión de rodeos ganaderos dedicada al engorde de ganado vacuno. Una correcta categorización del ganado es crucial para que se provea a los animales con la dieta correcta para maximizar la ganancia de peso. El programa propuesto permite al usuario crear los rodeos (categorías) y asignar el rango de peso mínimo y máximo de la categoría. El programa permite registrar los animales, identificándose con el número de caravana, y el programa los asigna al rodeo correspondiente. El programa permite registrar los pesajes realizados a los animales para monitorear la ganancia de peso individual y agregado.

Programa

El programa consta de un archivo main.py, que es el punto de entrada de programa y que tiene definida las funciones para ejecutar el menú de opciones. La carpeta modules contiene dos archivos, rodeos.py y animales.py. Cada uno de ellos contiene una clase homónima cuyos métodos permiten gestionar los rodeos y los animales.

main.py

```
import os
from modules.rodeos import Rodeo
from modules.animales import Animales
def menu_principal():
```

```
'''Muestra el menú principal y retorna la opción
seleccionada.'''
print("\nMenú Principal:")
print("1. Gestión de Rodeos")
print("2. Gestión de Animales")
print("3. Salir")

opcion = int(input("Ingrese una opción: "))
os.system("clear")
return opcion

def gestion_rodeos():
    """Muestra las opciones de gestión de rodeos y retorna la
opción seleccionada."""
    print("\nGestión de Rodeos:")
    print("1. Crear Rodeo")
    print("2. Modificar Rodeo")
    print("3. Eliminar Rodeo")
    print("4. Volver al menú principal")

    opcion = int(input("Ingrese una opción: "))
    return opcion

def gestion_animales():
    """Muestra las opciones de gestión de animales y retorna la
opción seleccionada."""
    print("\nGestión de Animales:")
    print("1. Alta de Caravana")
    print("2. Registro de Peso")
    print("3. Baja de Caravana")
    print("4. Volver al menú principal")

    opcion = int(input("Ingrese una opción: "))
    return opcion
```

```
'''Ejecuta el menu'''
while True:
    opcion = menu_principal()
    if opcion == 1:
        opcion_getion_rodeos = gestion_rodeos()
        if opcion_getion_rodeos == 1:
            Rodeo().crear_rodeo()
            input("\nPresione ENTER para continuar")
            os.system("clear")
            continue
        elif opcion_getion_rodeos == 2:
            Rodeo().modificar_rodeo()
            input("\nPresione ENTER para continuar")
            os.system("clear")
            continue
        elif opcion_getion_rodeos == 3:
            Rodeo().eliminar_rodeo()
            input("\nPresione ENTER para continuar")
            os.system("clear")
            continue
        elif opcion_getion_rodeos == 4:
            os.system("clear")
            continue
        else:
            print("Opción inválida")
            input("\nPresione ENTER para continuar")
            os.system("clear")
            continue
    elif opcion == 2:
        opcion_gestion_animales = gestion_animales()
        if opcion_gestion_animales == 1:
            Animales().alta_caravana()
            input("\nPresione ENTER para continuar")
            os.system("clear")
```

```

        continue
    elif opcion_gestion_animales == 2:
        Animales().registro_peso()
        input("\nPresione ENTER para continuar")
        os.system("clear")
        continue
    elif opcion_gestion_animales == 3:
        Animales().baja_caravana()
        input("\nPresione ENTER para continuar")
        os.system("clear")
        continue
    elif opcion_gestion_animales == 4:
        os.system("clear")
        continue
    else:
        print("Opción inválida")
        input("\nPresione ENTER para continuar")
        os.system("clear")
        continue
elif opcion == 3:
    break
else:
    print("Opción inválida")
    input("\nPresione ENTER para continuar")
    os.system("clear")

```

modules/animales.py

```

import os

class Animales():
    def __init__(self):
        self.caravana = ''
        self.peso = ''

```

```

self.fecha = ''

def alta_caravana(self):
    '''
        Este método da de alta la caravana asociada a un animal.
        El usuario debe ingresar el número de caravana, peso,
        fecha del alta.

        TODO:
        No puede registrarse caravanas si no ha rodeos creados
        Agregar lógica para asignar el animal a un rodeo de
        acuerdo a su peso
        El estado de la caravana al registrarse es activa
        Si el animal no alcanza el peso mínimo de ninguna
        categoria de rodeo se rechaza
    '''
    print("ALTA DE CARAVANA")
    os.system("clear")
    print("Este método crea una nueva caravana")
    self.caravana = input("Ingrese caravana: ")
    self.peso = float(input("Ingrese el peso: "))
    self.fecha = input("Ingrese la fecha de la caravana: ")
    print("Caravana creada")

def registro_peso(self):
    '''
        Este método registra el peso de un animal en una fecha

        TODO:
        Verificar que caravana existe
        Verificar que la caravana no ha sido dada de baja
        Guardar el nuevo registro en la tabla correspondiente
        Actualizar el peso relacionado a la caravana de la
        caravana
    '''

```

```

'''
print("REGISTRO DE PESO")
os.system("clear")
print("Este método registra el peso de la caravana")
self.caravana = input("Ingrese la caravana: ")
self.peso = float(input("Ingrese el peso del animal: "))
self.fecha = input("Ingrese la fecha del pesaje: ")
print("Peso registrado")

def baja_caravana(self):
    '''
    Este método da de baja la caravana asociada a un animal.
    El usuario debe ingresar el número de caravana, la razón
y la fecha de baja.
    '''
    print("BAJA DE CARAVANA")
    os.system("clear")
    print("Este método elimina una caravana")
    self.caravana = input("Ingrese el nombre de la caravana: ")
    self.fecha = input("Ingrese fecha de la baja: ")
    razon = input("Razón de la baja: ")
    print("Caravana dada de baja")

```

modules/rodeos.py

```

import os
class Rodeo():
    def __init__(self):
        self.nombre = ''
        self.peso_min = ''
        self.peso_max = ''

    def crear_rodeo(self):
        '''

```

Este método crea un nuevo rodeo.

El usuario debe ingresar el nombre del rodeo, el peso mínimo y el peso máximo.

TODO:

Cuando se crea un rodeo se debe crear un rodeo por defecto llamado rechazo, cuyo valor de

peso minimo es 0 el el peso máximo es menor al rango del rodeo de menor peso creado por el usuario

'''

```
print("CREAR RODEO")
```

```
os.system("clear")
```

```
print("Este método crea un nuevo rodeo")
```

```
self.nombre = input("Nombre del Rodeo: ")
```

```
self.peso_min = float(input("Ingresa peso mínimo: "))
```

```
self.peso_max = float(input("Ingresa peso maximo: "))
```

```
print("Rodeo creado")
```

```
def modificar_rodeo(self):
```

'''

Este método modifica un rodeo.

El usuario debe ingresar el nombre del rodeo, el peso mínimo y el peso máximo.

TODO:

Obtener el listado de rodeos desde la base de datos

Evitar el solapamineto de los pasos minimos y máximos respecto a los otros rodeos.

Mostrar la lista de rodeos con sus datos actualizados

'''

```
print("MODIFICAR RODEO")
```

```
os.system("clear")
```

```
print("Este método modifica un rodeo")
```

```
self.nombre = input("Nombre del Rodeo: ")
```

```
self.peso_min = float(input("Ingrese peso mínimo: "))
self.peso_max = float(input("Ingrese peso maximo: "))
print("Rodeo modificado")

def eliminar_rodeo(self):
    '''
    Este método elimina un rodeo.
    '''
    print("Este método elimina un rodeo")
    print("Rodeo eliminado")
```

Base De Datos

La base de datos consta de cuatro tablas: rodeos, caravanas, pesajes, baja_caravanas

El esquema de la base de datos se puede consultar en:

<https://dbdiagram.io/d/Gestion-de-Rodeos-664943bcf84ecd1d228af6d5>

En la carpeta documentos se comparte la versión pdf del esquema