



## ZOOLÓGICO

### Definición

- Desarrollar un zoológico virtual que simula la experiencia de un zoológico real.

El programa consta de dos modos, visitante y administrador.

- **El modo visitante**
  - Para entrar en este modo seleccionaremos 1, automáticamente aparecera una pantalla mostrando información sobre las entradas, seguidamente nos preguntará cuántas entradas queremos comprar, introduciremos los datos del cliente y pasaremos al menú de inicio, aquí podremos observar varios apartados con las distintas secciones del zoo, cuando entremos en una nos mostrará una imagen de la sección y nos dirá los animales que alberga, pudiendo ver información de estos animale además de una foto si la hay.
- **El modo administrador**
  - Desde este modo podremos administrar las distintas secciones del zoo, así como los empleados, clientes, animales, ganancias y zonas. En cada uno de estos apartados tendremos un menú:
    - **Clientes**
      - Podemos eliminar clientes y ver todos los clientes que hay
    - **Empleados**
      - Podemos contratar empleados, verlos y despedirlos
    - **Animales**
      - En este apartado podemos ver animales y agregarlos a las distintas zonas
    - **Ganancias**
      - Aquí se muestran todas las ganancias que ha generado el zoo
    - **Zonas**
      - Podremos elegir entre ver zonas o agregar zonas al zoo

### Líneas de investigación futuras:

- En un futuro se podrían implementar más áreas al zoo, así como una función que permite ver los problemas por solucionar del zoo.
- A su vez podríamos añadir la función tanto de eliminar animales como zonas.
- También podríamos añadirle base de datos y una interfaz gráfica para hacer el programa más amigable.

### Amenazas

- La competencia
  - Existen más programas que simulan la experiencia en zoo, estos pueden llegar a ser más amigables y completo del mio, pero debido a cuestiones personales no puedo desarrollar más este proyecto.





- Robo de datos
  - El programa carece de seguridad, esto puede conllevar a robo de código o de datos

### Oportunidades

- Pese a todo el programa es bastante completo y amigable lo que puede resultar en una experiencia gratificante y ser de mucha ayuda a la hora de gestionar un zoo

### Puntos fuertes

- Interfaz amigable y clara
- Cumple con las necesidades básicas
- Experiencia de visita gratificante
- Gestión de administración cómoda y eficaz

### Puntos débiles

- Falta de la existencia de una base de datos
- Carencia de interfaz gráfica
- Falta de algunos apartados en el menú de administrador

### Pseudocódigo

El programa en completo consta de 8 archivos y una carpeta, empezare describiendo cada una de las funciones de los archivos

archivo registro

Este archivo contiene una clase la cual alberga los diferentes diccionarios y arrays que serán utilizados por todos los archivos del usuario, funciona como una “base de datos”.

### archivo animal

En este se encuentra la clase que albergara los distintos atributos que tengan los animales así como un método que permitirá ver una breve descripción de estos, la clase consta de un constructor el cual debe recibir los parámetros `nombre`, `especie`, `dieta`, `habitat`, `enlace` todos estos se almacenaran en sus respectivos atributos además guardar el nombre del animal con su especie en un diccionario de registro

### archivo zona

Este archivo es uno de los principales ya que en este se crean las zonas y se agrega los animales, esto lo hace a través de dos clases una la clase zoo.

Esta es la encargada de crear las zonas y mostrarlas a través de dos métodos distintos, el método `crear_zona` el cual hay que pasarle los parámetros: `nombre=None`, `capacidad=None`, `clima=None`, `enlace=None`





siendo todos estos opcionales, este método se encarga de instanciar las zonas llamando a la otra clase “Zona” la que se encarga de almacenar la información de las zonas, pero además esta también se encarga de agregarlos animales entre otras muchas cosas, después de instanciar el objeto que identifica a la zona lo guarda en una lista del archivo registro y más tarde te informa de que la zona se ha sido creada.

El método mostrar\_zona para ver las distintas zonas del zoo y el método escoger zona que se usará para poder encontrar los objetos que identifican las distintas zonas.

La clase zona se encarga de almacenar los atributos de las zonas pero también tiene varios métodos:

- agregar\_anima: este método se encarga de añadir una animal llamando a la clase definida en el otro archivo
- ver\_animales\_extp: sirve para ver los animales de una determinada zona esto lo consigue imprimiendo un diccionario con los animales de esa zona y preguntando por el animal que deseas ver mas información llamando posteriormente al método ver\_animal del archivo animal.
- ver\_animales\_ext: este métodos es exactamente igual que el otro solo que no se limita a los animales de una determinada zona, este permite ver todos los animales del zoo.
- ver\_zona: imprime los detalles de una determinada zona además de abrir una imagen de esta.

En este archivo vienen las zonas y los animales predefinidos.

### archivo clientes

Su función principal es almacenar la información de todos los clientes que visiten el zoo

Para ello se usan varias funciones:

- registrar: el cual pregunta por los datos del cliente y los almacena, para ello primero instancia un objeto y luego ajusta los distintos atributos de este.
- eliminar\_usu: Elimina todo lo relacionado con un cliente
- ver\_t: Muestra todos los datos de un cliente

### archivo entradas

este archivo contiene una función la cual pregunta cuántas entradas se compraran después de que las indiques determinará a través de una condicional si sois un grupo o no en función de esto se hará descuento o no, más tarde, entra en un bucle el cual va llamando a la función registrar y en función de tus datos determina cuanto cuesta tu entrada y gestiona las ganancias.

### archivo empleados





En este archivo se define la clase empleados la cual contiene los distintos atributos que debe tener un empleado

En este archivo hay varias funciones:

- Contratar\_empleado: la cual pide los datos del personal, para más tarde instanciar un objeto el cual ajusta los distintos atributos del empleado, por último este se añade a una lista guardada en registro.
- Despedir\_empleado: Elimina al usuario restableciendo los atributos del objeto instanciándolo de nuevo y eliminándolo de las listas en las cuales están guardados.
- mostrar\_empleados: Muestra la información de los empleados a través de un bucle que llama a los distintos objetos guardados en la lista y estos se imprimen junto a sus datos.

En este archivo ya vienen definidos algunos de los empleados por defecto

### archivo interfaces

En este archivo únicamente vienen algunas variables definidas que serán nuestras interfaces en el programa final

### archivo proyecto

Este es el archivo final el cual inicia importando todos los módulos y metiéndose en un bucle que primero muestra una interfaz y después pregunta por qué quieres hacer si entrar en modo visitante o en modo administrador.

Ambas opciones están en un match case y ambas opciones entran en un bucle.

Si pulsamos 1 primero pasaremos por la fase de entradas el cual nos dará la información de estas. Después llamará a la función entradas y nos adentraremos en un bucle en el cual se muestra una interfaz con las distintas áreas del zoo, preguntando qué queremos hacer y a través de un match case nos llevará a una u otra.

Si pulsamos 2 entraremos en la opción administrador en esta hay varios apartados como: clientes empleados zonas ganancias y animales.

Este modo está orientado a la gestión del zoo llamando a las funciones definidas a través de un match case y va llamando a las distintas funciones definidas en los otros archivos

