

Presentación Pedagógica

Proyecto DataPet – Fundamentos de Python

Grado 11

Bienvenidos

Desarrollo de Software con Python

Proyecto:

DataPet

Sistema de Gestión para Clínica Veterinaria

Objetivo de la Clase

Al finalizar esta unidad los estudiantes podrán:

- Comprender cómo se construye un software desde cero.
 - Aplicar lógica de programación.
 - Utilizar estructuras de datos.
 - Trabajar con archivos JSON.
 - Crear menús interactivos.
 - Guardar información de manera permanente.
 - Desarrollar un sistema funcional en Python.
-

¿Qué es Python?

Python es un lenguaje de programación:

- Fácil de aprender.
- Muy utilizado en el mundo.
- Utilizado en:
 - Inteligencia Artificial.
 - Desarrollo web.
 - Videojuegos.
 - Automatización.

- Ciencia de datos.
 - Aplicaciones de escritorio.
-

¿Qué Vamos a Construir?

DataPet

Un software que permitirá:

- Registrar mascotas.
 - Consultar información.
 - Modificar datos.
 - Eliminar registros.
 - Guardar información usando JSON.
-

Conceptos que Necesitamos Aprender

Temas Fundamentales

- Variables
 - Entrada y salida de datos
 - Condicionales
 - Ciclos
 - Funciones
 - Listas
 - Diccionarios
 - Librerías
 - Archivos JSON
 - CRUD
 - Validaciones
-

Variables

¿Qué son?

Son espacios donde almacenamos información.

Ejemplo

```
nombre = "Max"  
edad = 5  
peso = 12.5
```

Tipos de Datos

Tipo	Ejemplo
Texto	"Firulais"
Entero	10
Decimal	12.5
Booleano	True

Entrada de Datos

input()

Permite ingresar información.

```
nombre = input("Ingrese nombre: ")
```

Salida de Datos

print()

Muestra información en pantalla.

```
print("Bienvenido a DataPet")
```

Condicionales

if – elif – else

Permiten tomar decisiones.

```
edad = 5

if edad >= 5:
    print("Es adulto")
else:
    print("Es cachorro")
```

Ciclos

while

Repite instrucciones.

```
while True:
    print("Menú principal")
```

Ciclo for

Permite recorrer listas.

```
for numero in range(5):
    print(numero)
```

Funciones

¿Qué son?

Bloques de código reutilizables.

Ejemplo

```
def saludar():  
    print("Hola")
```

¿Por Qué Usar Funciones?

Las funciones permiten:

- Organizar el código.
- Reutilizar procesos.
- Facilitar el mantenimiento.
- Dividir el programa en partes.

Listas

¿Qué son?

Estructuras que almacenan varios datos.

Ejemplo

```
mascotas = ["Max", "Luna", "Rocky"]
```

Operaciones con Listas

Agregar datos

```
mascotas.append("Firulaïs")
```

Recorrer listas

```
for mascota in mascotas:  
    print(mascota)
```

Diccionarios

¿Qué son?

Permiten guardar información organizada.

Ejemplo

```
mascota = {  
    "nombre": "Max",  
    "edad": 4,  
    "raza": "Labrador"  
}
```

¿Por Qué Usaremos Diccionarios?

Porque cada mascota tendrá:

- Nombre
- Edad
- Raza
- Dueño
- Identificación
- Teléfono

Y todo quedará organizado.

Librerías

¿Qué son?

Son herramientas ya creadas que Python ofrece.

Librería JSON

import json

Nos permitirá:

- Guardar información.
- Leer información.

- Trabajar como una base de datos sencilla.

```
import json
```

¿Qué es JSON?

JSON significa:

JavaScript Object Notation

Es un formato para guardar datos.

Ejemplo de JSON

```
[
  {
    "id": "1",
    "nombre": "Max",
    "raza": "Labrador"
  }
]
```

¿Por Qué Usaremos JSON?

Porque permite:

- Guardar información permanentemente.
- Simular una base de datos.
- Leer datos fácilmente.
- Compartir información.

CRUD

Operaciones Básicas del Software

Letra	Acción
C	Create (Crear)

Letra	Acción
R	Read (Consultar)
U	Update (Modificar)
D	Delete (Eliminar)

Validaciones

¿Por Qué Son Importantes?

Evitan errores.

Ejemplo:

- Que el usuario escriba letras donde deben ir números.

isdigit()

Validación Numérica

```
numero = input("Ingrese identificación: ")

if numero.isdigit():
    print("Correcto")
else:
    print("Solo números")
```

Menú del Sistema

```
===== DATAPET =====

1. Registrar mascota
2. Consultar mascota
3. Modificar mascota
4. Eliminar mascota
5. Salir
```


Estructura del Proyecto

```
DataPet/  
|  
├─ datapet.py  
└─ mascotas.json
```

¿Qué Aprenderemos Durante el Proyecto?

- Pensamiento lógico.
- Programación estructurada.
- Manejo de errores.
- Organización de software.
- Almacenamiento de información.
- Desarrollo de aplicaciones reales.

Recomendaciones para Programar

- Leer bien los errores.
- Comentar el código.
- Probar paso a paso.
- Tener paciencia.
- Practicar constantemente.

Metodología de Trabajo

Construiremos el proyecto por etapas:

1. Menú principal.
2. Registro de mascotas.
3. Consultas.
4. Modificaciones.
5. Eliminación.
6. JSON.
7. Mejoras del sistema.

Resultado Final Esperado

Los estudiantes desarrollarán:

DataPet

Un sistema funcional capaz de:

- Registrar mascotas.
 - Gestionar información.
 - Guardar datos en JSON.
 - Aplicar lógica de programación.
-

Reflexión Final

“Programar no es memorizar código.

Programar es aprender a resolver problemas.”

¡Comencemos a Construir DataPet!