



UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE

SEDE SANTO DOMINGO DE LOS TSÁCHILAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN



PERIODO	:	202450 Mayo – Sept 24
ASIGNATURA	:	Programación Orientada a Objetos
TEMA	:	Proyecto Grupal
INTEGRANTES	:	Ana Lombeida Noelia Mendoza
NIVEL-PARALELO - NRC:		Nrc. 15279
DOCENTE	:	Ing. Verónica Martínez C., Mgs.
FECHA DE ENTREGA	:	___ de julio del 2024

SANTO DOMINGO – ECUADOR

Programación Orientada a Objetos

1. Introducción

La adopción de mascotas en un acto de amor y responsabilidad

2. Objetivos

2.1. Objetivo General

Crear un sistema de adopción de mascotas aplicando todos los temas aprendidos durante el parcial para el gestionamiento de adopción de mascotas en pequeños refugios, de esta manera tener un control y seguimiento de las mascotas y así buscarles a las mascotas un buen hogar.

2.2. Objetivos Específicos

- Realizar los diagramas de clases y de usos para tener la base de nuestro sistema de adopción de mascotas.
- Aplicar los temas aprendidos durante el parcial como abstracción, encapsulamiento, polimorfismo, herencia y otros en la creación del sistema de adopción de mascotas.
- Realizar un sistema que ayude en la gestión de adopción de mascotas en pequeños refugios, que controle y realice un seguimiento a la mascota y al adoptante.

Programación Orientada a Objetos

Marco Teorico

GitHub

Es una plataforma en línea que utiliza Git, un sistema de control de versiones, para que los desarrolladores trabajen juntos en proyectos y registren los cambios realizados, manteniendo un registro de su evolución (HOSTINGER, 2023). Además integra las capacidades de control de versiones de Git para gestionar proyectos. Permite a los usuarios rastrear y gestionar los cambios en el código fuente en tiempo real, junto con otras funciones de Git, todo desde una interfaz de usuario más accesible.

La Programación Orientada a Objetos (POO) se basa en cuatro pilares fundamentales que permiten crear software modular, reutilizable y mantenible:

1. Abstracción:

La abstracción permite ocultar los detalles de implementación de un objeto y exponer solo las características y funcionalidades relevantes para su uso (Cueva, 2020).

2. Encapsulamiento:

El encapsulamiento combina los datos el comportamiento de un objeto en una unidad autosuficiente. Esto protege los datos internos del objeto y controla el acceso a ellos a través de los métodos, promoviendo la seguridad del código.

3. Herencia:

La herencia permite crear clases que heredan las propiedades y métodos de otras clases, creando una jerarquía de clases. Esto promueve la reutilización de código y facilita la especialización de clases a partir de clases más generales.

4. Polimorfismo:

El polimorfismo permite que objetos de diferentes clases respondan al mismo mensaje de maneras distintas. Esto se logra definiendo métodos con el mismo nombre pero diferentes implementaciones en cada clase derivada.

Definición de clases, objetos, atributos y métodos.

Clases: Son categorías de objetos que representan un tipo particular de objetos con características y comportamiento similar. Es un modelo o plantilla que define las características y atributos o comportamientos de una entidad.

Objetos: Son instancias de una clase. Representan entidades del mundo real que tienen características y comportamientos específicos definidos por la clase a la que pertenecen.

Atributos: Son las características o propiedades de un objeto que lo describen y definen su estado. Los atributos son variables que están asociadas a un objeto y representan datos específicos de ese objeto.

Programación Orientada a Objetos

```
//////// Atributos
private String tipoMascota;
private String nombre;
private String genero;
private String raza;
private int nVacunas;
```

1.1.1. UML: Diagramas de Casos de Uso.

Un diagrama de casos de uso UML (Lenguaje Unificado de Modelado) es una herramienta visual que describe los diferentes usos o funcionalidades que un sistema ofrece a sus usuarios (Gaskin, 2022). Se utiliza para modelar las interacciones entre el sistema y los actores externos (usuarios o entidades que interactúan con el sistema).

1.1.1. UML: Diagrama de clases.

Un diagrama de clases UML (Lenguaje Unificado de Modelado) muestra visualmente las clases de un sistema con sus atributos y operaciones, organizadas en rectángulos que contienen el nombre de la clase, sus atributos y operaciones (Fonseca, 2022). Esencialmente, es un plano que describe cómo se organizan las "piezas" de un sistema.

10.3. Modificadores de acceso

Los modificadores de acceso en Java son palabras clave que se utilizan para controlar el nivel de visibilidad de los atributos y métodos de una clase.

Public (public): Acceso desde cualquier parte del programa.

Protected (protected): Acceso dentro del mismo paquete y subclases.

Privado (private): Acceso solo dentro de la misma clase.

12 Métodos getters, setters.

Se utilizan para controlar el acceso y la modificación de los atributos de una clase, siguiendo el principio de encapsulamiento.

Getters: Un método getter se utiliza para obtener el valor de un atributo privado de una clase.

Setters: Un método setter se utiliza para asignar un valor a un atributo privado de una clase.

Arreglos y colecciones

Concepto de arreglos

Un arreglo es una estructura de datos que permite almacenar una colección de elementos del mismo tipo de manera ordenada y accesible. Los arreglos son esenciales para organizar y gestionar conjuntos de datos.

Entre las colecciones más comunes están List, ArrayList y LinkedList.

Programación Orientada a Objetos

Ventajas de los arreglos null

Los arreglos null son un concepto que se utiliza en algunos lenguajes de programación para representar un arreglo que no ha sido inicializado o que no apunta a ningún objeto en memoria.

Clasificación de los arreglos (Con datos Primitivos y objetos)

En base al tipo de datos que almacenan, los arreglos se pueden clasificar en dos categorías principales:

Arreglos de Datos Primitivos:

- Almacenan valores primitivos directamente.
- Más eficientes en términos de uso de memoria y velocidad.
- **Tipos:** int[], float[], double[], char[], boolean[], etc

Arreglos de Objetos:

- Almacenan referencias a objetos.
- Pueden contener cualquier tipo de objeto, incluidas instancias de clases personalizadas.
- **Tipos:** String[], Persona[], Integer[], Double[], etc.

Desarrollo

Para la realización del sistema lo primero que se realizó fue la creación de los diagramas de clases y de usos a través de la plataforma draw.io, de esta manera establecimos la base y estructura del programa.

Diagrama de Clases del Sistema de Adopción de Mascotas.

Diagrama de Usos del Sistema de Adopción de Mascotas.

1. Conclusiones

Recomendaciones

5. Bibliografía

- Belmonte, Ó. (s.f.). *Programación Avanzada. Programación Orientada a Objetos. Excepciones*. Universitat Jaume I.
- Charte, F. (19 de agosto de 2019). *Paquetes en Java: qué son, para qué se utilizan, y cómo se usan*. Obtenido de <https://www.campusmvp.es/recursos/post/paquetes-en-java-que-son-para-que-se-utilizan-y-como-se-usan.aspx>
- Coppola, M. (16 de marzo de 2023). *Qué es un constructor en Java, sus tipos y cómo implementarlo*. Obtenido de HubSpot: <https://blog.hubspot.es/website/que-es-constructor-java>

Programación Orientada a Objetos

Cueva, J. (2020). *Entendiendo los principios de la Programación orientada a objetos*.

Fonseca, L. (27 de junio de 2022). *Cómo crear un diagrama de clases*. Obtenido de <https://es.venngage.com/blog/diagrama-de-clases/>

Gaskin, J. (12 de junio de 2022). *Todo lo que necesitas saber sobre el diagrama de caso de uso*. Obtenido de <https://es.venngage.com/blog/diagrama-de-caso-de-uso/>

HOSTINGER. (2023 de enero de 2023). *Qué es GitHub y cómo empezar a usarlo*. Obtenido de <https://www.hostinger.es/tutoriales/que-es-github>

Quispe, A. (20 de marzo de 2024). *Manejo de Archivos en Java: Guía Completa*.

Obtenido de Medium: [https://amsoft.medium.com/manejo-de-archivos-en-java-gu%C3%ADa-completa-](https://amsoft.medium.com/manejo-de-archivos-en-java-gu%C3%ADa-completa-8d0f64ee5a5b#:~:text=En%20Java%2C%20un%20archivo%20se,las%20herramientas%20necesarias%20para%20manipularlos)

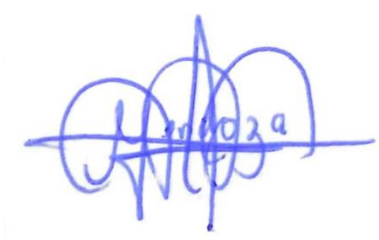
[8d0f64ee5a5b#:~:text=En%20Java%2C%20un%20archivo%20se,las%20herramientas%20necesarias%20para%20manipularlos](https://amsoft.medium.com/manejo-de-archivos-en-java-gu%C3%ADa-completa-8d0f64ee5a5b#:~:text=En%20Java%2C%20un%20archivo%20se,las%20herramientas%20necesarias%20para%20manipularlos).

7. Anexos

8. Legalización de documento

Nombres y Apellidos: Noelia Kassandra Mendoza Jiler.

CI: 1723912208



Firma

Nombres y Apellidos: Ana Lombeida

CI:

Firma