|  |  |
| --- | --- |
| **Objetivos:**   1. Analizar principales requerimientos. 2. Diseñar diagrama de clases de la aplicación**.** 3. Realizar un ejercicio con interfaz gráfica de usuario (GUI) en Java. 4. Almacenar información en la memoria temporal. 5. Serializar la información de un requerimiento. 6. Implementar excepciones para manejo de archivos. | **Recursos:**   * Internet * Eclipse o Netbeans. * Libro: Introducción a la estructura de datos - Jorge Villalobos. * GitHub ejercicios: <https://github.com/ThiagoAndres/EstructuraDatos1-IV-UNIMAR> |
| **Proyecto:**  **VETERINARIA**  **Descripción:**  En una veterinaria se hacen diferentes procedimientos para cada tipo y raza de animal. Para agendar una cita con la veterinaria debe permitir al usuario ingresar los datos de cada animal que registre en el programa: nombre, animal, raza, edad, teléfono y que síntomas presenta. De esta manera debe ordenar a estos animales por orden de edad de menor a mayor.  El usuario del programa debe permitir ver todos los datos registrados para asi poder agendar una cita.  La aplicación debe:  1.- Permite al usuario agregar un propietario.  2.- Permite al usuario agregar un animal.  3.- Permite eliminar un animal registrado.  4.- Permite modificar un animal registrado.  5.- Permite ordenar por edad de menor a mayor.  6.- Mostrar todos los datos del animal registrado.  7.- Genera un recibo.  **Autoría:**   |  |  | | --- | --- | | Proyecto Curso: | Estructuras de datos I | | Ejercicio: | Veterinaria | | Autores: | 1. Nicolas Santiago Maya Delgado | | 1. Santiago David Lina Patiño | |  | | Fecha realización: |  |   **Listado y comprensión de Requerimientos:**   |  |  | | --- | --- | | Nombre | R1- Agregar un animal. | | Resumen | Permite al usuario agregar un animal. | | Entradas | nombre, animal, raza, edad, teléfono, síntomas. | | Resultados | Se agrega al programa un animal. |  |  |  | | --- | --- | | Nombre | R2- Eliminar un animal registrado. | | Resumen | Elimina un animal que el usuario ya haya registrado anteriormente. | | Entradas | id | | Resultados | Se elimina el animal deseado. |  |  |  | | --- | --- | | Nombre | R3- Modificar un animal registrado. | | Resumen | Permite al usuario modificar algún dato del animal ya registrado. | | Entradas | id | | Resultados | Se modifica el dato del animal ya registrado. |  |  |  | | --- | --- | | Nombre | R4- Ordenar por edad de menor a mayor. | | Resumen | Permite ordenar a los animales registrados por su edad de menor a mayor. | | Entradas | edad | | Resultados | Se ordena por edad de manera ascendente. |  |  |  | | --- | --- | | Nombre | R5- Mostrar todos los datos del animal registrado. | | Resumen | Muestra al usuario los animales registrados con sus datos. | | Entradas | ninguna | | Resultados | Se muestran los datos de cada animal al usuario. |  |  |  | | --- | --- | | Nombre | R6- Generar recibo | | Resumen | Permite generar un recibo al usuario. | | Entradas | ninguna | | Resultados | Genera un recibo al usuario. |  |  |  | | --- | --- | | Nombre | R7- Agregar propietario del animal. | | Resumen | Permite agregar un propietario a su debido animal. | | Entradas | nombre, teléfono, id | | Resultados | Se agrega un propietario del animal. |   **Modelo conceptual (o diagrama de clases)**    **Requisitos Técnicos:**  1. Aplicar conceptos de programación orientada a objetos para modelar los datos y la funcionalidad del reproductor.  2. Crear una interfaz en donde los usuarios puedan ver la exposición de perros.  3. Implementar la serialización de objetos para almacenar y recuperar la información de la exposición de perros.  4. Utilizar un ArrayList para almacenar los objetos de manera temporal.  5. Realizar un adecuado uso de las excepciones.  **Pasos Sugeridos:**  1. Diseñar una clase `Perro` que contenga los atributos necesarios para representar un perro en la exposición.  2. Implementar métodos en la clase `ExposicionPerros` como principal.  3. Proponga una interfaz gráfica de usuario amigable  **Entregables Esperados:**  1. Debe crear un repositorio en GIT para el proyecto.  2. Documentación que explique la estructura del proyecto, boceto de la interfaz de usuario.  3. Capturas de pantalla que muestren la aplicación en funcionamiento.  **Criterios de Evaluación:**  Los proyectos serán evaluados en función de la correcta implementación de los requisitos funcionales y técnicos, la aplicación adecuada de la programación orientada a objetos, así como la usabilidad y apariencia de la interfaz. Los estudiantes también deben ser capaces de explicar y justificar las decisiones de diseño y tecnología que tomaron durante el desarrollo del proyecto. Con ello:   1. Todos los requisitos funcionales desarrollados. 2. Utilización de la programación orientada a objetos. 3. Funcionalidad, el programa debe funcionar en su totalidad sin ningún error. 4. Usabilidad, el programa debe ser fácil de utilizar, no debe dejar dudas al usuario. 5. El archivo en un repositorio GIT de forma adecuada. 6. Código documentado en su totalidad. 7. Puntualidad en la entrega 8. Esta documentación completa con el prototipo de la interfaz.   **Prototipo de interfaz (Ejemplo - Libre edición)**    Comience aquí a documentar todo el procedimiento que hacen como equipo. | | |