|  |  |
| --- | --- |
| **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:**  Desarrollo de un sitio web para una empresa hídrica  **Descripción:**  Una empresa hídrica es la encargada de garantizar un buen servicio a la sociedad ya sea con la instalación, y el mantenimiento de esta misma. También es muy importante la organización del servicio, y del consumo de agua que se realice, para esto es necesario un sitio web que permita el registro de las instalaciones o familias que necesiten el servicio hídrico, el cliente que adquiera este servicio deberá proporcionar información como el Nombre, ID, teléfono, el sistema sera el encargado de realizar el registro automático y de generar las facturas electrónicas del consumo que se haga durante un determinado tiempo.  La aplicación debe:  1.- Mostrar la lista de los perros registrados en la exposición, ordenada por raza, puntos o edad  2.- Mostrar la información de un perro específico  3.- Registrar un nuevo perro  4.- Localizar un perro por su nombre  5.- Buscar el perro ganador de la exposición (el que tiene un mayor puntaje asignado)  6.- Buscar el perro con el menor puntaje  7.- Buscar el perro más viejo de todos (con mayor edad).  La información inicial de los perros se obtiene de un archivo de propiedades. Al ejecutar el programa, se cargaesta información  **Autoría:**   |  |  | | --- | --- | | **Proyecto Curso:** | **Estructuras de datos I** | | **Ejercicio:** | Desarrollo de un sitio web para una empresa hídrica | | **Autores:** | 1. Karen Erazo | | 1. Johan Serrano | |  | | **Fecha realización:** | 01/08/2024 |   **Link de GitHub: <https://github.com/JohannSerrano/Estructura-De-Datos-I.git>**  **Listado y comprensión de Requerimientos:**   |  |  | | --- | --- | | **Nombre** | **R1- Registrar cliente** | | **Resumen** | Los clientes podrán ser registrados al sistema proporcionando sus datos personales | | **Entradas** | nombre  apellido  id  direccion  sector  estrato | | **Resultados** | Cliente registrado |  |  |  | | --- | --- | | **Nombre** | **R2- Registrar empleado** | | **Resumen** | Los empleados podrán ser registrados al sistema proporcionando sus datos personales | | **Entradas** | nombre  apellido  id  direccion  cargo | | **Resultados** | Empleado registrado |  |  |  | | --- | --- | | **Nombre** | **R3- Ingresar Empleado** | | **Resumen** | Permite ingresar al empleado al sistema teniendo en cuenta el nombre y el id | | **Entradas** | nombre  id | | **Resultados** | Empleado Ingresado |  |  |  | | --- | --- | | **Nombre** | **R4- Consultar Salario Empleado** | | **Resumen** | Permite al empleado consultar su salario dependiendo de las instalaciones y cancelaciones de servicios que haya hecho | | **Entradas** | numeroDeInstalaciones  numeroDeCancelaciones | | **Resultados** | Salario Consultado |  |  |  | | --- | --- | | **Nombre** | **R5- Ingresar cliente** | | **Resumen** | Permite ingresar al cliente al sistema teniendo en cuenta el nombre y el id | | **Entradas** | nombre  id | | **Resultados** | Cliente Ingresado |  |  |  | | --- | --- | | **Nombre** | **R6- Instalar Servicio de agua** | | **Resumen** | Asigna el costo de la tarifa de la instalación basada en el sector y el estrato | | **Entradas** | N/A | | **Resultados** | Visualización de Tarifa Asignada |  |  |  | | --- | --- | | **Nombre** | **R7- Consultar consumo mensual** | | **Resumen** | Permite consultar al usuario el consumo mensual de agua dependiendo a los litros que gasta durante el mes y el sector | | **Entradas** | cantidadDeLitros | | **Resultados** | Consumo del usuario presentado |  |  |  | | --- | --- | | **Nombre** | **R8- Modificar servicio de agua** | | **Resumen** | Permite al usuario modificar el sector y estrato del servicio ya instalado | | **Entradas** | nuevoSector  nuevoEstrato | | **Resultados** | Datos modificados |  |  |  | | --- | --- | | **Nombre** | **R9- Cancelar servicio de agua** | | **Resumen** | Permite al usuario cancelar su servicio de agua proporcionando su código personal asignado en la instalación del servicio | | **Entradas** | codigo | | **Resultados** | Servicio Cancelado |   **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. | | |