**FORMATO PARA DECLARACIÓN DE ESTRUCTURAS DE DATOS**

|  |  |
| --- | --- |
| **Nombre del Proyecto:** | **PRGCA:** Plataforma de reportes, gestión, control y seguimiento para la disminución de la contaminación ambiental en Cartagena. |
| **Nombre de los Integrantes:** | Esnneider Jose Matos Silgado  Rodolfo Jose Afanador Ricardo  Roswar Andres Diaz Villalobos. |
| **Fecha:** | Septiembre 2024 |

**ESTRUCTURAS DE DATOS UTILIZADAS:**

1. **Nombre de la Estructura de Datos:** Listas de usuarios

* **Descripción:** Estructura para almacenar la información de usuarios en una clase.
* **Tipo de Estructura:** Lista.
* **Implementación en el Código:** Se implementa utilizando una clase **ArrayList** donde cada elemento es un objeto que contiene la información del usuario.

En la entidad **Barrios:**

* private List<Reportes> reportes = new ArrayList<> ();

En la entidad **Contaminante:**

* private List<Reportes> reportes = new ArrayList<> ();

En la entidad **InformeAlerta**:

* private List<Contaminante> contaminantes = new ArrayList<> ();
* private List<Long> contaminantesIds;

1. **Nombre de la Estructura de Datos:** Page

* **Descripción:** Interfaz para contenido paginado que proporciona información sobre la página actual, total de elementos y páginas.
* **Tipo de Estructura:** Estructura de datos para paginación
* **Implementación en el Código:**

En **ReportesService:**

* public Page<Reportes> listarTodos(Pageable pageable)
* public Page<Reportes> buscarReportesPorBarrio(String barrio, Pageable pageable)

En **InformeAlertaService:**

* + - public Page<InformeAlerta> listarTodosPaginados(Pageable pageable)

3**. Nombre de la Estructura de Datos:** Map (HashMap)

* **Descripción:** Colección de pares clave-valor donde cada clave es única.
* **Tipo de Estructura:** Estructura de datos asociativa
* **Implementación en el Código:**

En **UserController:**

* public Map<String, Object> obtenerReporte(@PathVariable Long id) {  
   Map<String, Object> reporteDTO = new HashMap<>();

// ...  
}

En **AdminController:**

* public Map<String, Object> actualizarUser(@PathVariable Long id, @ModelAttribute User updatedUser) {  
  Map<String, Object> response = new HashMap<> ();  
  // ...  
  }

4**. Nombre de la Estructura de Datos:** Enum

* **Descripción:** Tipo de dato que consiste en un conjunto fijo de constantes.
* **Tipo de Estructura:** Estructura de datos enumerada
* **Implementación en el Código:**

En **InformeAlerta:**

* public enum EstadoInforme {  
  SIN\_RESOLVER, EN\_PROCESO, RESUELTO  
  }
* public enum ValoracionRiesgo {  
  BAJA, MEDIA, ALTA, CRITICA  
  }

En User:

* public enum UserTipo {  
  NORMAL, ADMIN, PRIVILEGED  
  }

5**. Nombre de la Estructura de Datos:** Arrays de Objetos

* **Descripción:** Estructura que almacena elementos del mismo tipo en posiciones contiguas de memoria.
* **Tipo de Estructura:** Estructura de datos lineal
* **Implementación en el Código:**
* public List<Object[]> contarReportesPorBarrioYContaminante(Long umbral)
* public List<Object[]> contaminantesMasReportados()

**Características Principales de las Estructuras Utilizadas**

1. **List/ArrayList**:
   * Acceso indexado O (1)
   * Inserción/eliminación al final O (1)
   * Inserción/eliminación en medio O(n)
   * Búsqueda O(n)
2. **Map/HashMap**:
   * Inserción O (1)
   * Búsqueda O (1)
   * Eliminación O (1)
   * No garantiza orden
3. **Page**:
   * Optimizado para grandes conjuntos de datos
   * Reduce el consumo de memoria
   * Mejora el rendimiento en consultas a base de datos
4. **Enum**:
   * Conjunto fijo de valores
   * Type-safe
   * Eficiente en memoria
5. **Arrays de Objetos**:
   * Acceso directo O (1)
   * Tamaño fijo
   * Memoria contigua

**OBSERVACIONES ADICIONALES:**