

1. Caso de Uso

El diagrama UML presentado detalla la estructura de un sistema que implica la interacción entre usuarios, gestión de objetos, control de un brazo robótico y la generación de reportes. A continuación, se describen detalladamente los casos de uso para cada uno de los componentes y relaciones en el diagrama:

1. InterfazUsuario

Usuario

- **nombre:** String
- **email:** String

Métodos:

- `iniciarSesion()` : Permite al usuario iniciar sesión en el sistema.
- `cerrarSesion()` : Permite al usuario cerrar sesión.
- `autenticacion()` : Verifica las credenciales del usuario.
- `modificarInformacion()` : Permite al usuario modificar su información personal.

PerfilUsuario

- **direccion:** String
- **horaInicio:** Date
- **horaFin:** Date

Métodos:

- `mostrarInformacion()` : Muestra la información del perfil del usuario.

GestionNotificaciones

Métodos:

- `mostrarNotificacion()` : Muestra notificaciones al usuario.
- `eliminarNotificacion()` : Permite al usuario eliminar notificaciones.

AdministracionUsuario

Métodos:

- `crearUsuario()` : Crea un nuevo usuario.
- `eliminarUsuario()` : Elimina un usuario existente.
- `modificarUsuario()` : Modifica la información de un usuario.

HistorialActividad

Métodos:

- `objetoRecoletado()` : Registra cuando un objeto ha sido recolectado.
- `movimientoRealizado()` : Registra los movimientos realizados por el usuario o el sistema.
- `tiempoConectado()` : Registra el tiempo de conexión del usuario.

Navegación (Interfaz)

Métodos:

- `verDatosRecolectados()` : Muestra los datos recolectados.
- `verVideo()` : Permite al usuario ver videos relacionados.
- `mensajesEstado()` : Muestra mensajes de estado del sistema.
- `verMateriales()` : Permite al usuario ver los materiales disponibles.

2. BrazoRobotico

Señal

- **remoto:** void
- **video:** void
- **resolucion:** void

Esp32cam

Métodos:

- `recibirSeñal()` : Recibe y procesa señales de la cámara ESP32.

ControlBrazoYCarro

Métodos:

- `moverBrazo()` : Controla el movimiento del brazo robótico.
- `abrirPinza()` : Abre la pinza del brazo robótico.
- `cerrarPinza()` : Cierra la pinza del brazo robótico.

3. Datos

Objeto

Métodos:

- `almacenarObjeto()` : Permite almacenar información sobre un objeto.
- `modificarObjeto()` : Permite modificar la información de un objeto.
- `mostrarInformacionObjeto()` : Muestra la información de un objeto.

Reporte

Métodos:

- `generarReporte()` : Genera un reporte sobre los objetos y actividades.
- `exportarReporte()` : Exporta el reporte generado a un formato adecuado.

InformacionObjeto

Métodos:

- `calcularPesoObjeto()` : Calcula el peso del objeto basado en una gráfica estadística.

4. ClasificacionObjeto

Este es un enumerado que clasifica los tipos de objetos en:

- PELIGROSO
- CORROSIVO
- INFLAMABLE
- ORGANICO
- INORGANICO
- QUIRURGICO

Relaciones Importantes:

- Un **Usuario** puede tener múltiples **PerfilUsuario**.
- **GestionNotificaciones** gestiona múltiples notificaciones por usuario.
- **HistorialActividad** registra actividades relacionadas con los usuarios.
- **ControlBrazoYCarro** controla el brazo robótico y la cámara ESP32.
- Los **Objetos** pueden ser clasificados y se almacenan con información específica.
- Los **Reportes** se generan y exportan para analizar datos sobre los objetos.

En conjunto, estos componentes colaboran para ofrecer una solución integral que combina gestión de usuarios, control robótico y análisis de datos en un sistema coherente y funcional.

Requerimientos del sistema separados por asignaturas

Probabilidad y Distribución en el Sistema de Gestión de Basura

Requerimientos del Sistema

El sistema de gestión de basura descrito en el diagrama UML tiene una serie de funcionalidades clave destinadas a optimizar y controlar la recolección y gestión de desechos. Entre estas funcionalidades, se incluyen requisitos específicos relacionados con la probabilidad y la distribución de los datos de recolección de basura. A continuación, se describen estos requisitos en detalle:

1. **Recolección y Visualización de Datos Diarios:**
 - **Cantidad de Basura Recogida al Día:** El sistema debe registrar la cantidad de basura recogida cada día. Este dato debe ser accesible para el usuario a través de la interfaz de usuario.

- **Cálculo de Estadísticas Diarias:** El sistema debe calcular y mostrar al usuario las siguientes estadísticas diarias:
 - **Media:** La media de la cantidad de basura recogida diariamente.
 - **Mediana:** La mediana de la cantidad de basura recogida diariamente.
 - **Moda:** La moda de la cantidad de basura recogida diariamente.
- 2. **Recolección y Visualización de Datos Mensuales:**
 - **Cantidad de Basura Recogida al Mes:** Además de los datos diarios, el sistema debe agregar y registrar la cantidad total de basura recogida cada mes.
 - **Cálculo de Estadísticas Mensuales:** Similar a los datos diarios, el sistema debe calcular y mostrar las siguientes estadísticas mensuales:
 - **Media:** La media de la cantidad de basura recogida mensualmente.
 - **Mediana:** La mediana de la cantidad de basura recogida mensualmente.
 - **Moda:** La moda de la cantidad de basura recogida mensualmente.
- 3. **Visualización Gráfica:**
 - **Histograma:** El sistema debe generar y mostrar un histograma de los datos de recolección de basura. Este histograma debe representar de forma visual la distribución de la cantidad de basura recogida, permitiendo al usuario entender mejor las tendencias y variaciones en la recolección de basura.

Implementación en el Sistema

Para cumplir con estos requisitos, el sistema deberá contar con los siguientes componentes y métodos:

- **Clase ~~Historial~~Actividad:** Esta clase será responsable de registrar y gestionar los datos de recolección de basura. Métodos relevantes podrían incluir:
 - `+objetoRecolectado() : void`: Para registrar cada instancia de recolección de basura.
 - `+mostrarInformacion() : String`: Para calcular y devolver las estadísticas requeridas (media, mediana, moda).
- **Clase ~~Reporte~~:** Esta clase generará los informes y visualizaciones gráficas de los datos.
 - `+generarReporte() : String`: Para generar un informe textual con las estadísticas diarias y mensuales.
 - `+exportarReporte() : String`: Para exportar el informe en un formato adecuado.
 - `+calcularPesoObjeto(graficaEstadisticaPeso : void) : float`: Método que podría ser extendido para calcular y mostrar el histograma de los datos.

Beneficios para el Usuario

La inclusión de estas funcionalidades permitirá a los usuarios:

- **Monitoreo Diario y Mensual:** Los usuarios podrán realizar un seguimiento detallado de la cantidad de basura recogida diariamente y mensualmente, ayudándoles a identificar patrones y picos en la generación de desechos.
- **Análisis Estadístico:** Las estadísticas de media, mediana y moda proporcionarán una comprensión más profunda de los datos de recolección de basura, facilitando la toma de decisiones informadas.

- **Visualización Intuitiva:** El histograma ofrecerá una representación visual clara de la distribución de los datos, lo que es particularmente útil para identificar tendencias y anomalías en la recolección de basura.

En resumen, estos requisitos y funcionalidades mejorarán significativamente la capacidad del sistema para gestionar y analizar los datos de recolección de basura, proporcionando a los usuarios herramientas valiosas para la optimización y el control efectivo de la gestión de desechos.

Requerimientos del Sistema: Complejidad de Rendimiento de un Algoritmo

El sistema de gestión de basura necesita garantizar la eficiencia de los algoritmos utilizados para diversas operaciones. A continuación, se especifican los requisitos del sistema relacionados con el análisis y optimización de la complejidad de rendimiento de los algoritmos:

1. ****Medición de la Complejidad de los Algoritmos**:**

- El sistema debe ser capaz de medir y reportar la complejidad temporal (tiempo de ejecución) y espacial (uso de memoria) de los algoritmos utilizados para las tareas de recolección de datos, cálculo de estadísticas y generación de reportes.
- Se debe utilizar notación Big-O para expresar la complejidad en el peor caso, notación Omega para el mejor caso y notación Theta para casos promedio.

2. ****Optimización de Algoritmos**:**

- El sistema debe implementar algoritmos eficientes que minimicen tanto el tiempo de ejecución como el uso de memoria.
- Debe haber un proceso continuo de monitoreo y optimización para mejorar la eficiencia de los algoritmos a medida que aumenta la cantidad de datos o se identifican cuellos de botella.

3. ****Análisis de Casos de Complejidad**:**

- El sistema debe ser capaz de analizar y reportar la complejidad de los algoritmos en diferentes escenarios: mejor caso, peor caso y caso promedio.
- Este análisis debe incluir pruebas y evaluaciones periódicas para asegurar que los algoritmos mantengan un rendimiento óptimo bajo distintas condiciones.

4. ****Documentación y Reportes**:**

- El sistema debe generar documentación detallada y reportes sobre la complejidad de los algoritmos, incluyendo gráficos y estadísticas que faciliten la comprensión y evaluación del rendimiento por parte de los usuarios.

- Estos reportes deben estar disponibles para los administradores del sistema y los usuarios avanzados que necesiten entender el comportamiento y la eficiencia de los algoritmos implementados.

Implementar estos requisitos asegura que el sistema de gestión de basura no solo sea funcional, sino también eficiente y escalable, proporcionando a los usuarios una experiencia fluida y confiable.

Requerimientos del Sistema para el Proyecto de Brazo Robótico

1. Segmentación de Mercado

a) Ubicación Geográfica

- El sistema debe ser capaz de identificar y segmentar a los usuarios en la ciudad de Loja, Ecuador.

b) Segmentación Social

- **Género:** El sistema debe almacenar y procesar datos de usuarios de ambos géneros.
- **Edad:** El sistema debe gestionar información de usuarios entre 18 y 65 años.
- **Estado Civil:** El sistema debe soportar la segmentación basada en el estado civil (solteros, casados, divorciados, viudos).
- **Nivel de Estudio:** El sistema debe registrar y utilizar datos sobre el nivel de estudio de los usuarios (desde estudiantes de secundaria hasta posgraduados).

2. Definición de Objetivos del Estudio de Mercado

a) Determinar el interés del público objetivo por el producto desarrollado.

- El sistema debe incluir funcionalidades para recopilar y analizar datos sobre el interés de los usuarios en el uso del brazo robótico.

b) Establecer el objetivo general del estudio de mercado.

- El sistema debe ser capaz de realizar estudios de mercado para evaluar la viabilidad comercial y la aceptación del brazo robótico en Loja.

c) Definir tres objetivos específicos:

- **Evaluar el nivel de aceptación de la aplicación entre diferentes segmentos demográficos.**

- El sistema debe segmentar y evaluar datos demográficos para medir la aceptación del producto.
- **Identificar las características más valoradas por los usuarios potenciales.**
 - El sistema debe permitir la identificación y análisis de las características del producto que son más valoradas.
- **Determinar las mejoras necesarias para aumentar la satisfacción del usuario.**
 - El sistema debe recopilar y analizar sugerencias de los usuarios para implementar mejoras en el producto.

3. Encuesta

a) Medir el interés en usar la aplicación.

- El sistema debe contar con módulos para diseñar y distribuir encuestas que midan el interés de los usuarios en el brazo robótico.

b) Determinar si los encuestados han usado este tipo de productos anteriormente.

- El sistema debe incluir preguntas en las encuestas que indaguen sobre experiencias previas con tecnologías similares.

c) Recoger expectativas sobre el producto.

- El sistema debe permitir la recopilación de datos sobre las expectativas de los usuarios respecto al brazo robótico.

4. Aplicación de Encuestas

a) Describir el método y el proceso de distribución de las encuestas.

- El sistema debe soportar múltiples métodos de distribución de encuestas (en línea, en persona) y gestionar su proceso de distribución de manera eficiente.

b) Detallar la población objetivo y el tamaño de la muestra.

- El sistema debe definir y manejar la población objetivo (residentes de Loja entre 18 y 65 años) y garantizar que se alcance un tamaño de muestra representativo (mínimo 500 personas).

5. Análisis de Resultados

a) Utilizar gráficos estadísticos para representar los datos recolectados.

- El sistema debe generar gráficos estadísticos (barras, tortas, histogramas) para visualizar los resultados de las encuestas.

b) Interpretar los resultados obtenidos en las encuestas.

- El sistema debe proporcionar herramientas de análisis para interpretar los resultados de las encuestas y detectar patrones y tendencias.

6. Informe Final

a) Sectores de mercado:

- **Identificar los sectores que muestran mayor interés en el producto desarrollado, basándose en los resultados de la encuesta.**
 - El sistema debe identificar y reportar los sectores del mercado con mayor interés en el brazo robótico.

b) Análisis de competencias:

- **Evaluar la competencia existente y su impacto potencial en el mercado utilizando el formulario de estudio de mercado de competidores.**
 - El sistema debe evaluar la competencia y su impacto, utilizando datos de mercado y análisis comparativo.

c) Oportunidades de negocio:

- **Identificar y describir las oportunidades de negocio basadas en la información recolectada y cómo aprovecharlas.**
 - El sistema debe identificar y describir oportunidades de negocio, y proporcionar recomendaciones sobre cómo aprovecharlas basándose en los datos recolectados.

Estos requerimientos del sistema aseguran que el proyecto del brazo robótico esté bien alineado con las necesidades y expectativas del mercado objetivo, y que se tomen decisiones informadas para mejorar y comercializar el producto de manera efectiva.