

Problemática y ODS

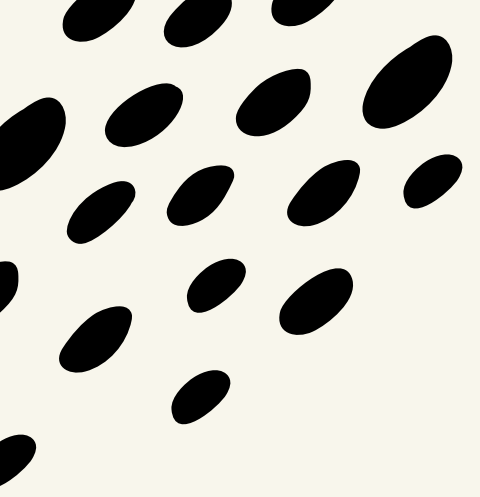
La **ausencia** de **rutas óptimas** para la recolección de basura es un **desafío** crucial en la **gestión de residuos** en el Perú, lo cual compromete la eficiencia del sistema y contribuye al deterioro del entorno urbano

¿Cuáles son su metas?

1. Facilitar el trabajo de los **recogedores de desechos**
2. **Conservar la limpieza** de los distritos donde se encuentre implementado nuestra solución
3. Unir al país en el camino de las ODS
4. **Reducir la aglomeración de desperdicios en los contenedores**



Figura: Metas e indicadores de ODS 11

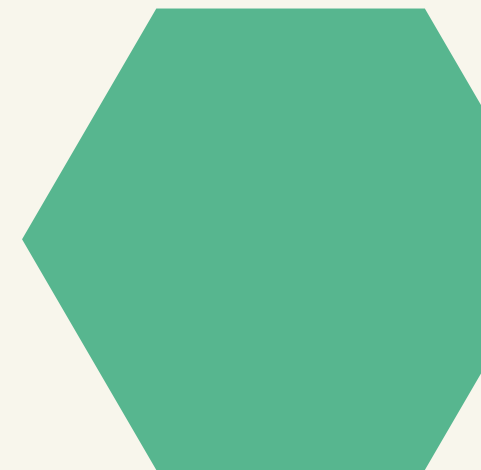
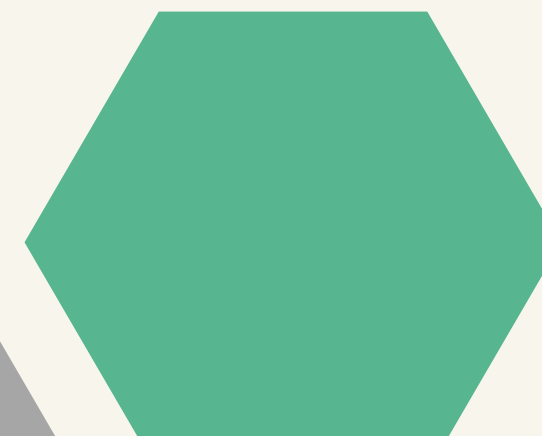
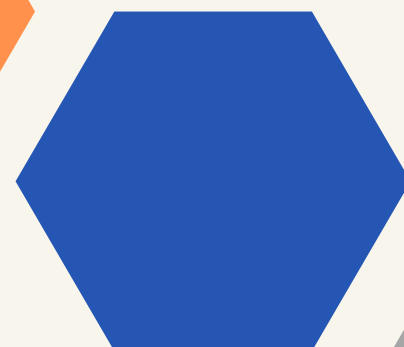
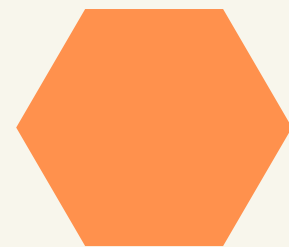


PROYECTO EcoRouteX

Integrantes:

- Archiño Cerna, Noemi Salomina
- Aybar Escobar, Edithson Ricardo
- Colla Cervantes, Marelly Massiel
- Quezada Marceliano, Gian Carlos
- Salazar Cobian, Arny Eliu

Grupo 4



Ineficiencia en rutas de recolección

Actualmente, los camiones compactadores **no siguen una ruta específica** para la **recolección de residuos**. Estos camiones simplemente reciben órdenes sobre los puntos que deben visitar, pero no se les indica el **camino más eficiente** para llegar a cada destino. Esto provoca que los camiones deambulen, **incrementando el tiempo de operación, el consumo de combustible** y las **emisiones contaminantes**. Además, el tráfico pesado y continuo generado por estos camiones **deteriora las vías públicas**, especialmente en zonas residenciales y calles con infraestructura débil.



Figura: Problemas viales en la ciudad de Lima, FRAPIAL 2021

Lista de requerimientos

Categoría	Requerimientos adaptados al proyecto de recolección de basura
Integración	El sistema debe integrarse con al menos 3 tecnologías: GPS, sensores de llenado y mapas. La actualización de los datos debe ocurrir en tiempo real, al menos cada 5 minutos.
Interfaces	Deben existir 2 interfaces principales: una para los operadores (administración) y otra para los conductores de camiones. Ambas deben ser 100% compatibles con dispositivos móviles.
Actualizaciones	El software debe recibir actualizaciones remotas al menos una vez al mes. Las actualizaciones deben realizarse en una ventana de mantenimiento que no dure más de 2 horas.
Hardware	Los dispositivos que optimizan las rutas y leen los sensores deben tener procesadores de al menos 4 núcleos y los sensores de llenado deben ser precisos, con un margen de error del $\pm 5\%$.
Testabilidad	Se deben hacer al menos 2 pruebas al mes usando un “gemelo digital” (una simulación) para probar la eficiencia de las rutas sin necesidad de implementarlas en la vida real.
Modos de operación	El sistema debe tener 3 modos de operación: emergencia (responde en menos de 15 minutos), automático (ajusta las rutas en tiempo real) y personalizado (usa rutas predefinidas).
Entorno de desarrollo	El desarrollo debe seguir un enfoque ágil con sprints, y cumplir con al menos el 80% de los estándares GIS y de manejo de datos en tiempo real.
Versionado	Se debe registrar una nueva versión del software cada vez que se hagan cambios importantes, con descripciones claras de al menos el 90% de los cambios realizados.
Selección de lenguajes	El software debe estar programado en al menos dos lenguajes: uno principal para el backend (por ejemplo, Python o Java) y otro para las interfaces de usuario (JavaScript).
Documentación	El 100% del código debe estar bien documentado según los estándares del proyecto, con especial atención a los algoritmos de optimización de rutas y la integración con los sensores.
Archivado de documentos	Toda la documentación debe almacenarse de forma organizada en una base de datos accesible. Las consultas a esta base deben responder en menos de 5 segundos.
Mantenibilidad del código	El código debe ser lo suficientemente flexible como para permitir la adición de nuevas áreas o tecnologías, con un tiempo de implementación de menos de una semana por cada cambio.
Requisitos de tiempo real	El sistema debe ajustar las rutas en tiempo real con una latencia máxima de 30 segundos cuando reciba alertas de llenado de contenedores o congestión.
Planificación	Los desarrollos deben organizarse en sprints de 2 semanas, con una meta de completar al menos el 90% de las tareas planeadas en cada sprint.

Tabla 1: Lista de requerimientos, creación propia.

Proyecto, ¿Qué estamos haciendo?

Modelar

Modelar un grafo del distrito en cuestion, apoyandonos de OpenStreetMap, que es una base de datos global



OpenStreetMap

Tomamos como ejemplo el distrito de San Borja

```
grafo = ox.graph_from_place(nombre_ciudad, network_type='drive')  
ox.plot_graph(grafo)
```

Proyecto, ¿Qué estamos haciendo?

Ubicar

Ubicar manualmente en el mapa cada contenedor relevante (usando sus coordenadas geográficas), considerándolos como relevante a los contenedores que cuenten con los sensores pertinentes

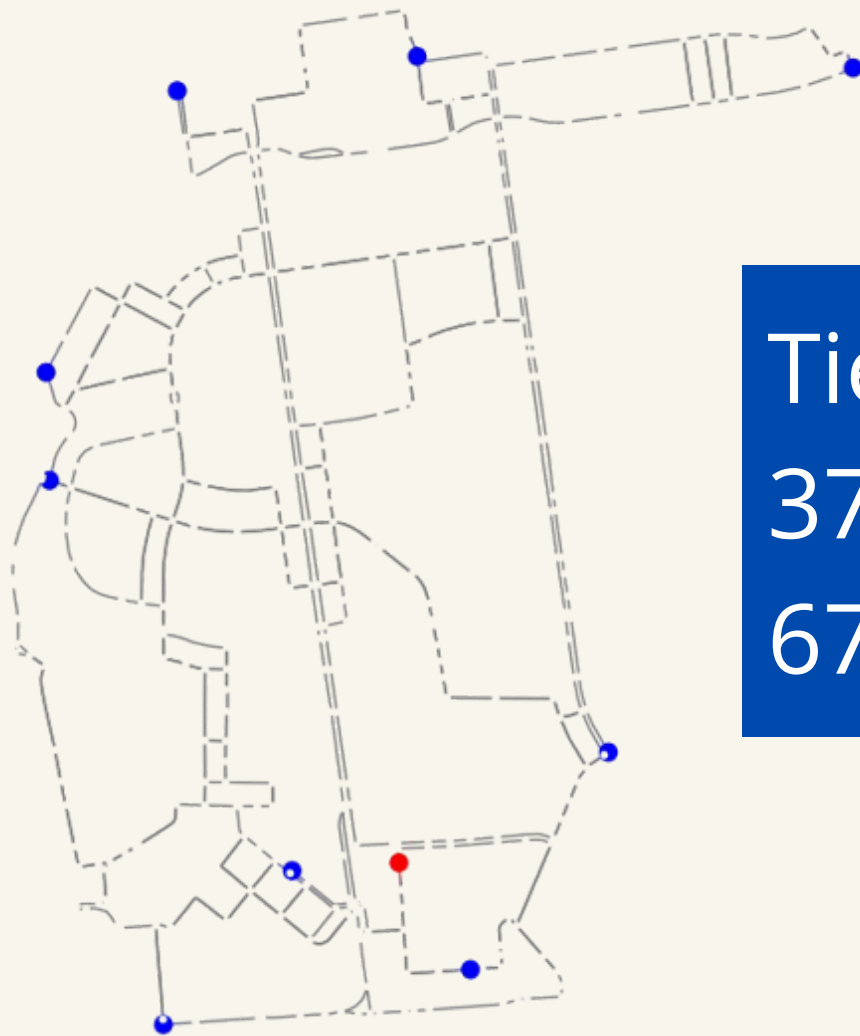


Tiene un total de:
1402 nodos y
3266 aristas

Proyecto, ¿Qué estamos haciendo?

Filtrar

Una vez modelado el grafo, se realiza un filtrado de los nodos no relevantes para así obtener un grafo reducido, “filtrado”



Tiene un total de:
375 nodos y
675 aristas

Asignar

Una vez modelado y filtrado nuestro grafo, se le asigna un peso a cada arista considerando (distancia, capacidad de cada tacho, tiempo de demora).

Proyecto, ¿Qué estamos haciendo?

Calcular iterativamente

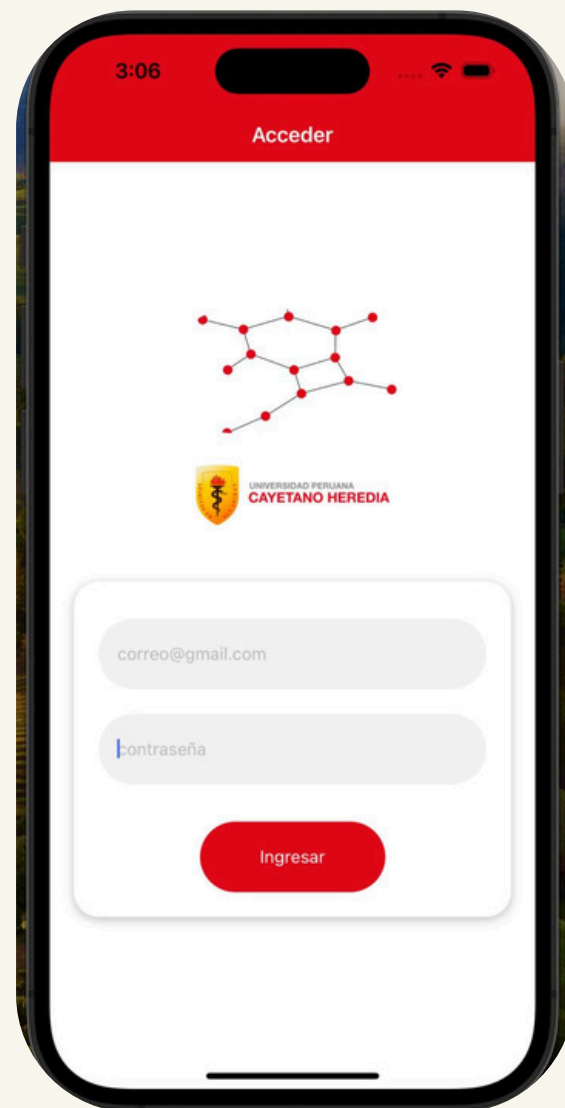
Gracias a los pesos de cada arista, nuestro programa calculará mediante un sistema de prioridades, cual vértice es el adecuado para su respectiva elección. Y así hasta que acabe con todos los vértices

Mostrar

El resultado podrá ser visualizado en una pagina web que será desarrollada para este fin, podremos ver la ruta que debemos seguir,

Proyecto, ¿Qué estamos haciendo?

Aplicativo móvil



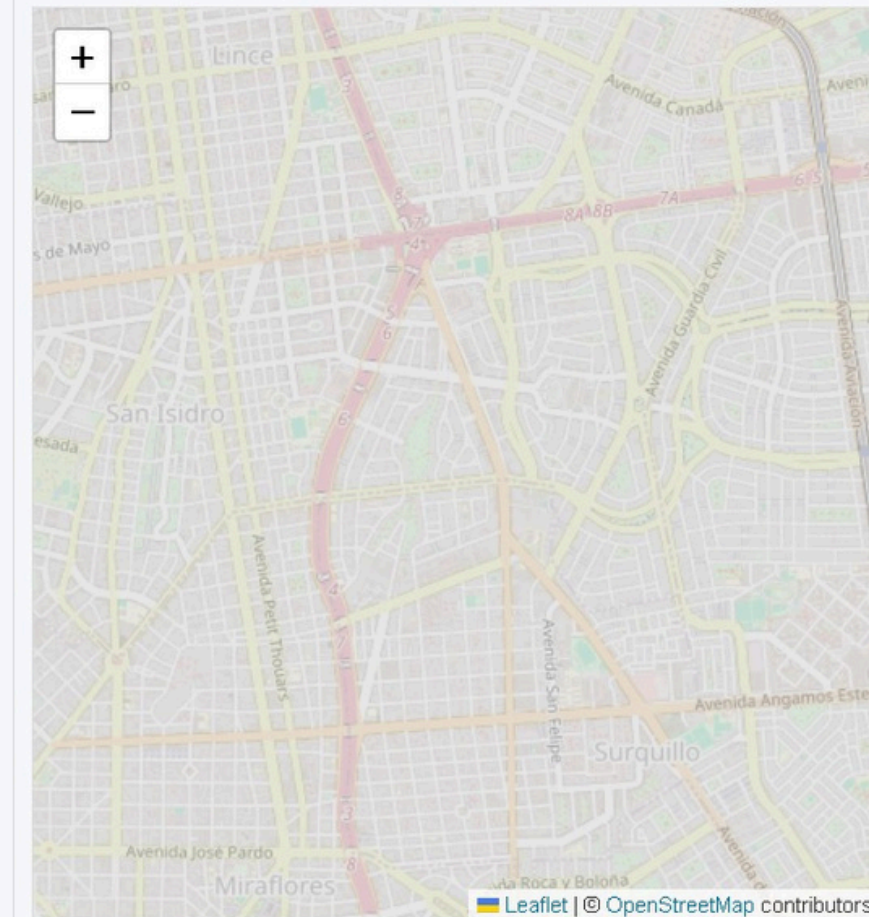
Aplicativo web

Ruta Óptima de Recojo de Residuos

Valores de los Potenciómetros

Potenciometro	Valor
Origen	0
potentiometer1	500
potentiometer2	50
potentiometer3	234
potentiometer4	456
potentiometer5	5678
potentiometer6	0
potentiometer7	45
potentiometer8	56
potentiometer9	890

Distancia Total: 0.06074097058068388 metros
Basura Total Recogida: 2231 unidades
Capacidad del Camión Recolector: 6000 unidades



Circuito electrónico – PCB

