



Instituto Tecnológico de Costa Rica

Escuela de ingeniería en Computación  
IC2101 Programación Orientada a Objetos

Proyecto n°1  
Primer Proyecto Programado POO IIS2025: Neo-Urbe

Estudiantes:  
Juan Esteban Ceballos Lopez  
Genesis Valeria Fajardo Díaz  
James Lucas Morgan Brenes  
Maripaz Quesada Leitón

## Introducción

El proyecto consiste en diseñar y programar una simulación en Java de una ciudad futurista llamada *Neo-Urbe*. En este lugar interactúan ciudadanos, robots asistentes, drones que patrullan, estaciones de recarga, edificios inteligentes y un Consejo de Inteligencia encargado de procesar los eventos que ocurren en la ciudad. El trabajo nos permitió poner en práctica varios de los conceptos principales de la Programación Orientada a Objetos (POO), como la creación de clases, el uso de relaciones entre ellas y la organización del código de forma modular.

La idea principal fue crear un sistema que muestre cómo estos elementos se comunican y trabajan entre sí. Por ejemplo, los ciudadanos pueden tener robots que realizan tareas, los drones vigilan diferentes zonas y reportan incidentes, y las estaciones de recarga permiten que tanto robots como drones sigan funcionando. Con esta simulación no solo se busca cumplir con los requerimientos del curso, sino también aprender a aplicar principios de diseño como la composición, la cohesión y la separación de responsabilidades, que son muy útiles al momento de programar proyectos más grandes.

Además de la implementación, esta memoria incluye los requerimientos planteados, el diseño UML con las clases y relaciones del sistema, la explicación de la solución en Java y la bitácora de trabajo realizada entre el 09/09/2025 y el 05/10/2025. De esta forma, el documento refleja tanto la parte técnica como el proceso de construcción del proyecto, mostrando cómo se aplicaron los conocimientos aprendidos en clase para lograr un sistema que simula una ciudad con elementos futuristas.

## Índice

1. Introducción.....	
2. Índice.....	
3. Requerimientos funcionales (RF).....	
4. Propuesta de metodología de trabajo.....	
5. Cronograma de trabajo.....	
6. Diagrama de clases UML.....	
7. Justificación de diagrama.....	
8. Análisis de resultados.....	
9. Bitácora de trabajo.....	
10. Aspectos relevantes y lecciones aprendidas.....	
11. Bitácora de trabajo.....	

## Requerimientos funcionales (RF)

RFN	Descripción breve
RF-1: Gestión de Edificios Inteligentes	Crear entre 3 y 10 edificios inteligentes, registrando su id, nombre, ubicación y capacidad máxima de residentes.
RF-2: Gestión de Estaciones de Energía	Permitir al Administrador crear entre 5 y 8 estaciones de energía, con id, descripción, ubicación, capacidad máxima de atención simultánea y estado
RF-3: Definir eventos predeterminados	Mantener lista de eventos anómalos pre-registrados (incendio, colisión, humo, gases, etc.).
RF-4: Gestionar acciones asociadas	CRUD de acciones (bomberos, tránsito, 911, ambulancias) y asociación evento con acciones.
RF-5: Reglas globales	Definir reglas configurables (umbral batería para drones y robots; activar/inactivar reglas).
RF-6: Gestión de Ciudadanos (CRUD)	Permitir al perfil Operador realizar operaciones CRUD sobre ciudadanos, asegurando que cada ciudadano se asigne a un edificio con capacidad disponible.
RF-7: Gestión de Robots Asistentes (CRUD)	Permitir al perfil Operador crear y administrar robots asistentes, con un procesador único, nivel inicial de batería (20–100%), y un conjunto aleatorio de tareas predeterminadas que no pueden ser modificadas
RF-8: Asignar robot a ciudadano	Servicio para asignar un robot aleatorio a un ciudadano por id.
RF-9: Generación de drones	Generar conjunto de drones automáticamente; cantidad $\leq 2 \times$ edificios; no dejar edificio sin vigilancia; no permitir modificar/eliminar la cantidad generada
RF-10: Propiedades de drones	Cada dron tiene procesador, Id único, horas de vuelo aleatorias 1–4, batería = $25\% \times$ horas.
RF-11: Simulación de Actividades en Neo-Urbe	Simulaciones en las que ciudadanos solicitan tareas a robots (con consumo energético), drones patrullan, estaciones registran recargas y el Consejo de Inteligencia atiende eventos detectados por drones, guardando en cada caso los registros de fecha, hora y resultados.
RF-12: Registro de tareas del	Registrar historial de tareas por robot (tarea, fecha y hora) cuando un ciudadano solicita acción.

robot	
RF-13: Patrullaje de drones	Un conjunto aleatorio de drones activos patrulla 1 hora (consume 25%); si patrullaje dejaría batería < umbral, ir a recarga y no patrullar.
RF-14: Gestión de Acciones Asociadas a Eventos	El Administrador debe poder asociar una o más acciones a cada evento. No puede existir un evento sin acción asociada.
RF-15: Gestión de Reglas del Entorno	El Administrador debe poder activar/inactivar reglas generales del entorno.
RF-16: Gestión de Drones de Vigilancia	El Operador debe poder generar drones automáticamente, asegurando vigilancia para todos los edificios
RF-17: Registro de Recargas en Estaciones	Las estaciones deben registrar cada recarga realizada a robots o drones, incluyendo fecha, hora y dispositivo atendido
RF-18: Consejo de Inteligencia (acciones)	El Consejo recibe reportes de drones y ejecuta/registrar acciones asociadas al evento (con timestamp, drone, acción).
RF-19: Dashboard (perfil General)	Mostrar dashboard con dimensiones: Energía, Infraestructura, Seguridad, Bienestar; incluir KPIs, gráficos y al menos 1 alerta por dimensión (el equipo elige).
RF-20: Validaciones al crear ciudadanos/robots	Validar unicidad de ids; validar capacidad de edificios al crear ciudadano; validar capacidad de estación al recargar.
RF-21: Módulos por perfil (UI)	Soportar módulos/funcionalidades por perfil: Administrador (configuración), Operador (CRUD y gestión) y General (dashboard).
RF-22: Simulación de Actividades	Debe poder ejecutarse una simulación que involucre tareas de robots, patrullaje de drones, recargas, detección de eventos y acciones del Consejo.

<p>RF-23: Análisis de Ocupación y Cobertura</p>	<p>El sistema debe analizar la ocupación de edificios y la cobertura de robots por ciudadano, generando alertas si hay sobreasignación o falta de cobertura.</p>
<p>RF-24: Hacer las interfases para la aplicación</p>	<p>Se deben hacer las interfases necesarias para mostrar los datos, y generar edificios, ciudadanos, drones y estaciones para la simulacion</p>

## Propuesta de metodología de trabajo

Para el desarrollo de este proyecto no seguimos una metodología formal, sino que adoptamos una forma de trabajo sencilla y práctica que se ajustó a nuestras necesidades como grupo. Principalmente nos organizamos mediante reuniones virtuales en Discord, donde conversábamos sobre los requerimientos y decidíamos qué parte del proyecto debía hacerse primero. En estas reuniones también repartíamos las tareas según las fortalezas de cada integrante, de manera que todos aportáramos en el avance del trabajo.

Además de Discord, utilizamos WhatsApp para mantenernos en contacto de forma rápida durante la semana. Ahí compartíamos dudas, recordatorios y pequeños avances. Gracias a este canal de comunicación constante, pudimos resolver problemas más rápido y coordinarnos sin necesidad de reunirnos siempre de manera formal.

En cuanto a la organización del trabajo, la estrategia que seguimos fue avanzar por partes: primero nos enfocamos en diseñar las clases principales, luego en implementar la lógica básica y finalmente en agregar detalles como la simulación, los registros y parte de la interfaz gráfica.

## Cronograma de trabajo

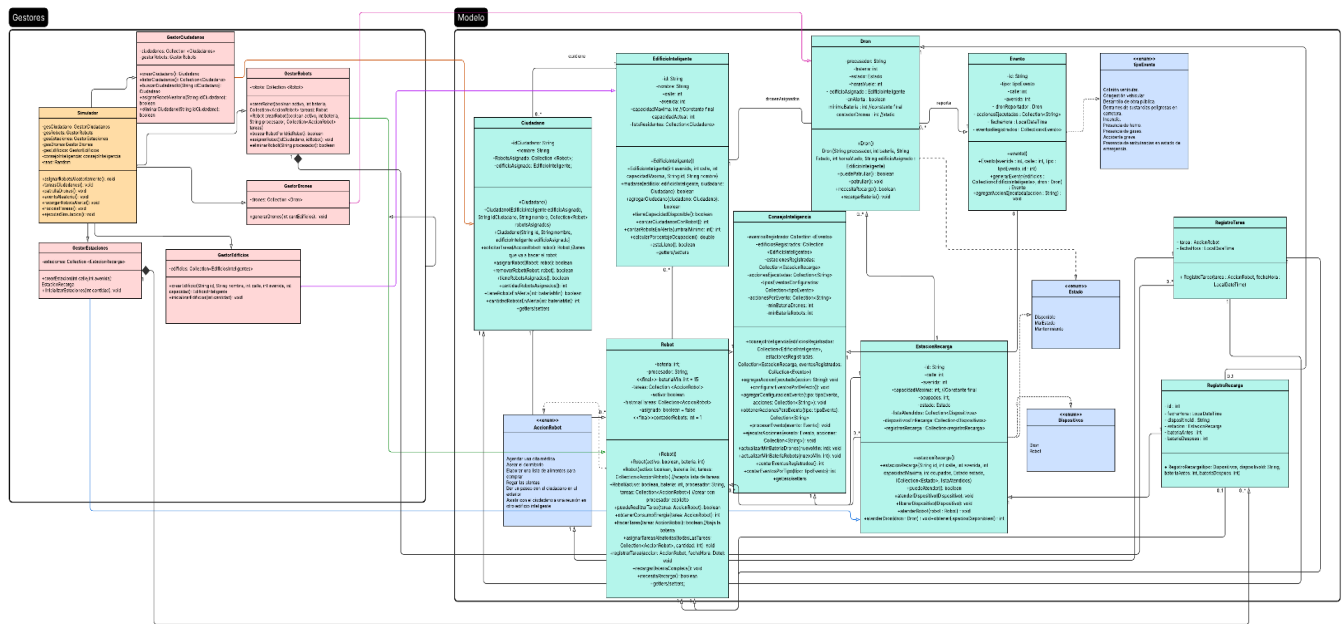
Semana	Fechas	Tareas planificadas	% completado (estimado)
1	09/09 – 12/09	Se identifican las clases principales, y se definen los atributos, métodos y relaciones entre clases	% 75
1-2	14/09 – 19/09	Empezar a diseñar en lenguaje UML las clases con sus atributos y métodos identificados anteriormente y agendar una reunión con el profesor encargado.  Robots, asignación, estaciones y recarga, RegistroRecarga/RegistroTarea (borrador).	%75

2-3	20/09 – 22/09	<p>Implementar los cambios en el UML sugeridos por el profesor.</p> <p>Empezar a crear el proyecto .Java</p> <p>Simulador, ConsejoInteligencia, generación de eventos, formateo de logs (fechas).</p>	%100
3	23/09 – 25/10	Se planea implementar las clases definidas en el UML, además de crear Enumeraciones	%100
3-4	26/09–29/09	Implementar los gestores necesarios para manejar las clases implementadas anteriormente y empezar hacer un borrador de Simulación.	%60
4	30/09–02/10	Terminar de realizar la Simulación de Neo-Urbe, agregando si es necesario cambios a los Gestores y empezar la interfaz gráfica del proyecto.	%75
4	01/10–05/10	Terminar Interfaz Gráfica del proyecto Neo-Urbe y verificar su correcto funcionamiento.	%100





# Diagrama de clases UML



## Justificación del diagrama

EdificioInteligente → Ciudadano

- Dirección: EdificioInteligente tiene listaResidentes: Collection<Ciudadano>; Ciudadano tiene edificioAsignado: EdificioInteligente.
- Multiplicidades habituales: EdificioInteligente 1 ← 0..\* Ciudadano y Ciudadano 0..1 → EdificioInteligente.
- Justificación: un edificio puede alojar cero o muchos ciudadanos (capacidad variable); un ciudadano puede no estar actualmente asignado (0..1) o estar asignado a exactamente un edificio. La relación refleja una agregación lógica (el edificio contiene residentes pero los ciudadanos existen independientemente del edificio). Además hay atributos en Edificio (capacidadMaxima, capacidadActual) que motivan la multiplicidad 0..\* y las operaciones tipo tieneCapacidadDisponible().

Ciudadano → Robot

- Dirección: Ciudadano contiene RobotsAsignado: Collection<Robot>; Robot tiene flag asignado: boolean (no se ve un atributo directo que apunte al ciudadano).
- Multiplicidades sugeridas: Ciudadano 0..1 ← 0..\* Robot (es decir, un ciudadano puede tener 0..\* robots asignados; un robot puede estar asignado a 0..1 ciudadano).
- Justificación: un robot solo puede ser responsable (asignado) a un ciudadano a la vez (si el diseño lo exige), por eso 0..1; en cambio un ciudadano puede tener múltiples robots. Esto facilita métodos como cantidadRobotsAsignados() y tieneRobotsAsignados().

#### Robot → GestorRobots / Simulador

- GestorRobots mantiene robots: Collection<Robot>.
- Tipo: agregación/gestor (no composición estricta).
- Justificación: el gestor actúa como contenedor y administrador; los robots pueden crearse, eliminarse o reasignarse sin que el gestor sea necesariamente el propietario exclusivo. Es coherente con métodos de búsqueda y creación (crearRobot, buscarRobotPorId).

#### Dron → EdificioInteligente → Evento

- Dron tiene edificioAsignado: EdificioInteligente y Evento tiene dronReportador: Dron.
- Multiplicidades: Edificio 1  $\leftarrow$  0..\* Dron (un edificio puede tener varios drones asignados) y Dron 1  $\leftarrow$  0..\* Evento (un dron puede reportar muchos eventos; cada evento lo reporta un dron específico: Evento 1  $\rightarrow$  Dron).
- Justificación: los drones patrullan y reportan múltiples eventos a través del tiempo; cada evento es generado por un dron concreto (necesitamos saber qué dron lo reportó para trazabilidad). El vínculo con el edificio como asignación asegura que el dron tenga un “área base” o responsable institucional.

#### ConsejoInteligencia → Evento / Edificios / Estaciones

- ConsejoInteligencia mantiene colecciones: eventosRegistrado, edificiosRegistrados, estacionesRegistradas.
- Multiplicidades: ConsejoInteligencia 1  $\rightarrow$  0..\* Evento; ConsejoInteligencia 1  $\rightarrow$  0..\* Edificio; ConsejoInteligencia 1  $\rightarrow$  0..\* EstacionRecarga.
- Justificación: el consejo centraliza información, procesa eventos y ejecuta acciones: necesita referenciar múltiples entidades. La relación es de agregación/servicio (no composición): los edificios y estaciones existen por separado.

#### EstacionRecarga → Dispositivos (Robot/Dron) → RegistroRecarga

- EstacionRecarga tiene listaAtendidos: Collection<Dispositivos> y registrosRecarga: Collection<RegistroRecarga>. RegistroRecarga incluye estacion : EstacionRecarga y dispositivoid.
- Multiplicidades: EstacionRecarga 1  $\leftarrow$  0..\* RegistroRecarga y EstacionRecarga 0..\*  $\leftrightarrow$  0..\* Dispositivo (registro histórico: many-to-many conceptualmente, pero modelado mediante RegistroRecarga que actúa como entidad asociativa).
- Justificación: las estaciones atienden varios dispositivos de forma concurrente o secuencial; cada recarga genera un registro (1 recarga  $\rightarrow$  1 registro; estación 1  $\rightarrow$  \* registros). Usar RegistroRecarga como entidad permite mantener historial y consultas posteriores.


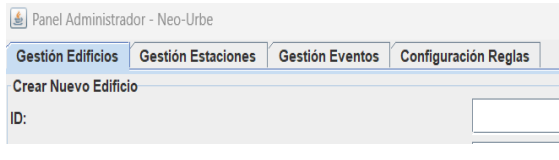
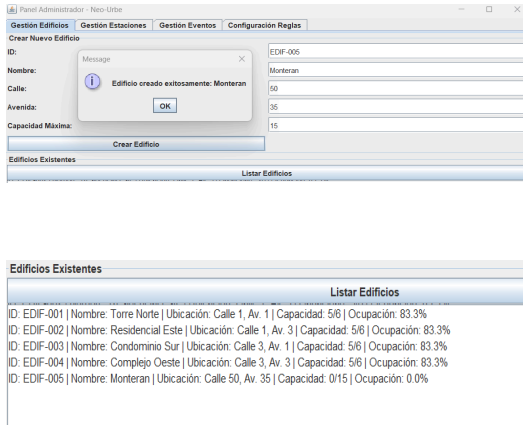
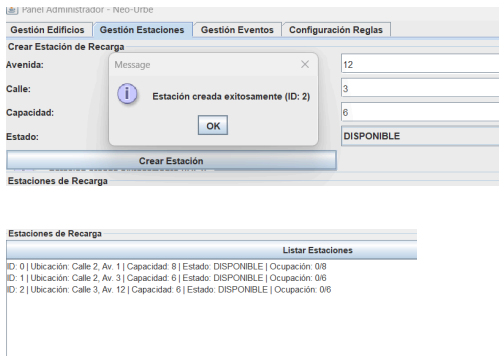
RegistroTarea → Robot → AccionRobot

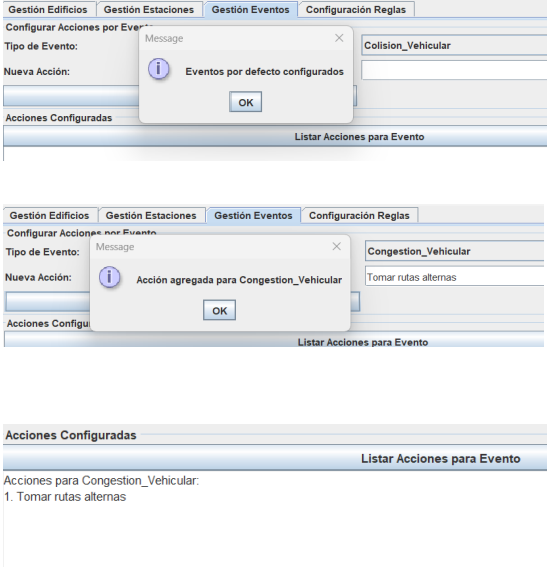
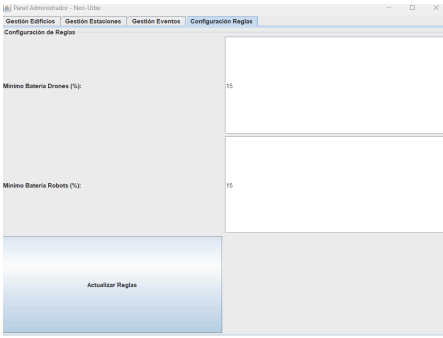
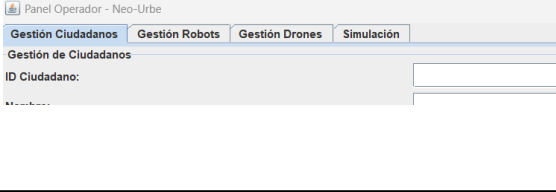
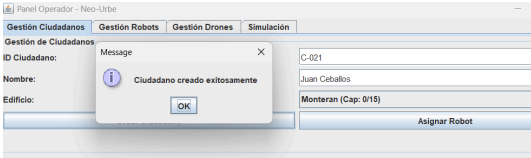
- RegistroTarea(tarea: AccionRobot, fechaHora) y Robot tiene historialTareas: Collection<AccionRobot>.
- Multiplicidad: Robot 1 ←— 0..\* RegistroTarea (un robot puede tener muchos registros de tareas).
- Justificación: mantener trazabilidad de tareas realizadas por cada robot; facilita métricas y auditoría.

Gestores (Edificios, Drones, Estaciones, Ciudadanos) → Simulador

- Simulador tiene referencias a los gestores (gesCiudadano, gesRobots, gesEstaciones, etc.).
- Tipo: dependencia/uso.
- Justificación: el simulador orquesta acciones y usa los gestores para crear, asignar y operar entidades. Es una relación de control/no-propiedad — el simulador no “posee” permanentemente las entidades.

## Análisis de resultados

Actividades y Tareas	Porcentaje	Prueba
Cuando se ejecute el proyecto mostrar una pantalla con los diferentes perfiles a manejar	100%	
Ingresar a perfil administrador y demostrar sus diferentes gestaciones que se puedan realizar	100%	
En gestor de edificios añadir correctamente un edificio y mostrar la lista de estos mismos creados	100%	
En gestor de estaciones añadir correctamente una estación y mostrar la lista de estas mismas creadas	100%	

<p>Configurar un evento por defecto o seleccionar un tipo en específico y agregar una acción.</p> <p>Mostrar lista de acciones para evento</p>	100%	 <p>Acciones Configuradas</p> <p>Listar Acciones para Evento</p> <p>Acciones para Congestion_Vehicular:</p> <ol style="list-style-type: none"> <li>1. Tomar rutas alternas</li> </ol>
<p>Configurar las reglas de las baterías de los drones y los robots, además de actualizarlas</p>	100%	
<p>Mostrar los diferentes gestores en perfil operador además de la simulación</p>	100%	
<p>Crear correctamente un ciudadano agregándole a un edificio ya creado.</p> <p>Listar a los ciudadanos</p>	100%	

		<div><div><div>Listado de Ciudadanos</div><div><div>Actualizar Listado</div><div>Eliminar Ciudadano</div></div><div><div>ID: C-001   Nombre: Ana García   Edificio: Torre Norte   Robots: 1</div><div>ID: C-002   Nombre: Carlos López   Edificio: Torre Norte   Robots: 1</div><div>ID: C-003   Nombre: María Rodríguez   Edificio: Torre Norte   Robots: 1</div><div>ID: C-004   Nombre: José Martínez   Edificio: Torre Norte   Robots: 1</div><div>ID: C-005   Nombre: Laura Hernández   Edificio: Torre Norte   Robots: 1</div><div>ID: C-006   Nombre: Miguel Sánchez   Edificio: Residencial Este   Robots: 1</div><div>ID: C-007   Nombre: Elena González   Edificio: Residencial Este   Robots: 1</div><div>ID: C-008   Nombre: David Pérez   Edificio: Residencial Este   Robots: 1</div><div>ID: C-009   Nombre: Sofía Ramírez   Edificio: Residencial Este   Robots: 1</div><div>ID: C-010   Nombre: Jorge Torres   Edificio: Residencial Este   Robots: 1</div><div>ID: C-011   Nombre: Isabel Flores   Edificio: Condominio Sur   Robots: 1</div><div>ID: C-012   Nombre: Roberto Vargas   Edificio: Condominio Sur   Robots: 1</div><div>ID: C-013   Nombre: Carmen Ruiz   Edificio: Condominio Sur   Robots: 1</div><div>ID: C-014   Nombre: Fernando Díaz   Edificio: Condominio Sur   Robots: 1</div><div>ID: C-015   Nombre: Patricia Cruz   Edificio: Condominio Sur   Robots: 1</div><div>ID: C-016   Nombre: Ricardo Morales   Edificio: Complejo Oeste   Robots: 0</div><div>ID: C-017   Nombre: Adriana Reyes   Edificio: Complejo Oeste   Robots: 0</div><div>ID: C-018   Nombre: Oscar Ortega   Edificio: Complejo Oeste   Robots: 0</div><div>ID: C-019   Nombre: Daniela Silva   Edificio: Complejo Oeste   Robots: 0</div><div>ID: C-020   Nombre: Andrés Castro   Edificio: Complejo Oeste   Robots: 0</div><div>ID: C-021   Nombre: Juan Ceballos   Edificio: Monteran   Robots: 0</div></div></div></div>
<div>Crear un robot ya sea individual o en lote de 5 con su batería correspondiente</div> <div>Listar los robots creados además de un pequeño resumen</div>	100%	<div><div><div>Gestión CiudadanosGestión RobotsGestión DronesSimulación</div><div><div>Crear Robot</div><div>Batería Inicial (%):</div><div>Activo:</div><div><div>Message</div><div>Robot creado exitosamente: ROB-16</div><div>OK</div></div><div><div>100</div><div><input checked="" type="checkbox"/></div><div>Crear Lote (5 robots)</div></div></div><div><div>Robots Existentes</div><div></div></div></div></div> <div><div><div>Gestión CiudadanosGestión RobotsGestión DronesSimulación</div><div><div>Crear Robot</div><div>Batería Inicial (%):</div><div>Activo:</div><div><div>Message</div><div>Lote de 5 robots creado exitosamente</div><div>OK</div></div><div><div>80</div><div><input checked="" type="checkbox"/></div><div>Crear Lote (5 robots)</div></div></div><div><div>Robots Existentes</div><div></div></div><div><div>Listar Robots</div><div></div></div></div><div><div>Robots Existentes</div><div><div>Listar Robots</div><div><div>Procesador ROB-4   Batería: 97%   Activo: SI   Asignado: SI   Tareas: 4</div><div>Procesador ROB-5   Batería: 26%   Activo: SI   Asignado: SI   Tareas: 6</div><div>Procesador ROB-6   Batería: 65%   Activo: SI   Asignado: SI   Tareas: 4</div><div>Procesador ROB-7   Batería: 81%   Activo: SI   Asignado: SI   Tareas: 6</div><div>Procesador ROB-8   Batería: 45%   Activo: SI   Asignado: SI   Tareas: 2</div><div>Procesador ROB-9   Batería: 53%   Activo: SI   Asignado: SI   Tareas: 4</div><div>Procesador ROB-10   Batería: 25%   Activo: SI   Asignado: SI   Tareas: 1</div><div>Procesador ROB-11   Batería: 37%   Activo: SI   Asignado: SI   Tareas: 3</div><div>Procesador ROB-12   Batería: 62%   Activo: SI   Asignado: SI   Tareas: 4</div><div>Procesador ROB-13   Batería: 70%   Activo: SI   Asignado: SI   Tareas: 1</div><div>Procesador ROB-14   Batería: 44%   Activo: SI   Asignado: SI   Tareas: 1</div><div>Procesador ROB-15   Batería: 60%   Activo: SI   Asignado: SI   Tareas: 1</div><div>Procesador ROB-16   Batería: 100%   Activo: SI   Asignado: No   Tareas: 4</div><div>Procesador ROB-17   Batería: 54%   Activo: SI   Asignado: No   Tareas: 1</div><div>Procesador ROB-18   Batería: 52%   Activo: SI   Asignado: No   Tareas: 5</div><div>Procesador ROB-19   Batería: 50%   Activo: SI   Asignado: No   Tareas: 2</div><div>Procesador ROB-20   Batería: 50%   Activo: SI   Asignado: No   Tareas: 1</div><div>Procesador ROB-21   Batería: 49%   Activo: SI   Asignado: No   Tareas: 4</div></div></div><div><div>--- RESUMEN ---</div><div>Total robots: 21</div><div>Asignados: 15</div><div>Disponibles: 6</div></div></div></div>
<div>Asignar un robot a un ciudadano por medio de su ID</div>	100%	<div><div><div>Listado de Ciudadanos</div><div><div>Actualizar Listado</div><div>Eliminar Ciudadano</div></div><div><div><div>Input</div><div>ID del ciudadano:</div><div>C-021</div><div>OK</div><div>Cancel</div></div></div><div><div>C-001   Nombre: Ana García   Edificio: Torre Norte   Robots: 1</div><div>C-002   Nombre: Carlos López   Edificio: Torre Norte   Robots: 1</div><div>C-003   Nombre: María Rodríguez   Edificio: Torre Norte   Robots: 1</div><div>C-004   Nombre: José Martínez   Edificio: Torre Norte   Robots: 1</div><div>C-005   Nombre: Laura Hernández   Edificio: Torre Norte   Robots: 1</div><div>C-006   Nombre: Miguel Sánchez   Edificio: Residencial Este   Robots: 1</div><div>C-007   Nombre: Elena González   Edificio: Residencial Este   Robots: 1</div><div>C-008   Nombre: David Pérez   Edificio: Residencial Este   Robots: 1</div><div>C-009   Nombre: Sofía Ramírez   Edificio: Residencial Este   Robots: 1</div><div>C-010   Nombre: Jorge Torres   Edificio: Residencial Este   Robots: 1</div><div>C-011   Nombre: Isabel Flores   Edificio: Condominio Sur   Robots: 1</div><div>C-012   Nombre: Roberto Vargas   Edificio: Condominio Sur   Robots: 1</div><div>C-013   Nombre: Carmen Ruiz   Edificio: Condominio Sur   Robots: 1</div><div>C-014   Nombre: Fernando Díaz   Edificio: Condominio Sur   Robots: 1</div><div>C-015   Nombre: Patricia Cruz   Edificio: Condominio Sur   Robots: 1</div><div>C-016   Nombre: Ricardo Morales   Edificio: Complejo Oeste   Robots: 0</div><div>C-017   Nombre: Adriana Reyes   Edificio: Complejo Oeste   Robots: 0</div><div>C-018   Nombre: Oscar Ortega   Edificio: Complejo Oeste   Robots: 0</div><div>C-019   Nombre: Daniela Silva   Edificio: Complejo Oeste   Robots: 0</div><div>C-020   Nombre: Andrés Castro   Edificio: Complejo Oeste   Robots: 1</div><div>C-021   Nombre: Juan Ceballos   Edificio: Monteran   Robots: 1</div></div></div></div>

<p>Genera los drones y los enlista, además de asignarlos a los edificios creados</p>	<p>100%</p>	<div><div>Generación de Drones</div><div><div>Generar Drones Automáticamente</div><div>Lista Drones</div></div><div><div>Procesador DRON-1</div><div>Batería: 50%</div><div>Horas Vuelo: 2</div><div>Estado: DISPONIBLE</div><div>Edificio: Torre Norte</div><div>En Alerta: No</div></div><div>Procesador DRON-2</div><div>Batería: 75%</div><div>Horas Vuelo: 3</div><div>Estado: DISPONIBLE</div><div>Edificio: Residencial Este</div><div>En Alerta: No</div></div> <div>Procesador DRON-3</div> <div>Batería: 25%</div> <div>Horas Vuelo: 1</div> <div>Estado: DISPONIBLE</div> <div>Edificio: Condominio Sur</div> <div>En Alerta: No</div>
--	-------------	---

Procesador DRON-4

Batería: 25%

Horas Vuelo: 1

Estado: DISPONIBLE

Edificio: Complejo Oeste

En Alerta: No

Procesador DRON-5

Batería: 25%

Horas Vuelo: 1

Estado: DISPONIBLE

Edificio: Torre Norte

En Alerta: No

Procesador DRON-6

Batería: 100%

Horas Vuelo: 4

Estado: DISPONIBLE

Edificio: Residencial Este

En Alerta: No

Procesador DRON-7

Batería: 75%

Horas Vuelo: 3

Estado: DISPONIBLE

Edificio: Condominio Sur

En Alerta: No

Procesador DRON-8

Batería: 75%

Horas Vuelo: 3

Estado: DISPONIBLE

Edificio: Complejo Oeste

En Alerta: No

Procesador DRON-9

Batería: 100%

Horas Vuelo: 4

Estado: DISPONIBLE

Edificio: Torre Norte

En Alerta: No

Procesador DRON-10

Batería: 50%

Horas Vuelo: 2

Estado: DISPONIBLE

Edificio: Residencial Este

En Alerta: No

Procesador DRON-11

Batería: 75%

Horas Vuelo: 3

Estado: DISPONIBLE

Edificio: Condominio Sur

En Alerta: No

Procesador DRON-12

Batería: 50%

Horas Vuelo: 2

Estado: DISPONIBLE

Edificio: Complejo Oeste

En Alerta: No

Procesador DRON-13

Batería: 100%

Horas Vuelo: 4

Estado: DISPONIBLE

Edificio: Montañas

En Alerta: No

Procesador DRON-14

Batería: 50%

Horas Vuelo: 2

Estado: DISPONIBLE

Edificio: Torre Norte

En Alerta: No

Procesador DRON-15

Batería: 25%

Horas Vuelo: 1

Estado: DISPONIBLE

Edificio: Residencial Este

En Alerta: No

Procesador DRON-16

Batería: 100%

Horas Vuelo: 4

Estado: DISPONIBLE

Edificio: Condominio Sur

En Alerta: No

Procesador DRON-17

Batería: 25%

Horas Vuelo: 1

Estado: DISPONIBLE

Edificio: Complejo Oeste

En Alerta: No

Procesador DRON-18

Batería: 25%

Horas Vuelo: 1

Estado: DISPONIBLE

Edificio: Montañas

En Alerta: No

Drones generados exitosamente

OK





Enlace al JAVADOC

<https://proyectopoo1.netlify.app/>

## Aspectos relevantes y Lecciones aprendidas

Durante el desarrollo de este primer proyecto programado nos dimos cuenta de la importancia de organizarnos desde el principio. Al inicio nos costó un poco definir un plan de acción claro y eso hizo que los primeros avances fueran un poco más lentos. Con el tiempo comprendimos que distribuir las tareas desde temprano y tener un esquema de trabajo ayuda bastante a mantener el ritmo del grupo y a no atrasarse con el proyecto.

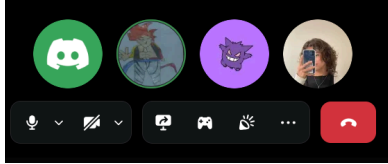
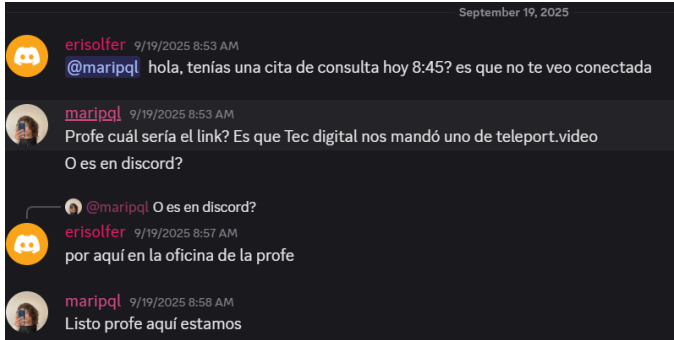
Otro aspecto que resultó muy importante fue la comunicación. Aunque desde el inicio usamos medios como WhatsApp y Discord, no siempre hablábamos con frecuencia y eso generaba ciertos vacíos en el trabajo. Conforme avanzamos, nos fuimos acostumbrando a reportar lo que íbamos haciendo y a plantear dudas de inmediato, lo cual fortaleció la coordinación y evitó que repitiéramos esfuerzos o que dejáramos detalles sin resolver.

En la parte técnica aprendimos a aplicar de forma práctica los conceptos de Programación Orientada a Objetos. No solo tuvimos que implementar clases y relaciones, sino también pensar en cómo se conectaban entre sí para simular una ciudad futurista, creo que esa fue una de las partes más complicadas pues se nos hacía un poco confuso este tema de relaciones entre clases. Esto nos llevó a reforzar conocimientos de composición, agregación y uso de colecciones en Java. Además, investigamos nuevas formas de manejar fechas, eventos y estructuras.

También fue valioso darnos cuenta de la importancia de la planificación del tiempo. En algunos momentos sentimos la presión de los plazos y entendimos que es mejor avanzar de manera constante, aunque sea poco a poco, en lugar de acumular demasiado trabajo para el final. Este aprendizaje nos ayudará en proyectos futuros, ya que ahora sabemos que con una mejor gestión del tiempo se puede trabajar de forma más tranquila y equitativa.

Por último, este proyecto nos dejó lecciones en el ámbito personal y de trabajo en equipo. Vimos lo útil que es apoyarnos entre compañeros, repartir responsabilidades de manera equitativa y confiar en las habilidades de cada uno. Además aprendimos a escuchar las ideas de los demás y a expresarnos con claridad, lo que hizo que el trabajo en grupo fuera más llevadero. En conclusión, este proyecto no solo nos dio conocimientos técnicos, sino también experiencias que podemos aplicar tanto en la universidad como en futuros entornos profesionales.

## Bitácora de trabajo

Fecha	Integrantes	Detalles
9 de septiembre 2025 Duración : 30 minutos	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se discutió y coordinó la metodología de trabajo con la que se hará el proyecto 
11 de septiembre 2025 Duración : 2 horas y 30 min	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se comenzó a trabajar en el diagrama de clases del trabajo en la aplicación web Lucidchart
15 de septiembre 2025 Duración : 1 hora	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se siguió trabajando en el diagrama de clases del proyecto.
19 de septiembre 2025	Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Una semana antes se había agendado una cita con la profe para revisar lo que llevábamos avanzado del proyecto. Un compañero no pudo estar presente pero luego le informamos sobre las correcciones y recomendaciones que nos hizo la profe. 
21 de septiembre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se siguió trabajando en el diagrama de acuerdo con las correcciones y recomendaciones de la profe.
23 de septiembre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se revisaron las correcciones hechas al diagrama UML y se empezó con la creación del proyecto en Java. Se distribuyeron responsabilidades para avanzar en paralelo con la implementación inicial de clases.

25 de septiembre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se avanzó en la codificación de las clases principales (Simulador, ConsejoInteligencia, y generación de eventos). Al mismo tiempo, se enfocó en el formateo de fechas y estructura de logs.
29 de septiembre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se continuó con la implementación de enumeraciones y gestores necesarios para el control de ciudadanos, robots y drones. Parte del equipo trabajó en pruebas iniciales del simulador mientras otros redactaban avances para la memoria del proyecto.
30 de septiembre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se realizó un ajuste en algunos gestores y se inició el diseño preliminar de la interfaz de usuario. Se comentaron posibles mejoras para la simulación y se dejaron tareas pendientes para el día siguiente.
01 de octubre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se continuó con la interfaz gráfica del proyecto y se probaron interacciones básicas. Algunos integrantes se dedicaron también a pulir detalles del documento, especialmente la metodología y justificación del diseño.
02 de octubre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se hicieron correcciones finales en la simulación y ajustes de la interfaz. Se dividieron las tareas entre revisión de código y redacción de secciones del documento, especialmente la introducción.
03 de Octubre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se probaron los gestores junto con el simulador para verificar su funcionamiento. Se corrigieron errores menores en las relaciones de clases y se trabajó en la documentación externa e interna.
04 de Octubre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se avanzó en la redacción de la bitácora y las secciones de lecciones aprendidas. Mientras tanto, se verificaron detalles finales en la interfaz gráfica para asegurar la correcta interacción con el simulador.
05 de Octubre 2025	Juan Esteban Ceballos Lopez Genesis Valeria Fajardo Díaz James Lucas Morgan Brenes Maripaz Quesada Leitón	Se hizo la revisión final del proyecto completo, tanto del código como del documento escrito. El equipo verificó el correcto funcionamiento de la simulación y la interfaz, y se ajustaron pequeños detalles de presentación para la entrega. 