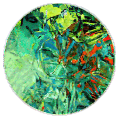


Índice

APLICACIÓN RESILIA	2
Ejercicio 1: Elabora un diagrama de casos de uso que represente gráficamente las principales funcionalidades de la aplicación RESILIA e identifique correctamente los actores principales del sistema.	3
Ejercicio 2: Rellena la tabla siguiente describiendo de manera clara el flujo de eventos asociados al camino básico del caso de uso reportar incidencia.	3
Ejercicio 3: Diseña un diagrama de clases básico que represente la estructura de datos necesaria para la aplicación. El diagrama debe reflejar al menos las siguientes clases: Usuario (y posibles roles como Administrador, Coordinador), PlanResiliencia., Recurso, Incidencia y Zona.	5
Anexo:	6
Requisitos no funcionales:	6
1. Requisitos de Rendimiento (Performance Requirements)	6
2. Requisitos de Usabilidad (Usability Requirements)	6
3. Requisitos de Seguridad (Security Requirements)	6
4. Requisitos de Fiabilidad y Disponibilidad (Reliability & Availability)	7
5. Requisitos de Mantenibilidad y Evolución (Maintainability Requirements)	7
6. Requisitos de Portabilidad y Compatibilidad (Portability Requirements)	7



APLICACIÓN RESILIA

Dado la siguiente información:

Se desea desarrollar una aplicación para una red de colaboración ciudadana, denominada **RESILIA**. La aplicación, que se llamará **RESILIA**, debe permitir gestionar de forma organizada la respuesta colectiva ante situaciones de apagón eléctrico o interrupciones prolongadas del suministro de energía. El objetivo de RESILIA es mejorar la resiliencia de barrios, asociaciones o comunidades locales frente a emergencias energéticas.

La funcionalidad principal de la aplicación consiste en la creación, difusión y actualización de planes de acción comunitarios ante un apagón. Cada barrio o zona geográfica podrá registrar su propio plan, que incluirá información relevante como puntos seguros de reunión, recursos disponibles (como generadores, linternas, botiquines, agua potable), responsables de área y protocolos de actuación. Además, los usuarios podrán apuntarse como voluntarios para tareas específicas (asistencia a personas vulnerables, reparto de suministros, comunicación de emergencias).

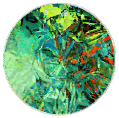
Cuando se crea un plan de resiliencia, se deben registrar los datos esenciales del mismo: nombre de la zona, responsable principal, recursos disponibles y rutas de comunicación seguras. Cada plan podrá ser modificado para actualizar información sobre la disponibilidad de recursos o cambios en los protocolos, y quedará archivado en el histórico una vez finalizada la emergencia para su evaluación y mejora posterior.

La aplicación permitirá también a los usuarios reportar incidentes durante el apagón, como zonas especialmente afectadas, personas que requieren ayuda urgente o fallos críticos en infraestructuras. Estos reportes incluirán ubicación, tipo de incidente y fotografías, y estarán disponibles para los coordinadores locales.

Los usuarios tendrán un perfil donde podrán gestionar su participación en planes de acción, inscribirse como voluntarios, consultar recursos de autoprotección y recibir alertas importantes. Para facilitar la comunicación en situaciones de emergencia, se incorporará un sistema de notificaciones por medio de datos móviles o SMS (en caso de corte de internet).

Además, RESILIA dispondrá de un módulo de administración desde el que se podrán gestionar zonas, usuarios, planes activos y reportes de incidencias, así como elaborar estadísticas que permitan analizar la respuesta de cada comunidad tras una emergencia.

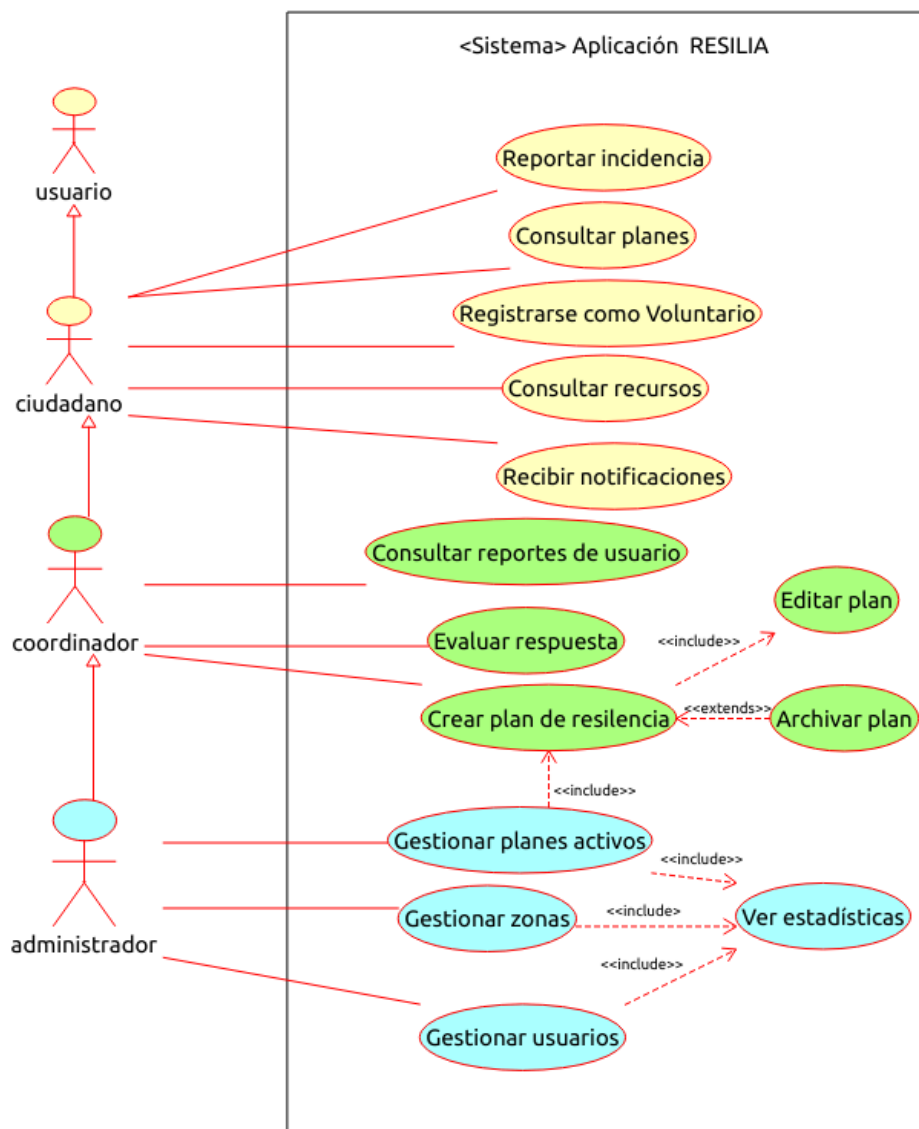
La aplicación deberá ser accesible desde dispositivos móviles y de escritorio, contar con una interfaz sencilla y estar optimizada para funcionar de forma ligera incluso con conectividad limitada. Se priorizará la autonomía energética y el bajo consumo de datos, para que pueda ser usada en condiciones adversas.

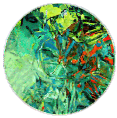


Ejercicio 1: Elabora un diagrama de casos de uso que represente gráficamente las principales funcionalidades de la aplicación **RESILIA** e identifique correctamente los actores principales del sistema.

Actores principales:

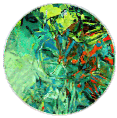
- Ciudadano
- Coordinador (usuario con permisos especiales)
- Administrador (usuario con acceso total a un sistema, pudiendo realizar cualquier acción, incluyendo la configuración, gestión de usuarios y roles, y acceso a todos los datos)





Ejercicio 2: Rellena la tabla siguiente describiendo de manera clara el **flujo de eventos** asociados al **camino básico** del caso de uso **reportar incidencia**.

	RF_2: Reportar incidencia.	
Descripción	El ciudadano reporta un incidente ocurrido durante el apagón.	
Actor(es)	ciudadano	
Precondición	El usuario está autenticado en el sistema.	
Postcondición	El reporte queda almacenado y es visible para los coordinadores locales.	
Flujo principal	Acción del actor	Respuesta del sistema
1.	Accede al formulario de reporte	Muestra formulario con campos: tipo, ubicación, foto
2	Completa los datos y envía	Valida la información
3	Confirma envío	Guarda el incidente y muestra mensaje de confirmación
Flujo alternativo		
2a	El usuario no tiene conexión	El sistema almacena el reporte localmente y lo envía cuando se restablece la conexión.
2b	Faltan datos obligatorios	El sistema informa del error y solicita completar los campos requeridos.

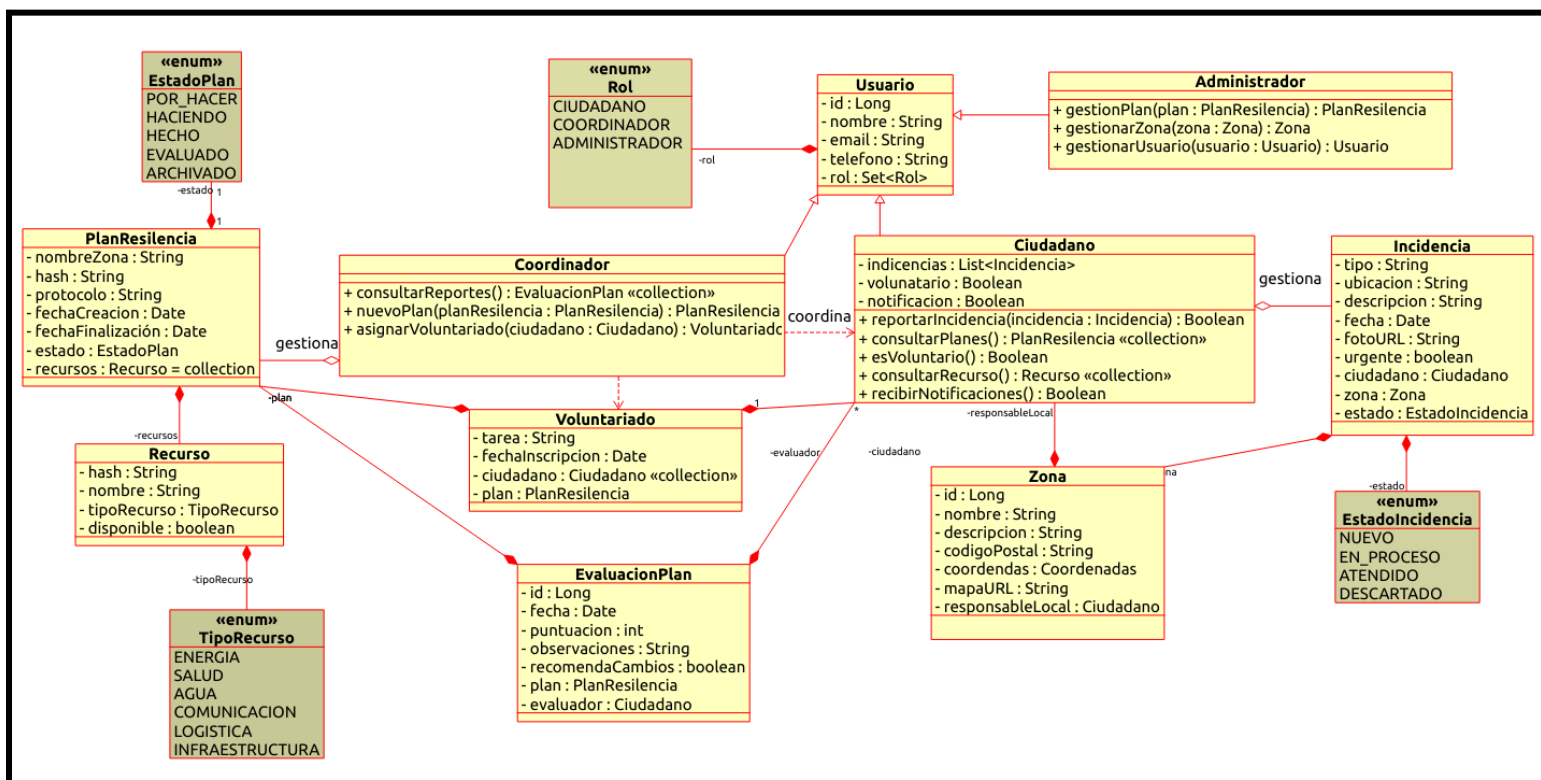
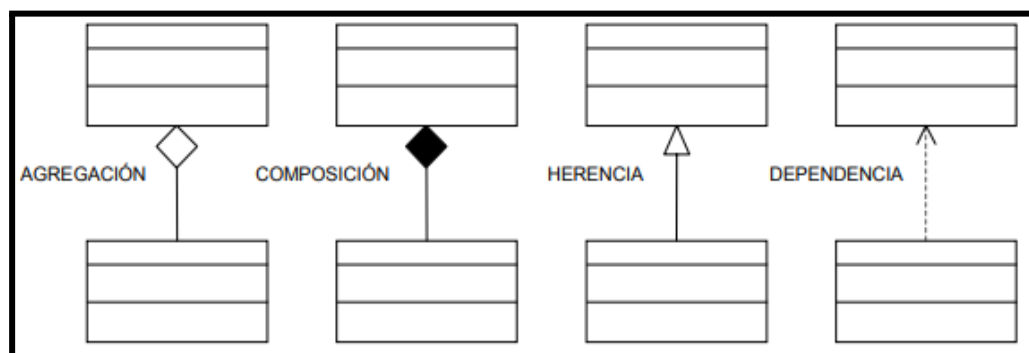


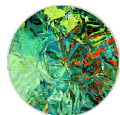
Ejercicio 3: Diseña un **diagrama de clases** básico que represente la estructura de datos necesaria para la aplicación.

El diagrama debe reflejar al menos las siguientes clases:

Usuario (y posibles roles como Administrador, Coordinador), PlanResiliencia., Recurso, Incidencia y Zona.

- Define los **atributos** principales en cada clase.
- Representa correctamente las **relaciones** y **multiplicidades** (por ejemplo, un plan tiene varios recursos; un usuario puede reportar varias incidencias).





Anexo:

Requisitos no funcionales:

Los requisitos no funcionales (RNF) son especificaciones que describen cómo debe comportarse un sistema más allá de lo que hace (funcionalidad). En otras palabras, no definen funcionalidades concretas, sino las **características de calidad** que debe cumplir el sistema para ser usable, eficiente, seguro, mantenible, etc.

1. Requisitos de Rendimiento (Performance Requirements)

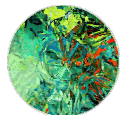
RNF-1.1	La aplicación debe garantizar tiempos de respuesta inferiores a 1 segundo para operaciones críticas (consultar planes, reportar incidencias) bajo condiciones de conectividad limitada.
RNF-1.2	El sistema debe ser capaz de soportar al menos 5.000 usuarios concurrentes sin degradación significativa del servicio.
RNF-1.3:	La transferencia de datos debe ser optimizada para un consumo inferior a 200 KB por operación básica en red 3G o inferior.

2. Requisitos de Usabilidad (Usability Requirements)

RNF-2.1	La interfaz debe cumplir con el estándar de accesibilidad WCAG 2.1 Nivel AA, permitiendo el acceso a personas con discapacidad.
RNF-2.2	La curva de aprendizaje para un usuario nuevo no debe superar los 10 minutos en tareas básicas (registrarse, consultar un plan, reportar una incidencia).
RNF-2.3	La aplicación debe ofrecer una experiencia de usuario uniforme en dispositivos móviles y de escritorio, utilizando un diseño adaptativo (responsive design).

3. Requisitos de Seguridad (Security Requirements)

RNF-3.1	Todo intercambio de datos deberá realizarse mediante conexiones cifradas bajo protocolo HTTPS con TLS 1.2 o superior.
RNF-3.2	Los datos personales (nombre, email, ubicación) deben almacenarse cifrados en reposo y en tránsito.
RNF-3.3	El sistema debe implementar un modelo de control de acceso basado en roles (RBAC) que limite las acciones según el tipo de usuario.
RNF-3.4	La autenticación debe requerir contraseñas fuertes y mecanismos de recuperación seguros.
RNF-3.5	La aplicación deberá estar protegida contra vulnerabilidades conocidas según el estándar OWASP Top 10.



4. Requisitos de Fiabilidad y Disponibilidad (Reliability & Availability)

RNF-4.1	La aplicación deberá alcanzar una disponibilidad mínima del 99.5% mensual, excluyendo mantenimiento programado.
RNF-4.2	El sistema debe garantizar persistencia de los datos durante cortes de red, permitiendo su sincronización posterior (modo offline).
RNF-4.3	Ante fallos críticos, el sistema deberá registrar eventos y permitir su trazabilidad mediante logs seguros.

5. Requisitos de Mantenibilidad y Evolución (Maintainability Requirements)

RNF-5.1	El código deberá seguir estándares de calidad (como Clean Code, SOLID y documentación Javadoc).
RNF-5.2	Las funcionalidades deben estar desacopladas mediante una arquitectura modular para facilitar su extensión y mantenimiento.
RNF-5.3	El sistema debe contar con una cobertura mínima del 80% de pruebas unitarias y de integración.

6. Requisitos de Portabilidad y Compatibilidad (Portability Requirements)

RNF-6.1	La aplicación debe ejecutarse correctamente en navegadores modernos (Chrome, Firefox, Edge, Safari) y sistemas móviles Android/iOS.
RNF-6.2:	La solución deberá estar desarrollada como una PWA (Progressive Web App) para habilitar su instalación en dispositivos sin necesidad de tiendas oficiales.
RNF-6.3:	La aplicación deberá permitir su despliegue en servidores con sistemas operativos Linux o Windows mediante contenedores (Docker).