

PROYECTO 1

RANDY ALEXANDER CAN AJUCHAN | 201801527
LUIS PEDRO PINEDA GONZÁLEZ | 201700995
PASCUAL DOMINGO DIEGO | 201403504
SERGIO RIVELINO PÉREZ RIVERA | 201603156

INVOLUCRADOS

- Vecino del municipio
- Empleado municipal
- Administrador
- Base de Datos

OBJETIVO DEL PROYECTO

Presentar una solución para que un vecino del municipio de Mixco pueda realizar una denuncia.

OBJETIVOS ESPECÍFICOS

- Que el vecino realice una denuncia desde su hogar.
- Que el vecino pueda denunciar de forma anónima.
- Que se de solución pronta al problema por parte de los empleados municipales.
- Reportar a las instituciones correspondientes.

EXPLICACIÓN DE PLATAFORMA

Lenguaje para el FrontEnd:

Para la realización de la página web se utilizó angular. Angular es un framework Javascript para la creación de páginas web SPA (Single Page Application). A su vez, que una web sea SPA quiere decir que cuando el usuario entra en la web se carga todo el contenido de todas las páginas a la vez. Esto quiere decir que la primera carga nada más entrar es más lenta pero luego los cambios entre páginas son instantáneos. Además, solo se utiliza un fichero con el código de la web por lo que no se requiere de muchas llamadas a servidor.

Otra de las características de Angular es que utiliza Typescript, un lenguaje que luego se transpila a Javascript y que sirve para tener tipado de variables, interfaces, inyección de dependencias y más cosas.

Para programar la aplicación móvil se utilizó IONIC, Ionic Framework es un SDK de frontend de código abierto para desarrollar aplicaciones híbridas basado en tecnologías web (HTML, CSS y JS). Es decir, un framework que nos permite desarrollar aplicaciones para iOS nativo, Android y la web, desde una única base de código. Su compatibilidad y, gracias a la implementación de Cordova e Ionic Native, hacen posible trabajar con componentes híbridos. Se integra con los principales frameworks de frontend, como Angular, React y Vue, aunque también se puede usar Vanilla JavaScript. Este framework fue creado en 2013 por Drifty Co. y hasta la llegada de React Native ha sido una de las tecnologías líderes para el desarrollo de aplicaciones móviles híbridas.

Lenguaje para el BackEnd:

El servidor se desarrollo en NodeJS, Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

La base de datos utilizada fue Mysql, MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

REQUERIMIENTOS DEL SISTEMA

- **Requerimientos funcionales**

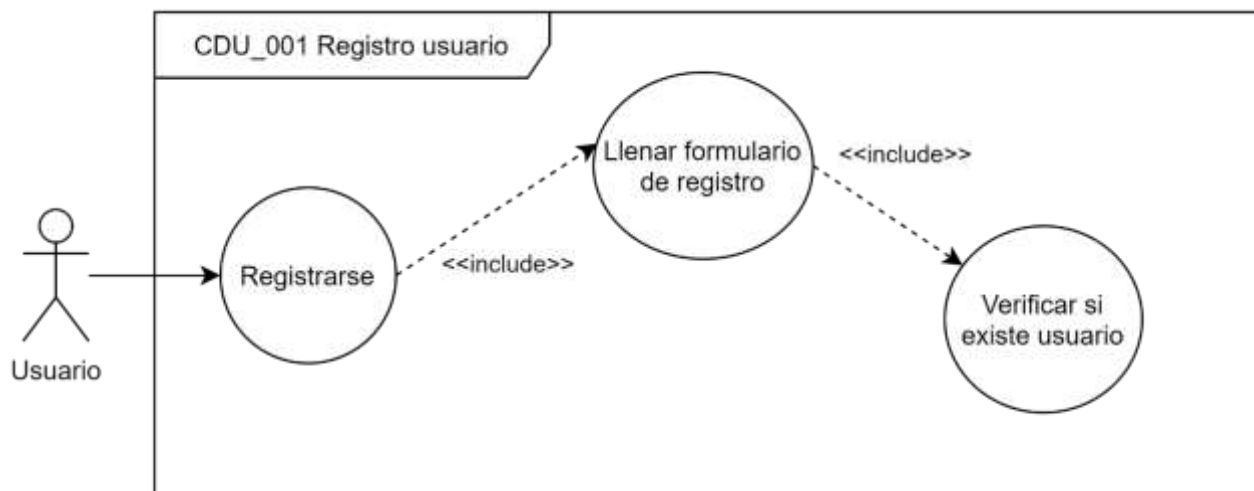
1. Registrar usuario vecino (app movil)
2. Registrar usuario municipal (app web)
3. Login (usuario vecino)
4. Login (usuario municipal)
5. Registrar denuncia en app móvil (usuario vecino)
6. Llenar formulario en app móvil (usuario vecino)
7. Enviar formulario en app móvil (usuario vecino)
8. Ingresar como usuario anónimo en app móvil (usuario vecino)
9. Reportar de forma anónima en app móvil (usuario vecino)
10. Responder notificación a través de mensaje (usuario empleado)
11. Ver denuncias en app web (usuario empleado)

- **Requerimientos no funcionales**

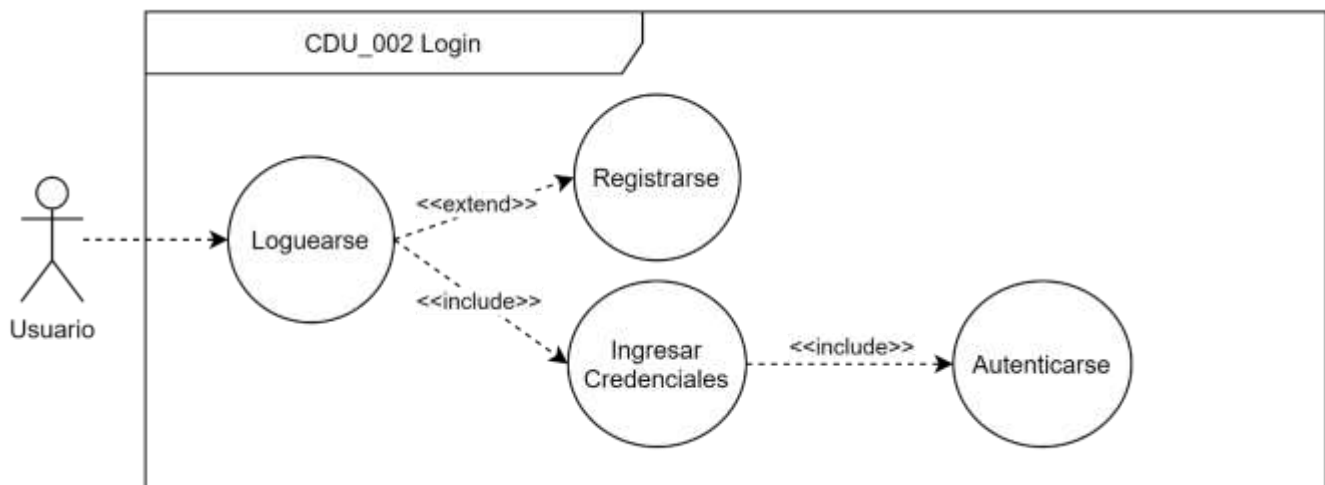
1. Eficiencia: Este sistema hará más rápido el proceso de denuncias en la municipalidad.
2. Seguridad: en el sistema el único autorizado para registrar a los empleados municipales será el administrador.
3. Usabilidad: Tendrá una interfaz amigable y fácil de usar por el usuario, además permitirá la mejor organización de las actividades de cada usuario.
4. Accesibilidad: Debido a que es un sistema web podrá accederse a él desde una computadora portátil, desde un Smartphone o desde cualquier punto que tenga acceso a internet.
5. Evolucionabilidad: Estará dispuesto a aceptar cambios que le permitan satisfacer nuevos requerimientos.
6. Robustez: Cada vez que se introduzca un dato incorrecto o que se lleve a cabo cualquier actividad anómala en el sistema, este desplegará errores y corregirá los que sean posibles, lo que garantizarán su robustez

CASOS DE USO

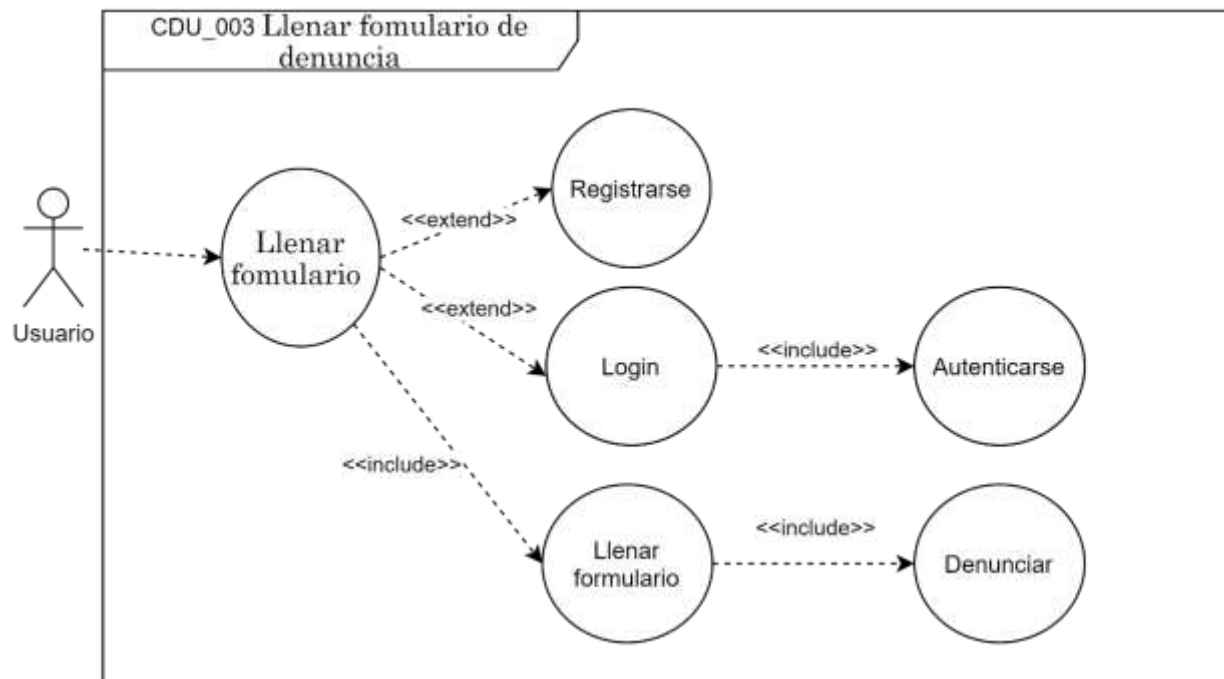
Caso de Uso	CDU_001 Registro Usuario (Vecino)
Descripción	Este caso de uso se da cuando un vecino del municipio quiere realizar una denuncia y desea hacer uso del sistema.
Flujo básico	<ol style="list-style-type: none">1. Dirigirse a la pestaña de registro e ingresa los datos solicitados por el sistema.2. En la parte de abajo del formulario deberá ingresar nuevamente la contraseña.3. Presiona el botón registrarse.4. El sistema verifica que las contraseñas coincidan.5. El sistema verifica que el usuario ingresado no exista.6. El sistema realiza el registro del usuario en la base de datos.7. Se redirige a la pestaña de login.
Flujo Alterno	<ol style="list-style-type: none">1. Faltan datos: si el vecino al momento de registrar su usuario deja campos del formulario en blanco, el sistema mostrará un mensaje de error indicando que debe llenarlos.2. Contraseña incorrecta: si las contraseñas ingresadas no coinciden el sistema la pedirá de nuevo.3. Usuario existente: si el usuario que el vecino ingresó ya fue registrado con anterioridad, el sistema mostrará un mensaje indicando que debe utilizar otro.
Requerimientos Especiales	<ul style="list-style-type: none">• Tener un correo electrónico
Punto de ampliación	Login de usuario



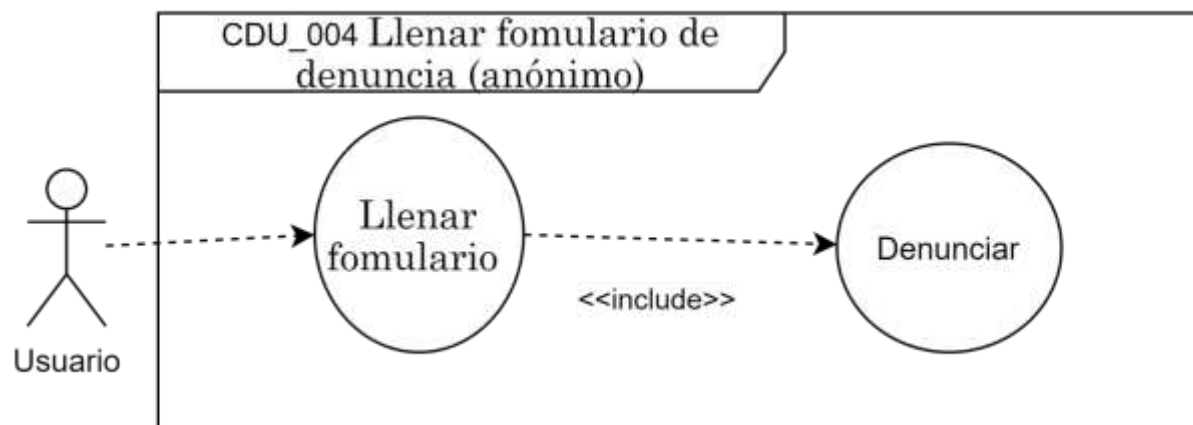
Caso de Uso	CDU_002 Login (Vecino)
Descripción	Este caso de uso se da cuando un vecino del municipio quiere realizar una denuncia y desea hacer uso del sistema y además ya se encuentra registrado.
Flujo básico	<ol style="list-style-type: none"> 1. Se ingresa a la pestaña de login. 2. Se ingresa el usuario y la contraseña correspondiente 3. Se presiona el botón de login. 4. El sistema verifica las credenciales. 5. Se ingresa al sistema.
Flujo Alternativo	<ol style="list-style-type: none"> 1. Verificación de credenciales: si el usuario ingresa la contraseña o el usuario incorrecto, el sistema notificará para que el usuario vuelva a ingresar las credenciales.
Requerimientos Especiales	<ul style="list-style-type: none"> • Estar previamente registrado en el sistema.
Punto de ampliación	Llenar formulario de denuncia



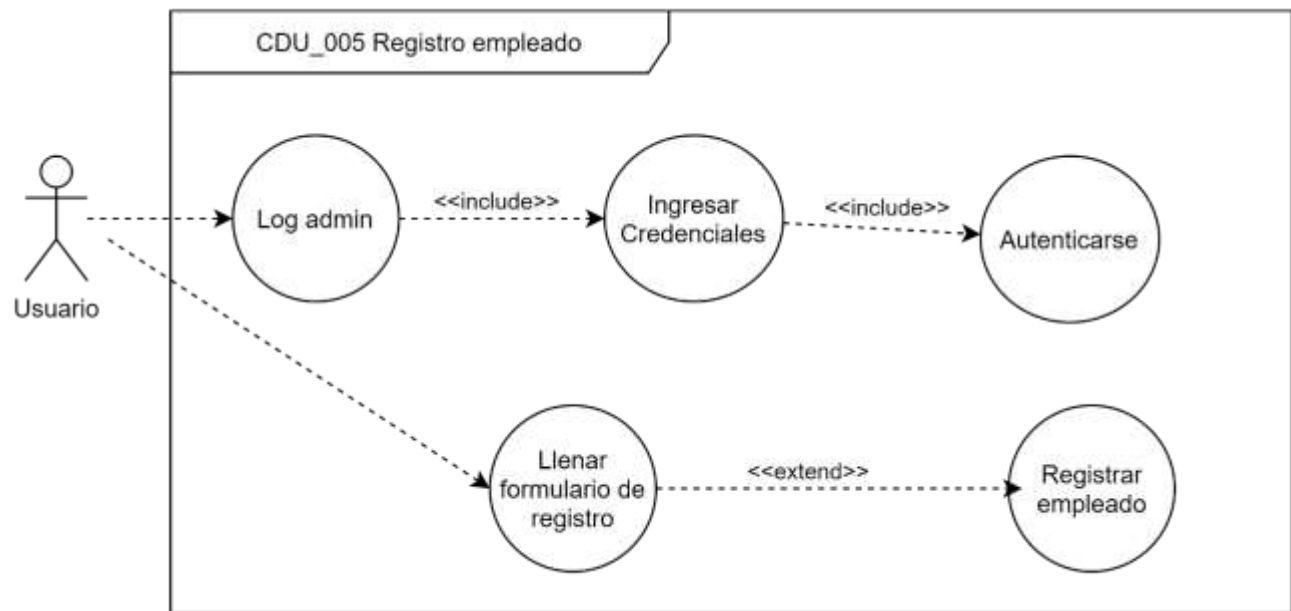
Caso de Uso	CDU_003 Llenar formulario de denuncia (Vecino)
Descripción	Este caso de uso se da cuando un vecino del municipio quiere realizar una denuncia y desea hacer uso del sistema para ello debe estar registrado e iniciar sesión luego llenar un formulario para realizar la denuncia.
Flujo básico	<ol style="list-style-type: none"> 1. Después de haber iniciado sesión se debe dirigir a la pestaña de denuncia. 2. Debe llenar el formulario correspondiente. 3. Presiona el botón de denunciar. 4. Se registra la denuncia en la base de datos.
Flujo Alternativo	<ol style="list-style-type: none"> 1. Campos en blanco: si el usuario dejó campos en blanco el sistema los solicitará.
Requerimientos Especiales	<ul style="list-style-type: none"> • Tener pruebas gráficas del problema. Al menos una imagen
Punto de ampliación	Ver Notificaciones



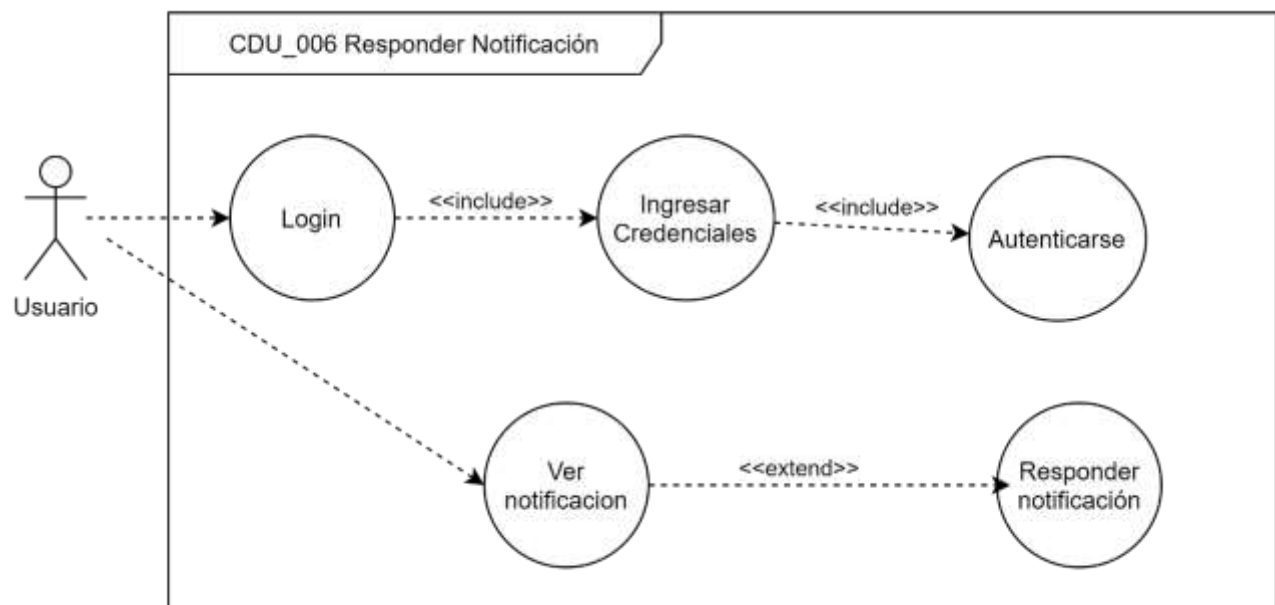
Caso de Uso	CDU_004 Denuncia anónima (Vecino)
Descripción	Este caso de uso se da cuando un vecino del municipio quiere realizar una denuncia y desea hacer uso del sistema, pero quiere que sea de forma anónima.
Flujo básico	<ol style="list-style-type: none"> 1. Ingresar al sistema como usuario anónimo. 2. Llenar el formulario de denuncia. 3. Presionar el botón para denunciar.
Flujo Alterno	<ol style="list-style-type: none"> 1. Campos en blanco: si el usuario dejó campos en blanco el sistema los solicitará.
Requerimientos Especiales	<ul style="list-style-type: none"> • Tener pruebas gráficas del problema. Al menos una imagen
Punto de ampliación	



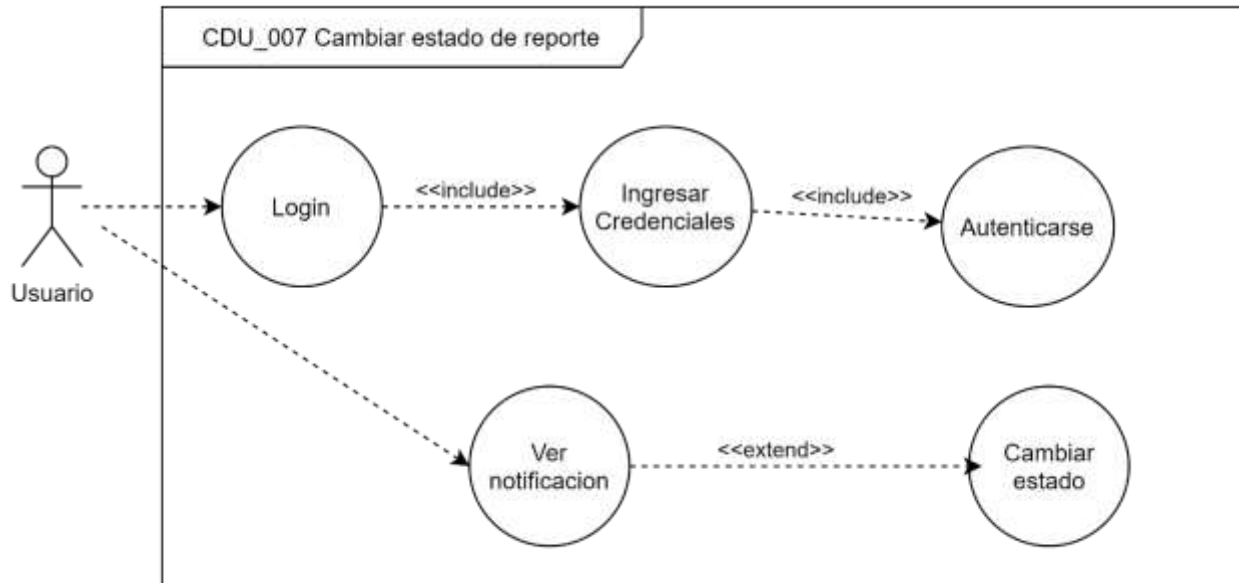
Caso de Uso	CDU_005 Registrar empleado
Descripción	Este caso de uso se da cuando el administrador del sistema desea registrar el usuario de un empleado de la municipalidad.
Flujo básico	<ol style="list-style-type: none"> 1. El empleado solicita al administrador por escrito la creación del usuario. 2. El administrador inicia sesión y se dirige a la pestaña de registro. 3. El administrador llena el respectivo formulario de registro. 4. El administrador presiona el botón de registro. 5. El sistema verifica el usuario. 6. El usuario se registra.
Flujo Alterno	<ol style="list-style-type: none"> 1. Campos en blanco: si faltan datos el sistema los solicitará. 2. Verificación de usuario: si el usuario ya existe el sistema mostrará un mensaje de error.
Requerimientos Especiales	<ol style="list-style-type: none"> 1. Correo del empleado. 2. Usuario requerido por el empleado. 3. Contraseña requerida por el usuario.
Punto de ampliación	Ver reportes de denuncias



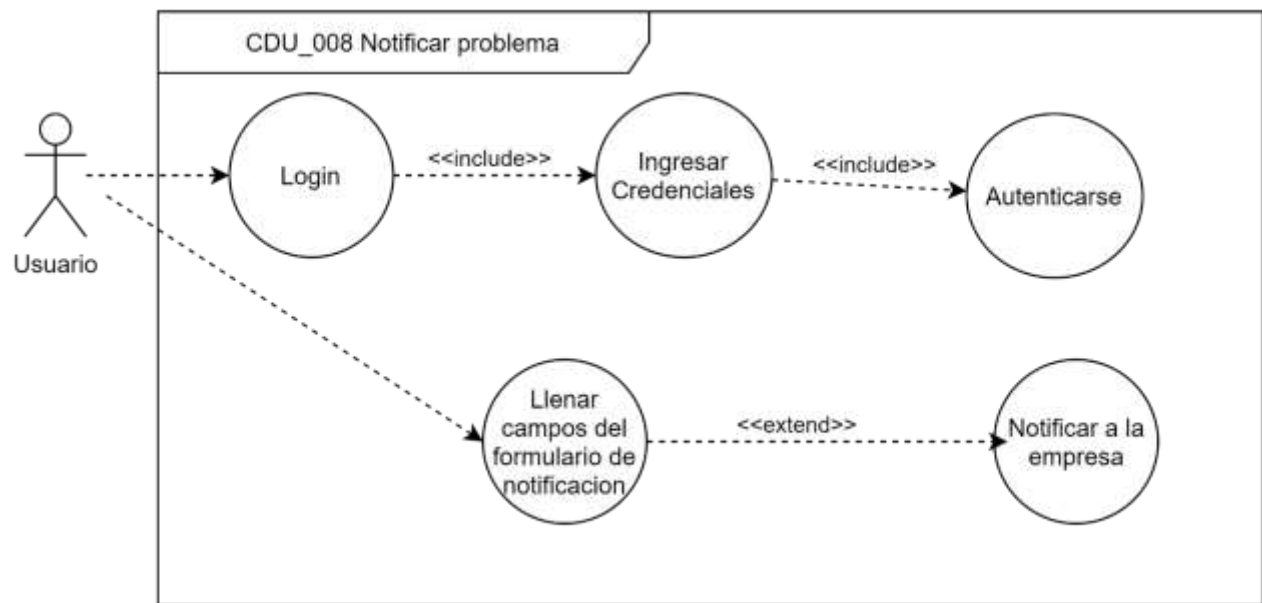
Caso de Uso	CDU_006 Responder denuncia
Descripción	Este caso de uso se da cuando un empleado municipal desea ver las notificaciones pendientes y así responderla.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema y se dirige a la pestaña de notificaciones. 2. Se muestra la lista de las notificaciones y su estado actual. 3. Se hace click en la notificación para ver el detalle. 4. El usuario ingresa el texto de respuesta para la notificación
Flujo Alternativo	<ol style="list-style-type: none"> 1. Campo en blanco: sino se ingresa un texto de respuesta el sistema mostrará un error.
Requerimientos Especiales	<ul style="list-style-type: none"> • Usuario municipal logueado.
Punto de ampliación	Cambiar estado de denuncia



Caso de Uso	CDU_007 Cambiar estado de reporte
Descripción	Este caso de uso se da cuando un empleado municipal desea cambiar el estado de las notificaciones de los reportes.
Flujo básico	<ol style="list-style-type: none"> 1. El usuario ingresa al sistema y se dirige a la pestaña de notificaciones. 2. Se muestra la lista de las notificaciones y su estado actual. 3. Se hace click en la notificación para ver el detalle. 4. Se selecciona el estado a actualizar 5. Se presiona actualizar y se guarda el estado seleccionado.
Flujo Alternativo	
Requerimientos Especiales	<ul style="list-style-type: none"> • Usuario municipal logueado.
Punto de ampliación	Notificar a empresa



Caso de Uso	CDU_008 Notificar problema
Descripción	Este caso de uso se da cuando un empleado municipal desea notificar a una determinada empresa el problema.
Flujo básico	<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Ingresar a la pestaña de reportes a empresas, ingresar el correo de la empresa. 3. Ingresar la descripción del problema. 4. Enviar correo.
Flujo Alterno	<ol style="list-style-type: none"> 1. Correo: si el correo no se ingresa o es invalido el sistema notificará de dicho error. 2. Descripción: si no se ingresa una descripción el sistema mostrará un error.
Requerimientos Especiales	<ul style="list-style-type: none"> • Usuario municipal logueado. • Correo de la institución • Descripción del problema.
Punto de ampliación	--



Historias de usuario

Registro de vecino	
Descripción	
Como vecino quiero poder registrarme en la aplicación para poder reportar cualquier anomalía, problema o desperfecto que suceda en espacios municipales.	
Acción y pasos a tomar	Prioridad
1. Ingresar mediante la aplicación móvil al registro de usuario. 2. Brindar la información necesaria para registrar al vecino. 3. presionar el botón de registrar.	100
	Tamaño
	Mediana
Condición previa	
<ul style="list-style-type: none">Tener instalada la aplicación móvil.	

Registro empleado municipal	
Descripción	
Como administrador quiero poder registrar a los empleados nuevos para que estos sean los responsables de revisar, notificar y solucionar los reportes que los vecinos envíen.	
Acción y pasos a tomar	Prioridad
1. Ingresar a la aplicación web. 2. Autenticarse. 3. Seleccionar la opción de agregar empleado. 4. Ingresar la información del empleado. 5. Presionar el botón de registrar.	80
	Tamaño
	Mediana
Condición previa	
<ul style="list-style-type: none">Tener credenciales como administrador.	

Reportar problema	
Descripción	
Como vecino quiero llenar un formulario para reportar cualquier anomalía, problema o desperfecto que suceda en espacios municipales para que la municipalidad pueda brindar una solución.	
Acción y pasos a tomar	Prioridad
1. Seleccionar opción reportar problema en la aplicación móvil. 2. Brindar la información necesaria para poder reportar el problema. 3. Presionar el botón de reportar.	100
	Tamaño
	Mediana
Condición previa	
<ul style="list-style-type: none">Tener instalada la aplicación móvil.	

Reportar problema anónimamente	
Descripción	
Como vecino quiero llenar un formulario para reportar cualquier anomalía, problema o desperfecto que suceda en espacios municipales de forma anónima para que la municipalidad pueda brindar una solución.	
Acción y pasos a tomar	Prioridad
<ol style="list-style-type: none"> 1. Seleccionar opción reportar problema de forma anónima en la aplicación móvil. 2. Brindar la información necesaria para poder reportar el problema. 3. Presionar el botón de reportar. 	70
	Tamaño
	Mediana
Condición previa	
<ul style="list-style-type: none"> • Tener instalada la aplicación móvil. 	

Ver reporte	
Descripción	
Como empleado municipal quiero poder visualizar todos los reportes realizados por los vecinos para poder brindar una solución óptima para el problema.	
Acción y pasos a tomar	Prioridad
<ol style="list-style-type: none"> 1. Autenticarse en la aplicación web. 2. Ir al apartado de notificaciones. 3. Seleccionar el reporte que desea visualizar. 	100
	Tamaño
	Mediana
Condición previa	
<ul style="list-style-type: none"> • Tener credenciales como empleado municipal. 	

Notificación a empresas	
Descripción	
Como empleado municipal quiero poder informar a cualquier empresa externa para que pueda brindar una solución que los empleados municipales no puedas resolver.	
Acción y pasos a tomar	Prioridad
<ol style="list-style-type: none"> 1. Seleccionar opción notificar a. 2. Ingresar el correo de la empresa externa. 3. Ingresar la descripción del problema. 4. Presionar enviar notificación. 	30
	Tamaño
	Pequeña
Condición previa	

Responder notificación	
Descripción	
Como empleado municipal quiero poder responder con un mensaje a los reportes para que el vecino se encuentre informado del estatus del reporte.	
Acción y pasos a tomar	Prioridad
1. Seleccionar el reporte deseado. 2. Presionar la opción responder. 3. Escribir el mensaje a enviar. 4. Enviar el mensaje.	100
	Tamaño
	Pequeña
Condición previa	
<ul style="list-style-type: none"> El reporte no debe estar en estado solucionado. 	

Login Vecino	
Descripción	
Como vecino quiero poder autenticarme en la aplicación móvil para poder enviar reportes a la municipalidad y tener un historial de estos reportes.	
Acción y pasos a tomar	Prioridad
1. Ingresar a la aplicación móvil. 2. Escribir las credenciales del vecino. 3. Autenticarse.	100
	Tamaño
	Pequeña
Condición previa	
<ul style="list-style-type: none"> Tener instalada la aplicación móvil. 	

Login Empleado	
Descripción	
Como empleado municipal quiero poder autenticarme mediante la aplicación web para poder revisar, notificar y solucionar los reportes que envíen los usuarios.	
Acción y pasos a tomar	Prioridad
1. Ingresar a la aplicación web. 2. Escribir las credenciales del empleado. 3. Autenticarse	100
	Tamaño
	Pequeña
Condición previa	
<ul style="list-style-type: none"> Debe haber sido registrado previamente por un administrador. 	

Mapeo de Historias



MVP | 11



miro

Diagrama de Clases Web

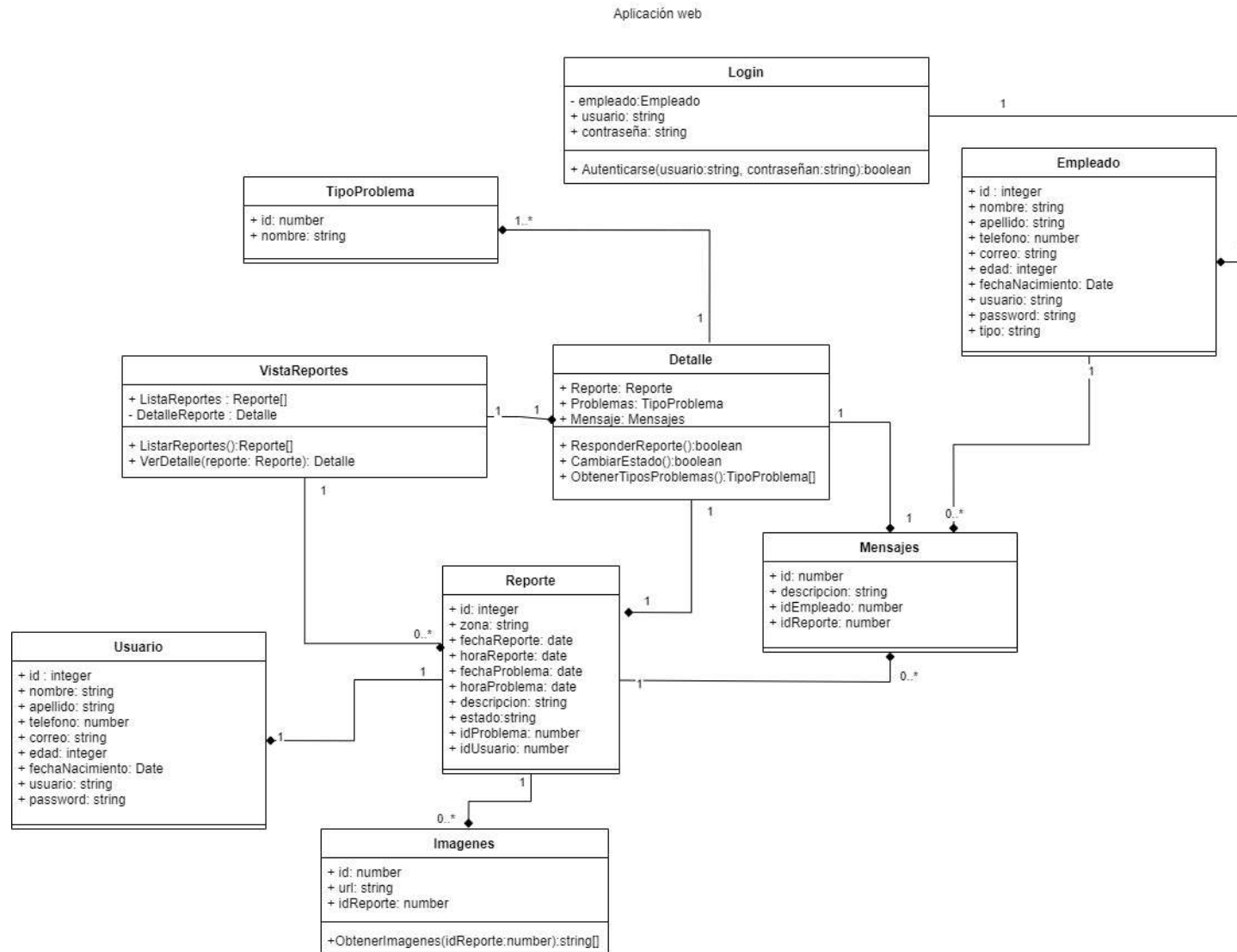
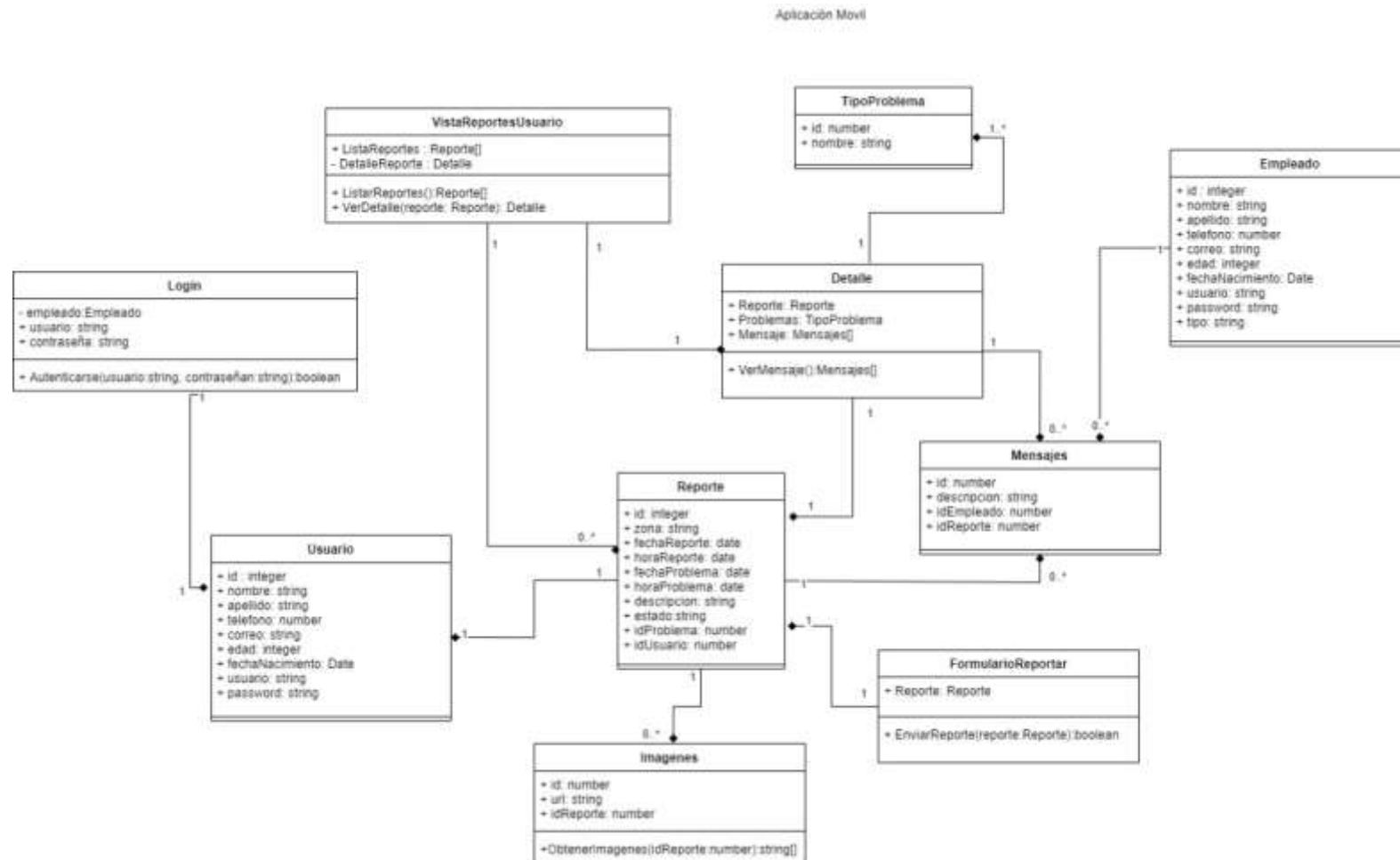
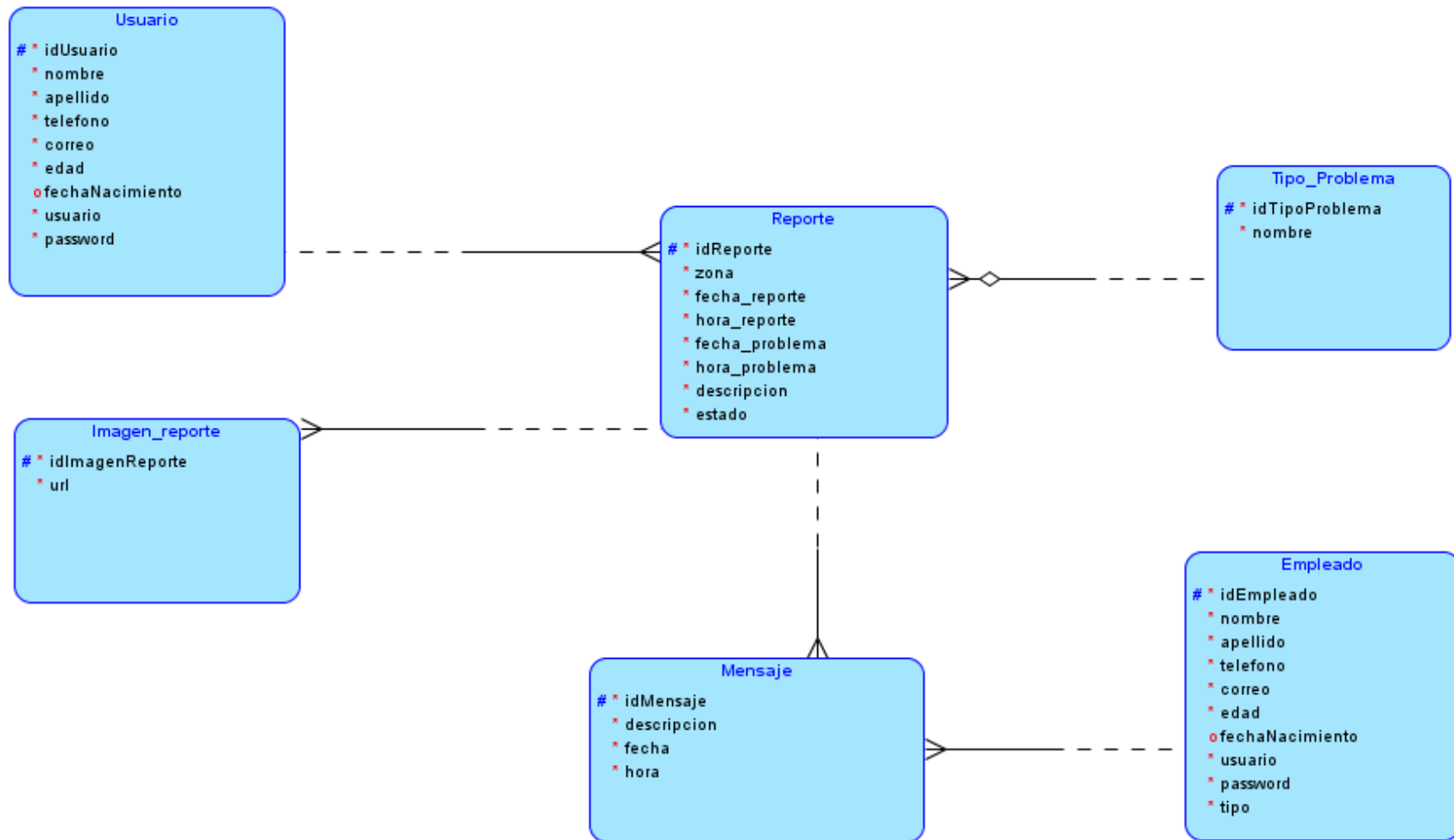


Diagrama de Clases APP Móvil

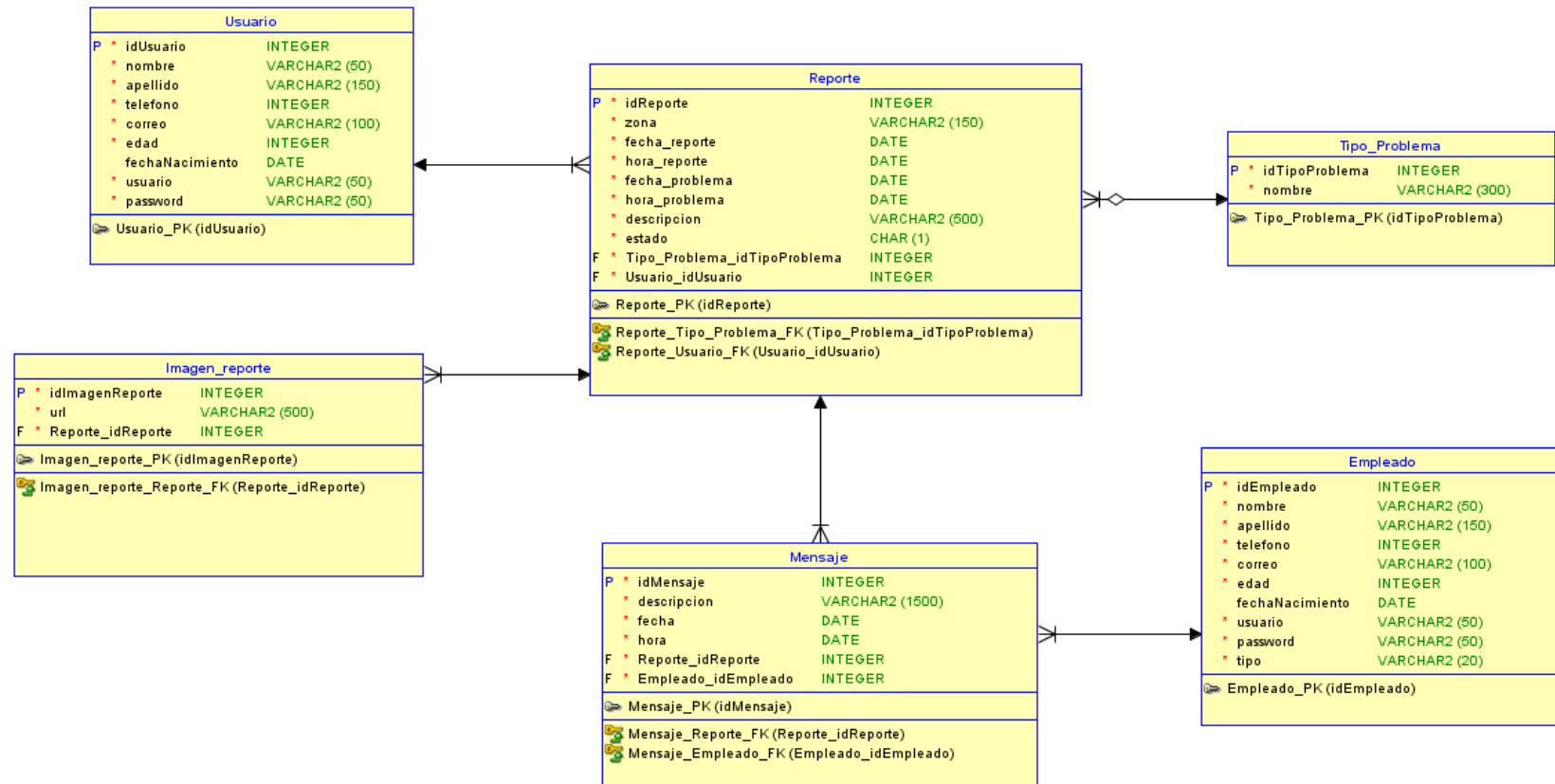


Modelos de la base de datos:

Lógico



Conceptual



Físico

```
17 CREATE TABLE empleado (
18     idempleado      INTEGER NOT NULL,
19     nombre           VARCHAR2(50) NOT NULL,
20     apellido         VARCHAR2(150) NOT NULL,
21     telefono         INTEGER NOT NULL,
22     correo           VARCHAR2(100) NOT NULL,
23     edad             INTEGER NOT NULL,
24     fechanacimiento DATE,
25     usuario          VARCHAR2(50) NOT NULL,
26     password         VARCHAR2(50) NOT NULL,
27     tipo             VARCHAR2(20) NOT NULL
28 );
29
30 ALTER TABLE empleado ADD CONSTRAINT empleado_pk PRIMARY KEY ( idempleado );
31
32 CREATE TABLE imagen_reporte (
33     idimagenreporte  INTEGER NOT NULL,
34     url              VARCHAR2(500) NOT NULL,
35     reporte_idreporte INTEGER NOT NULL
36 );
37
38 ALTER TABLE imagen_reporte ADD CONSTRAINT imagen_reporte_pk PRIMARY KEY ( idimagenreporte );
39
40 CREATE TABLE mensaje (
41     idmensaje        INTEGER NOT NULL,
42     descripcion      VARCHAR2(1500) NOT NULL,
43     fecha            DATE NOT NULL,
44     hora             DATE NOT NULL,
45     reporte_idreporte INTEGER NOT NULL,
46     empleado_idempleado INTEGER NOT NULL
47 );
48
49 ALTER TABLE mensaje ADD CONSTRAINT mensaje_pk PRIMARY KEY ( idmensaje );
50
51 CREATE TABLE reporte (
52     idreporte        INTEGER NOT NULL,
53     zona             VARCHAR2(150) NOT NULL,
54     fecha_reporte    DATE NOT NULL,
55     hora_reporte     DATE NOT NULL,
56     fecha_problema   DATE NOT NULL,
57     hora_problema    DATE NOT NULL,
58     descripcion      VARCHAR2(500) NOT NULL,
59     estado           CHAR(1) NOT NULL,
60     tipo_problema_idtipoproblema INTEGER NOT NULL,
61     usuario_idusuario INTEGER NOT NULL
62 );
63
64 ALTER TABLE reporte ADD CONSTRAINT reporte_pk PRIMARY KEY ( idreporte );
65
66 CREATE TABLE tipo_problema (
67     idtipoproblema   INTEGER NOT NULL,
68     nombre           VARCHAR2(300) NOT NULL
69 );
70
71 ALTER TABLE tipo_problema ADD CONSTRAINT tipo_problema_pk PRIMARY KEY ( idtipoproblema );
72
73 CREATE TABLE usuario (
74     idusuario        INTEGER NOT NULL,
75     nombre           VARCHAR2(50) NOT NULL,
76     apellido         VARCHAR2(150) NOT NULL,
77     telefono         INTEGER NOT NULL,
78     correo           VARCHAR2(100) NOT NULL,
79     edad             INTEGER NOT NULL,
80     fechanacimiento DATE,
81     usuario          VARCHAR2(50) NOT NULL,
82     password         VARCHAR2(50) NOT NULL
83 );
84
85 ALTER TABLE usuario ADD CONSTRAINT usuario_pk PRIMARY KEY ( idusuario );
86
87 ALTER TABLE imagen_reporte
88     ADD CONSTRAINT imagen_reporte_reporte_fk FOREIGN KEY ( reporte_idreporte )
89     REFERENCES reporte ( idreporte );
90
```

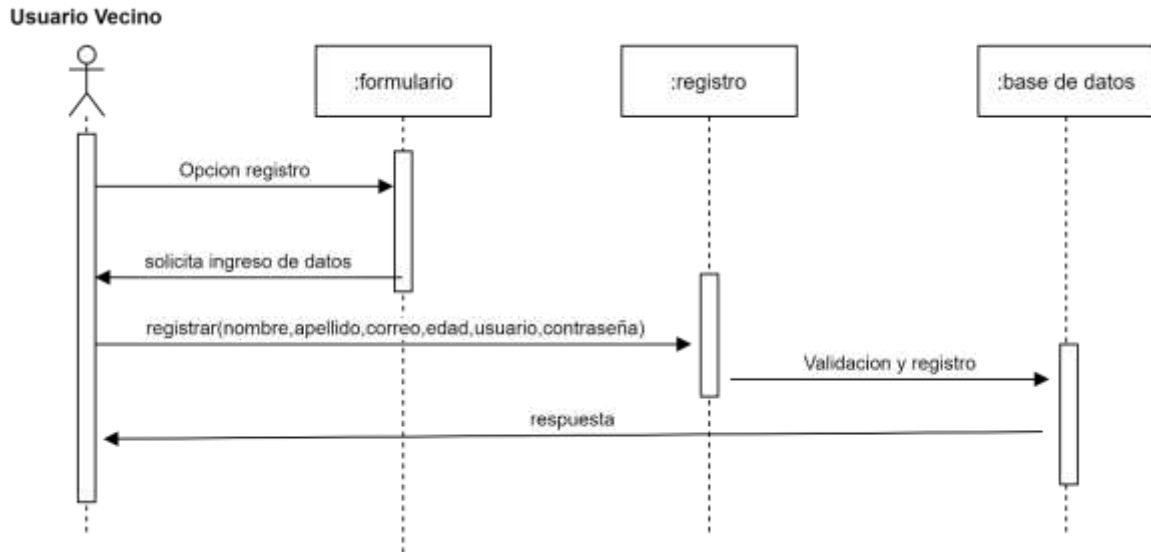
```

85
86 ALTER TABLE mensaje
87     ADD CONSTRAINT mensaje_empleado_fk FOREIGN KEY ( empleado_idempleado )
88     REFERENCES empleado ( idempleado );
89
90 ALTER TABLE mensaje
91     ADD CONSTRAINT mensaje_reporte_fk FOREIGN KEY ( reporte_idreporte )
92     REFERENCES reporte ( idreporte );
93
94 ALTER TABLE reporte
95     ADD CONSTRAINT reporte_tipo_problema_fk FOREIGN KEY ( tipo_problema_idtipoproblema )
96     REFERENCES tipo_problema ( idtipoproblema );
97
98 ALTER TABLE reporte
99     ADD CONSTRAINT reporte_usuario_fk FOREIGN KEY ( usuario_idusuario )
100     REFERENCES usuario ( idusuario );
101
102 CREATE OR REPLACE TRIGGER fntm_reporte BEFORE
103     UPDATE OF tipo_problema_idtipoproblema ON reporte
104 BEGIN
105     raise_application_error(-20225, 'Non Transferable FK constraint on table Reporte is violated');
106 END;
107 /
108

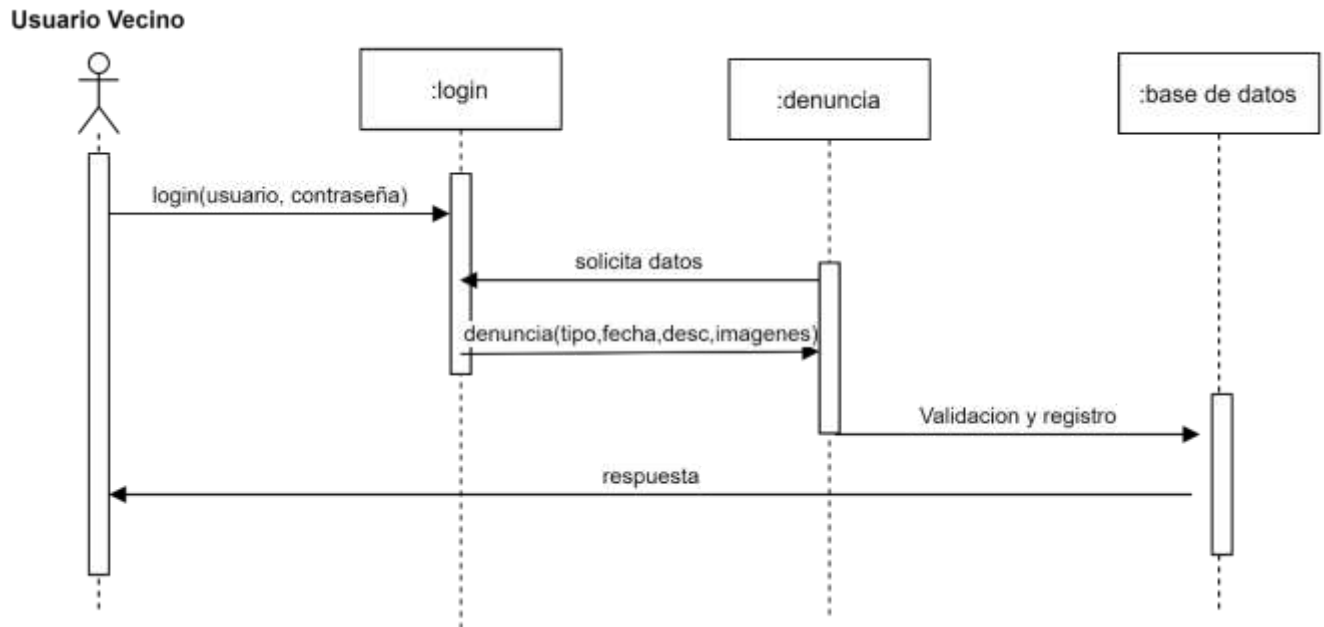
```

DIAGRAMAS DE SECUENCIAS

Registro Usuario Vecino



Realizar Denuncia



Responder Notificación

Usuario Empleado

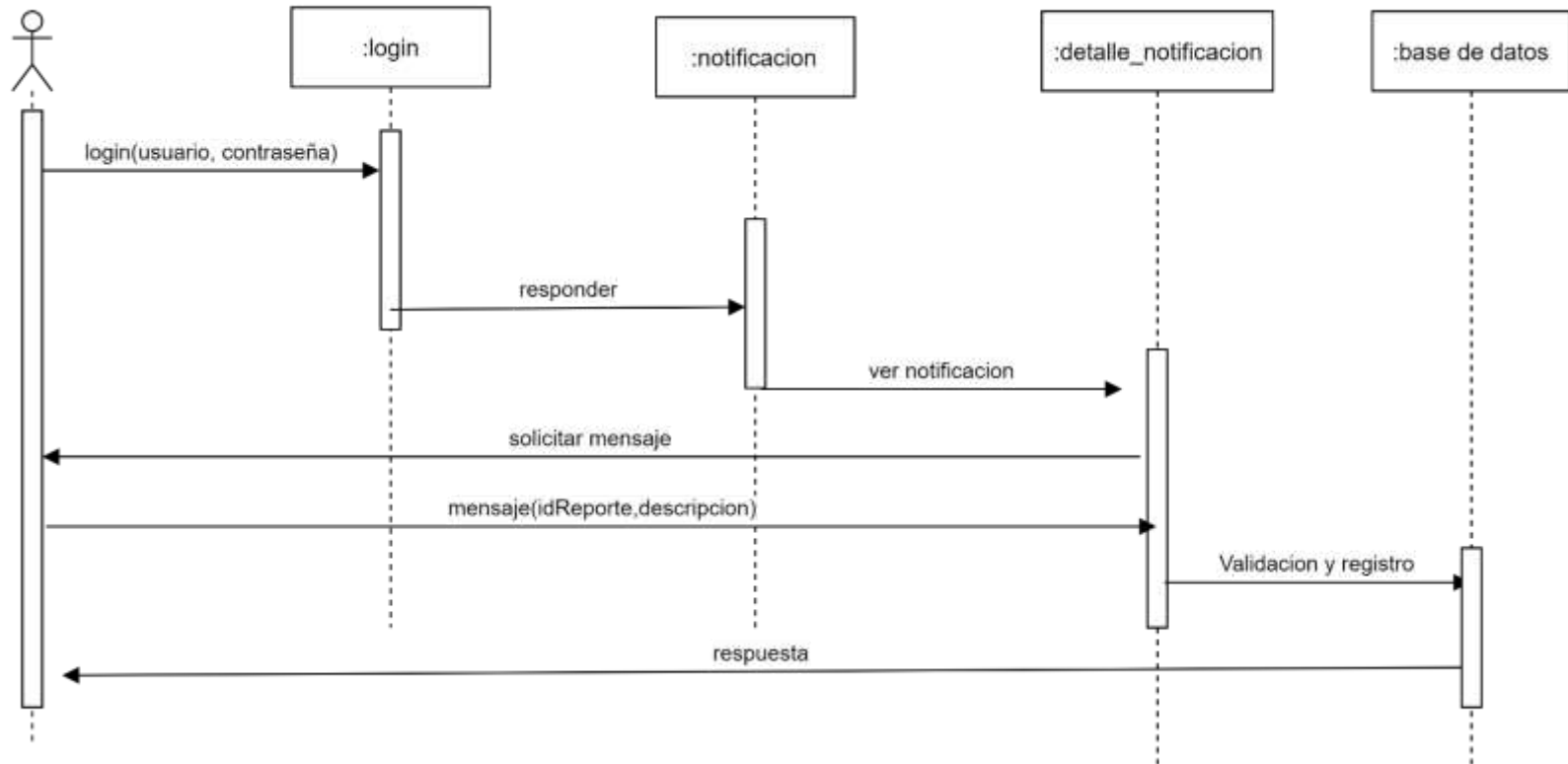


DIAGRAMA DE COMPONENTES

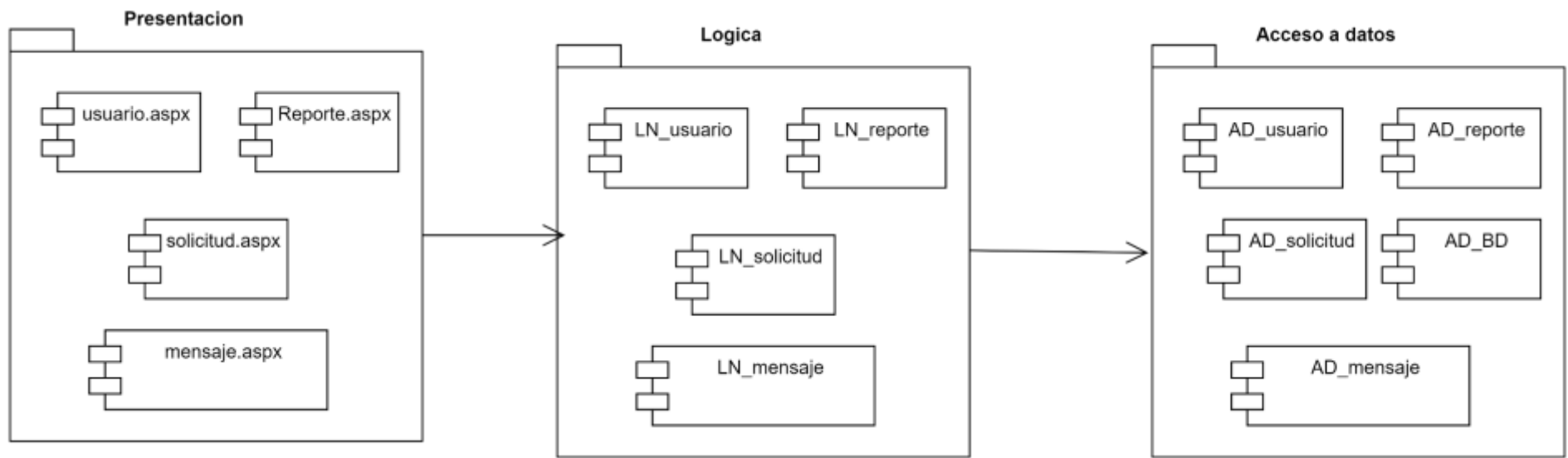
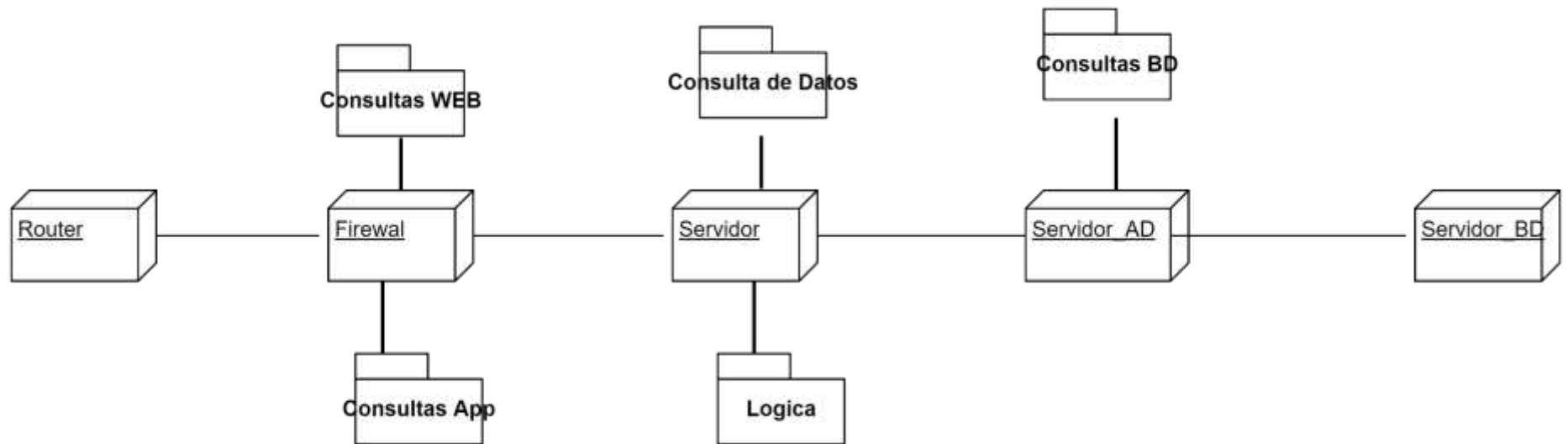


DIAGRAMA DE DESPLIEGUE



Pantallas web y app