

SERVICIO NACIONAL DE APRENDIZAJE SENA

Centro: Centro de Diseño e Innovación Tecnológica Industrial

Ficha: 3235906

Programa de Formación: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Nombre del Aprendiz: Laura Barona Saavedra

Nombre del Instructor: Andrés Felipe Parra Martínez

Fecha: 9 de septiembre 2025

INTRODUCCION

El Lenguaje Unificado de Modelado (UML) es un estándar utilizado en ingeniería de software para visualizar, especificar, construir y documentar los artefactos de un sistema. UML permite representar de manera gráfica los diferentes aspectos de un sistema, facilitando la comunicación entre analistas, desarrolladores, diseñadores y usuarios.

Dentro de UML, los **diagramas de caso de uso** son fundamentales, ya que muestran cómo interactúan los usuarios (actores) con el sistema para lograr determinados objetivos, definiendo así los requisitos funcionales del software.

UML

El UML (Unified Modeling Language) es un lenguaje de modelado visual utilizado en el desarrollo de software. Permite representar gráficamente los elementos, procesos, interacciones y arquitectura de un sistema. No es un lenguaje de programación, sino una herramienta de especificación y diseño.

IMPORTANCIA DE UML

- Facilita la **comunicación** entre los diferentes actores de un proyecto (analistas, clientes, desarrolladores).
- Ayuda a **documentar** de manera clara los requisitos y la estructura del sistema.
- Permite **visualizar** cómo funcionará el sistema antes de implementarlo.
- Reduce riesgos al detectar errores conceptuales en etapas tempranas.
- Proporciona una **base estandarizada** para proyectos de software complejos.

CLASIFICACIÓN DE LOS DIAGRAMAS UML

Los diagramas UML se dividen en dos grandes categorías:

1. Diagramas estructurales (qué compone el sistema).
 - Diagrama de clases
 - Diagrama de objetos
 - Diagrama de componentes
 - Diagrama de despliegue
 - Diagrama de paquetes
2. Diagramas de comportamiento (cómo funciona el sistema).
 - Diagrama de casos de uso
 - Diagrama de secuencia
 - Diagrama de actividades
 - Diagrama de estados
 - Diagrama de comunicación
 - Diagrama de tiempo

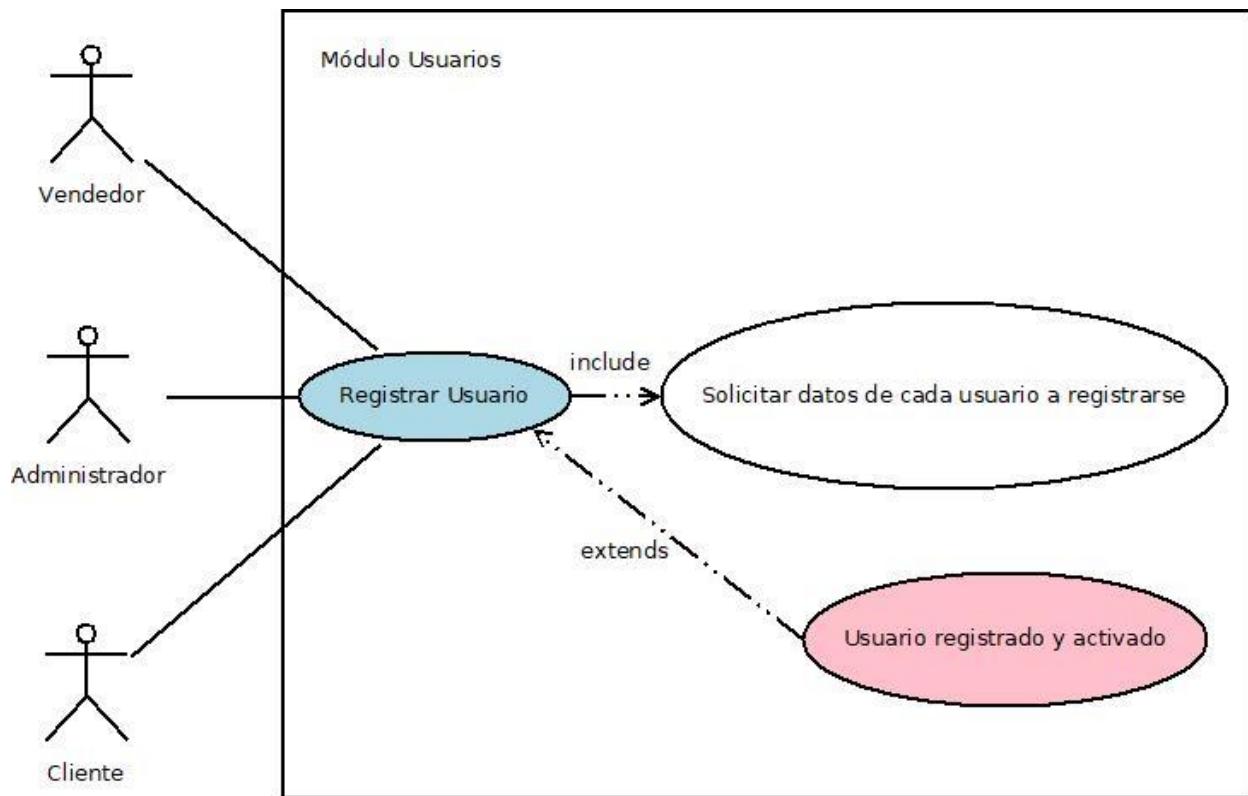
DIAGRAMAS UML MÁS USADOS Y POR QUÉ

- **Diagrama de casos de uso:** porque define los requisitos funcionales desde la perspectiva del usuario.
- **Diagrama de clases:** porque modela la estructura lógica del sistema (entidades, atributos, relaciones).
- **Diagrama de secuencia:** porque describe el flujo de mensajes entre objetos, útil para modelar la lógica de negocio.
- **Diagrama de actividades:** porque permite representar procesos o flujos de trabajo de forma clara.

Estos son los más utilizados porque aportan la visión general necesaria en el análisis y diseño, y ayudan a conectar los requisitos con la implementación.

DIAGRAMAS DE CASO DE USO

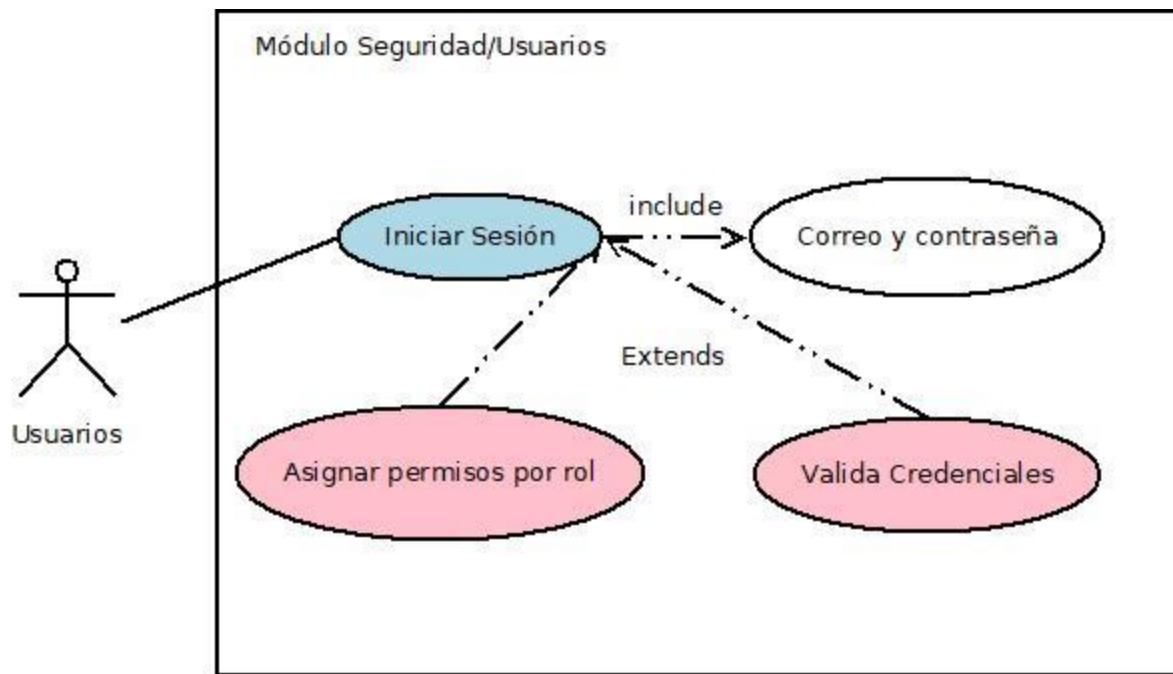
Módulo: Usuarios



ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Registrar usuarios
Actor(es)	Usuario, Administrador, Vendedor
Objetivo	Permitir que un usuario se registre en el sistema (puede ser comprador, vendedor o administrador).

Precondiciones	El usuario no debe estar previamente registrado.
Postcondiciones	El usuario queda registrado en la base de datos con su rol asignado.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa sus datos en el formulario. 2. El sistema valida que los datos sean correctos y completos. 3. El sistema registra al usuario con el rol correspondiente.
Flujos alternativos	<ol style="list-style-type: none"> 1. Si el correo ya existe → mostrar mensaje de error. 2. Si los datos están incompletos → solicitar corrección.
Requisitos relacionados	RF-01: Registrar usuarios
Observaciones	El administrador también puede registrar usuarios de manera manual.

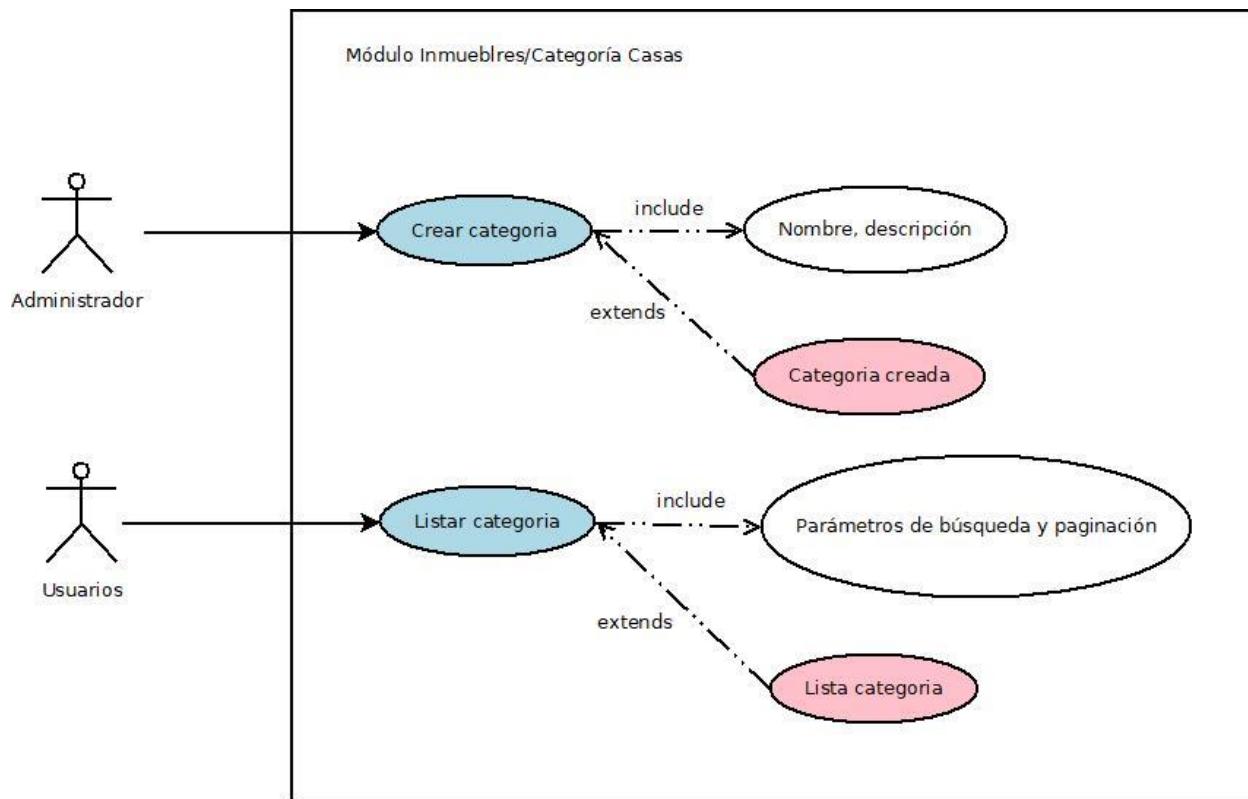
Módulo: Seguridad / Usuarios



ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Autenticación de usuarios y control de permisos por rol
Actor(es)	Usuario registrado (comprador, vendedor, administrador).
Objetivo	Garantizar el acceso seguro al sistema mediante autenticación y asignación de permisos según rol.
Precondiciones	El usuario debe estar registrado en el sistema.

Postcondiciones	El usuario accede al sistema con los permisos de su rol.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa sus credenciales. 2. El sistema valida los datos. 3. El sistema permite acceso y carga los permisos del rol.
Flujos alternativos	<ol style="list-style-type: none"> 1. Si las credenciales son inválidas → mensaje de error. 2. Si la cuenta está bloqueada → notificar al usuario.
Requisitos relacionados	RF-02: Autenticación de usuarios y control de permisos por rol
Observaciones	Se recomienda usar autenticación con token JWT.

Módulo: Inmuebles / Categorías Casas



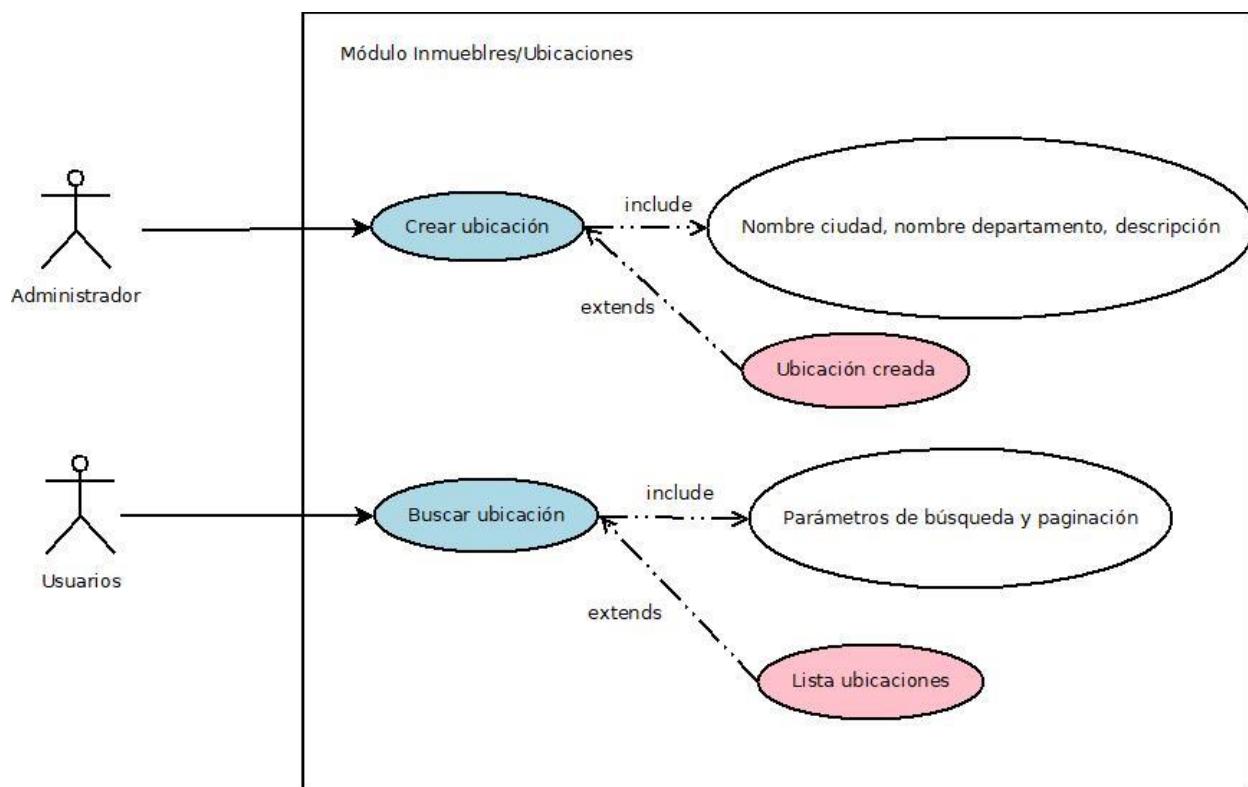
ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Crear categorías de inmuebles
Actor(es)	Administrador
Objetivo	Permitir al administrador crear nuevas categorías de inmuebles (ej: casa, apartamento).
Precondiciones	El administrador debe estar autenticado.
Postcondiciones	La nueva categoría queda registrada en la base de datos.

Flujo principal	<ol style="list-style-type: none"> 1. El administrador selecciona la opción de crear categoría. 2. Ingresa los datos de la categoría. 3. El sistema valida y guarda la nueva categoría.
Flujos alternativos	Si el nombre ya existe → mostrar error.
Requisitos relacionados	RF-03: Crear categorías de inmuebles
Observaciones	Las categorías permiten clasificar las propiedades.

ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Listar categorías de inmuebles
Actor(es)	Usuario, Administrador
Objetivo	Consultar el listado de categorías disponibles.
Precondiciones	Debe haber categorías registradas.
Postcondiciones	El sistema muestra la lista de categorías.
Flujo principal	<ol style="list-style-type: none"> 1. El actor solicita ver categorías. 2. El sistema recupera y muestra la información.

Flujos alternativos	Si no existen categorías → mostrar lista vacía.
Requisitos relacionados	RF-04: Listar categorías de inmuebles
Observaciones	Útil para búsqueda y registro de propiedades.

Módulo: Inmuebles / Ubicaciones



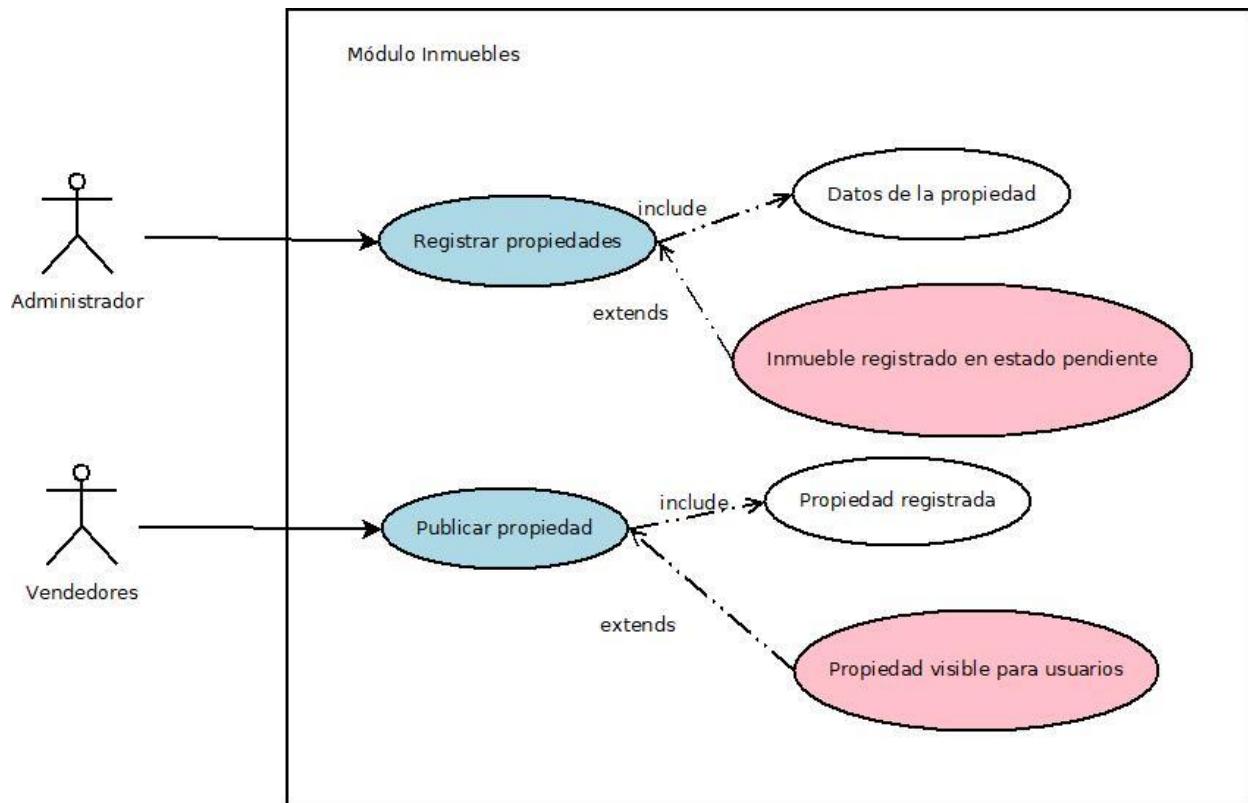
ELEMENTO	DESCRIPCIÓN

Nombre del caso de uso	Crear ubicaciones
Actor(es)	Administrador
Objetivo	Registrar nuevas ubicaciones geográficas para los inmuebles.
Precondiciones	El administrador debe estar autenticado.
Postcondiciones	La ubicación queda registrada en la base de datos.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador selecciona la opción de crear ubicación. 2. Ingresa los datos de la ubicación. 3. El sistema valida y guarda la ubicación.
Flujos alternativos	Si la ubicación ya existe → mostrar error.
Requisitos relacionados	RF-05: Crear ubicaciones
Observaciones	Base para búsquedas de propiedades.

ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Buscar ubicaciones
Actor(es)	Usuario

Objetivo	Permitir a los usuarios consultar ubicaciones registradas.
Precondiciones	Deben existir ubicaciones.
Postcondiciones	El sistema muestra la ubicación consultada.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario ingresa un criterio de búsqueda. 2. El sistema recupera y muestra las coincidencias.
Flujos alternativos	Si no existen resultados → mostrar mensaje “sin coincidencias”.
Requisitos relacionados	RF-06: Buscar ubicaciones
Observaciones	Apoya la búsqueda de propiedades filtradas.

Módulo: Inmuebles



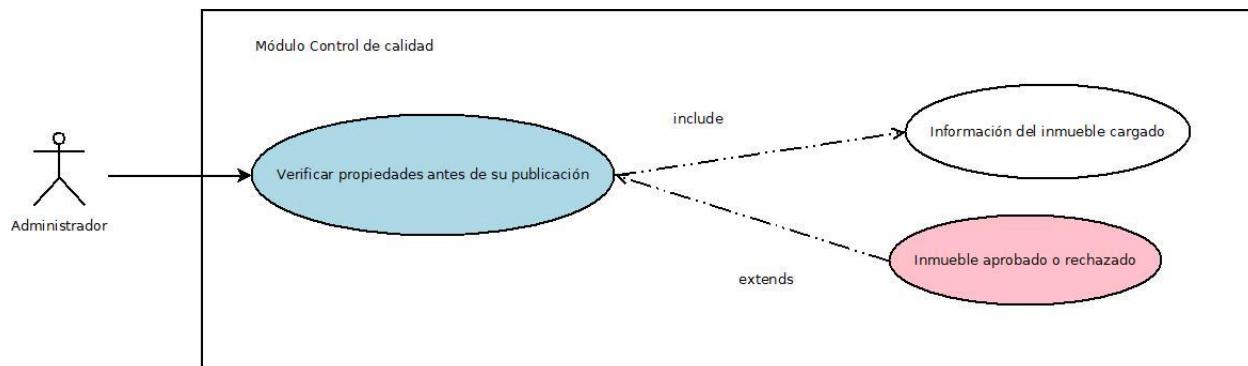
ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Registrar nuevas propiedades
Actor(es)	Vendedor, Administrador
Objetivo	Permitir registrar propiedades en el sistema.
Precondiciones	El actor debe estar autenticado y tener permisos.

Postcondiciones	La propiedad queda registrada en estado pendiente de publicación.
Flujo principal	<ol style="list-style-type: none"> 1. El actor ingresa los datos de la propiedad. 2. El sistema valida la información. 3. La propiedad queda guardada en estado “pendiente”.
Flujos alternativos	Datos incompletos → solicitar corrección.
Requisitos relacionados	RF-07: Registrar nuevas propiedades
Observaciones	El registro no implica publicación inmediata.

ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Publicar propiedades
Actor(es)	Administrador, Sistema (con scheduler).
Objetivo	Publicar las propiedades que cumplen requisitos.
Precondiciones	La propiedad debe estar registrada y aprobada.
Postcondiciones	La propiedad queda visible para los usuarios.

Flujo principal	<ol style="list-style-type: none"> 1. El administrador selecciona propiedades para publicar o el sistema publica automáticamente según la fecha. 2. El sistema cambia el estado a “publicada”.
Flujos alternativos	Si no está verificada → rechazar publicación.
Requisitos relacionados	RF-08: Publicar propiedades
Observaciones	Puede integrarse con un cronjob diario.

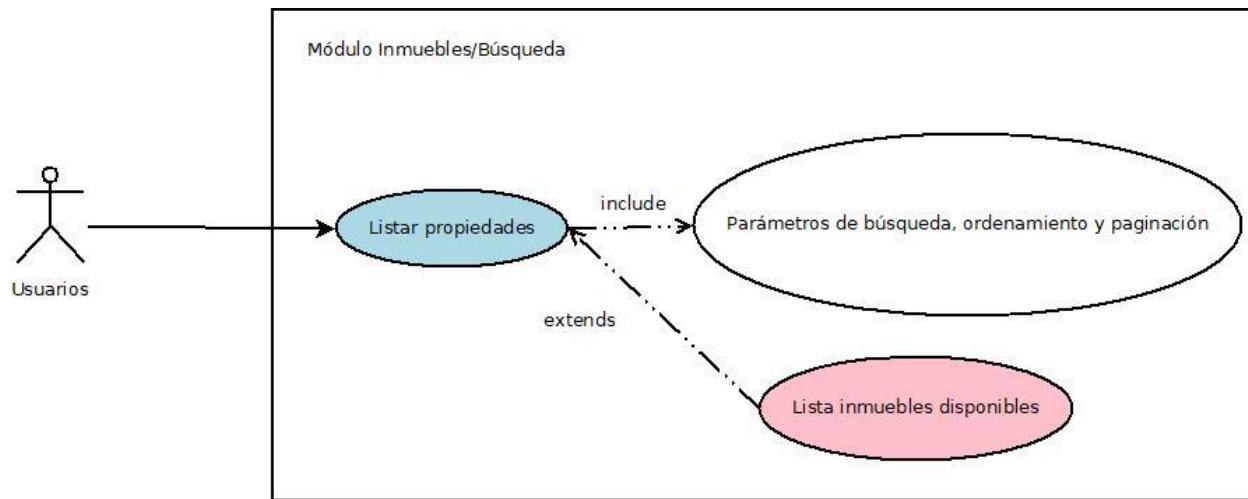
Módulo: Control de calidad



ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Verificar propiedades antes de su publicación

Actor(es)	Administrador de calidad
Objetivo	Validar que las propiedades cumplan con los criterios de publicación.
Precondiciones	La propiedad debe estar registrada.
Postcondiciones	La propiedad queda aprobada o rechazada.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador revisa la propiedad. 2. Marca como aprobada o rechazada. 3. El sistema actualiza el estado.
Flujos alternativos	Si hay información incompleta → solicitar corrección al vendedor.
Requisitos relacionados	RF-09: Verificar propiedades antes de su publicación
Observaciones	Garantiza la calidad de los anuncios.

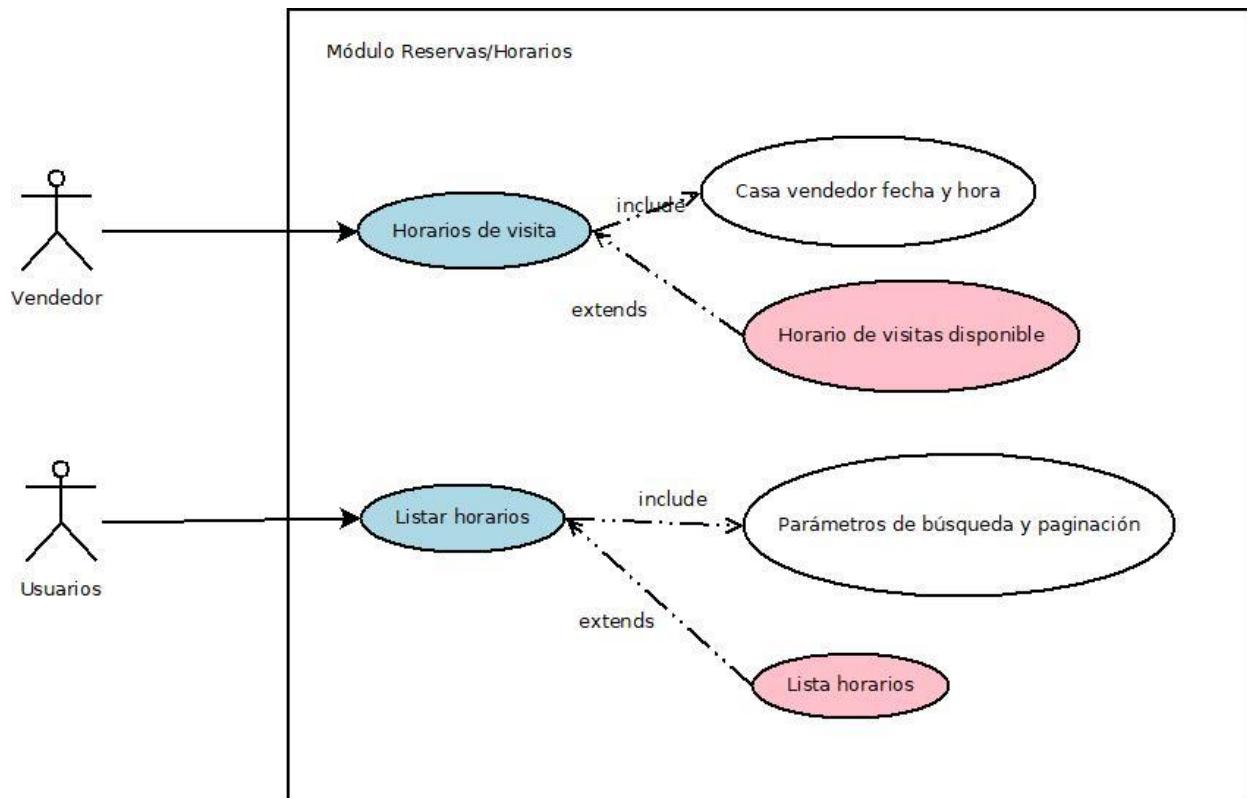
Módulo: Inmuebles / Búsqueda



ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Listar propiedades
Actor(es)	Usuario (comprador).
Objetivo	Permitir la consulta de propiedades publicadas.
Precondiciones	Deben existir propiedades publicadas.
Postcondiciones	El sistema muestra la lista de propiedades filtradas.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede a la opción de listar propiedades.

	2. El sistema muestra propiedades con filtros (ubicación, precio, categoría).
Flujos alternativos	Si no hay propiedades → mostrar mensaje.
Requisitos relacionados	RF-10: Listar propiedades
Observaciones	Incluye filtros avanzados y paginación.

Módulo: Reservas / Horarios

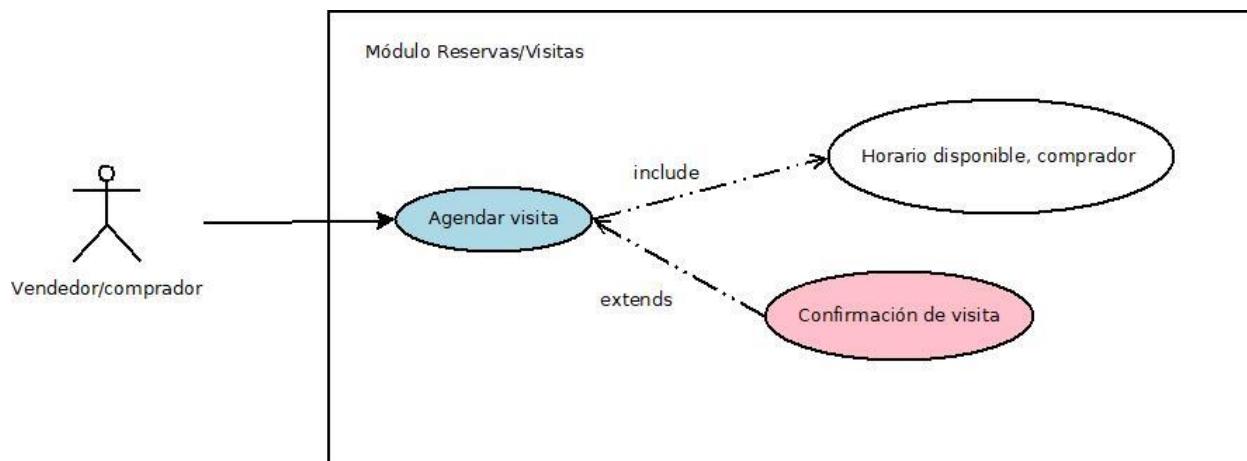


ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Disponibilizar horarios de visitas
Actor(es)	Vendedor, Administrador
Objetivo	Permitir que un vendedor o administrador registre horarios disponibles para visitas.
Precondiciones	Debe existir una propiedad publicada.
Postcondiciones	Los horarios quedan registrados como disponibles.
Flujo principal	<ol style="list-style-type: none"> 1. El actor selecciona una propiedad. 2. Ingresa los horarios disponibles. 3. El sistema guarda los horarios.
Flujos alternativos	Si el horario se cruza con otro → mostrar error.
Requisitos relacionados	RF-11: Disponibilizar horarios de visitas
Observaciones	Base para la agenda de visitas.

ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Listar horarios de visitas disponibles

Actor(es)	Usuario interesado (comprador).
Objetivo	Consultar los horarios disponibles de una propiedad.
Precondiciones	La propiedad debe estar publicada y con horarios registrados.
Postcondiciones	El sistema muestra horarios disponibles.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona una propiedad. 2. El sistema muestra los horarios disponibles.
Flujos alternativos	Si no hay horarios → mensaje informativo.
Requisitos relacionados	RF-12: Listar horarios de visitas disponibles
Observaciones	Se actualiza dinámicamente según reservas.

Módulo: Reservas / Visitas



ELEMENTO	DESCRIPCIÓN
Nombre del caso de uso	Agendar visitas
Actor(es)	Usuario interesado (comprador).
Objetivo	Permitir que un usuario reserve un horario de visita para una propiedad.
Precondiciones	Debe haber horarios disponibles.
Postcondiciones	El horario queda reservado para el usuario.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario selecciona un horario disponible. 2. El sistema valida la disponibilidad. 3. El sistema registra la visita y actualiza el estado del horario.
Flujos alternativos	Si el horario ya fue tomado → mostrar error.
Requisitos relacionados	RF-13: Agendar visitas
Observaciones	Puede integrarse con recordatorios por correo.

CONCLUSIONES

- UML es una herramienta esencial para la **documentación y diseño de sistemas de software**.
- Los **diagramas de casos de uso** permiten entender cómo interactúan los usuarios con el sistema y sirven como base para el levantamiento de requisitos.
- El uso de UML favorece la **comunicación** en los equipos de desarrollo, reduce riesgos y facilita la implementación de sistemas de calidad.
- Los diagramas más utilizados (casos de uso, clases, secuencia y actividades) ofrecen una visión clara tanto de los requisitos como de la estructura y funcionamiento del sistema.

REFERENCIAS

SENA. (s.f.). *Gestión de requisitos* [Material de clase, documento proporcionado por la institución].

