

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

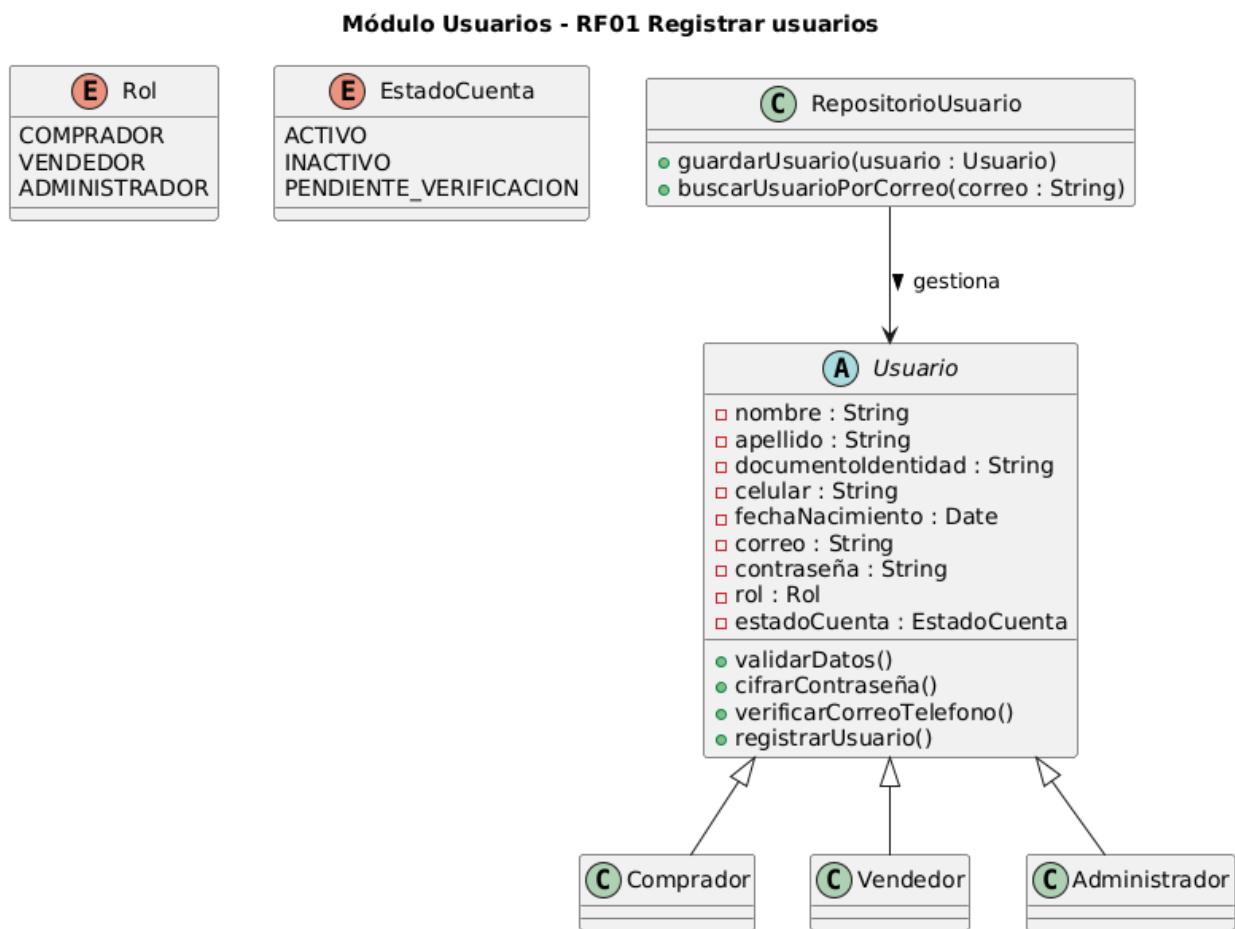
Fecha: 5 de octubre 2025

INTRODUCCION

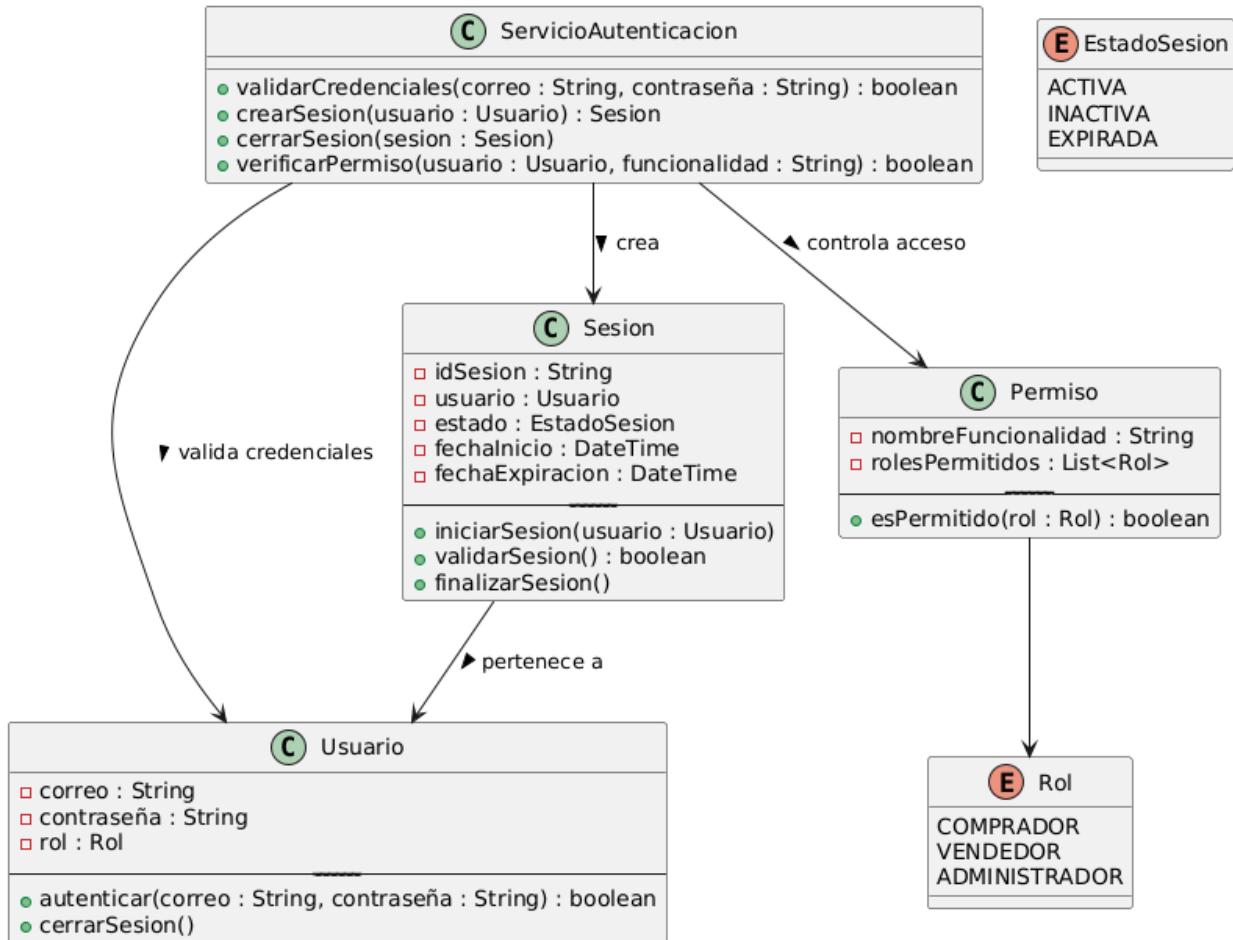
En el desarrollo de software orientado a objetos, es fundamental comprender la estructura y organización de los elementos que conforman un sistema. Para ello, los diagramas de clases (DC) desempeñan un papel esencial, ya que permiten representar de manera visual las clases, sus atributos, métodos y las relaciones que existen entre ellas. Este tipo de diagrama, perteneciente al lenguaje UML (Unified Modeling Language), proporciona una visión detallada y estructurada del modelo lógico del sistema, facilitando la comunicación entre los equipos de análisis, diseño y desarrollo.

A través de su uso, es posible identificar las responsabilidades de cada clase, definir correctamente las interacciones y garantizar una base sólida para la implementación del código. En este documento se explican los conceptos principales del diagrama de clases, su importancia, las partes que lo componen, los tipos de relaciones (asociación, agregación y composición) y cómo se representan, acompañados de ejemplos y conclusiones que refuerzan su utilidad en el proceso de ingeniería de software.

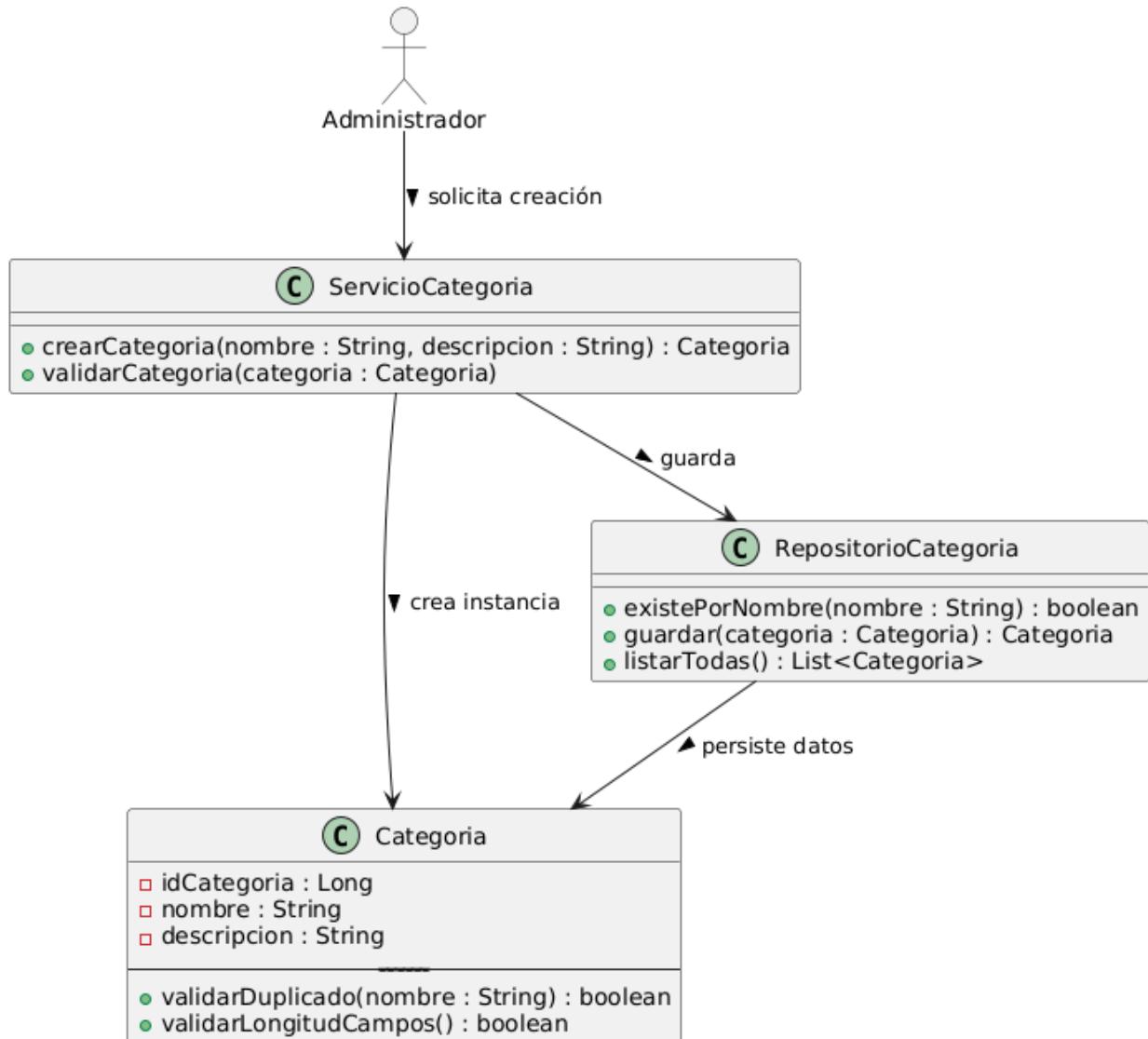
DIAGRAMAS



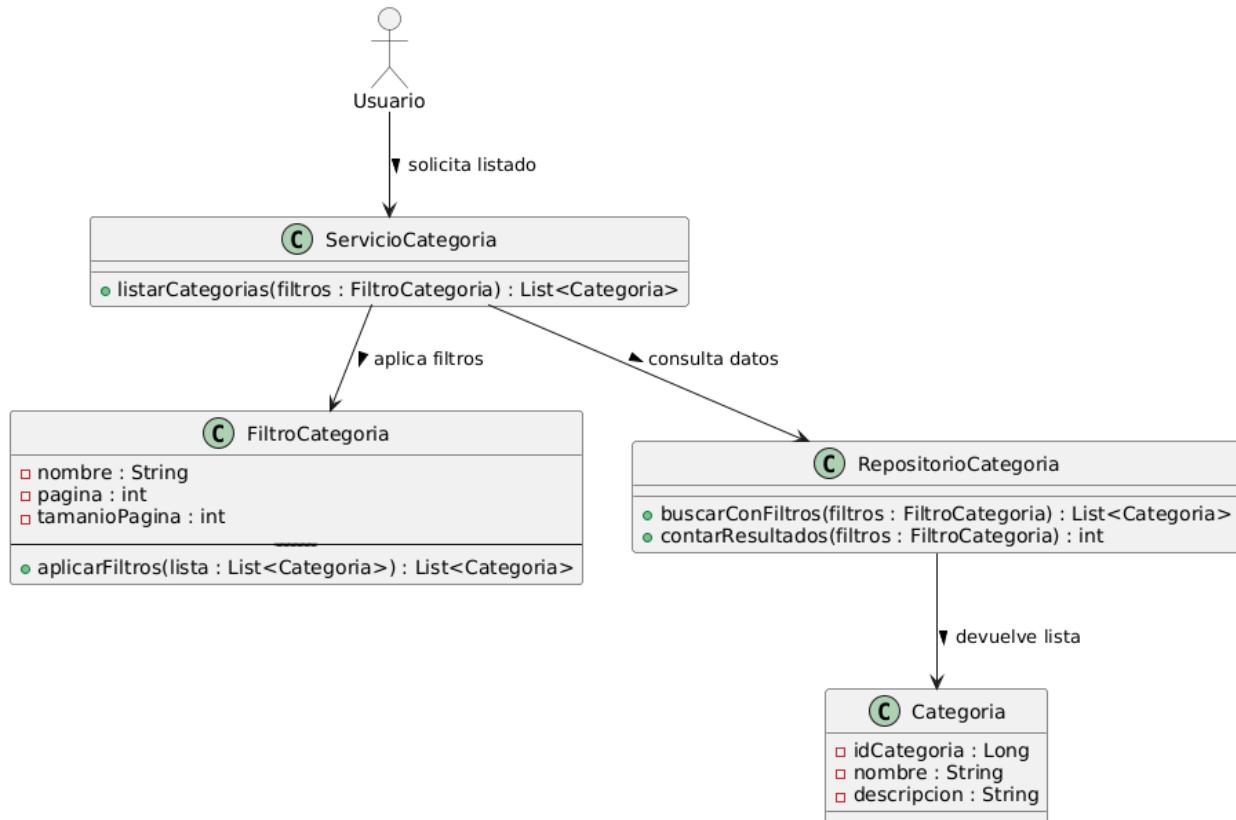
Módulo Seguridad / Usuarios - RF02 Autenticación y control de permisos por rol



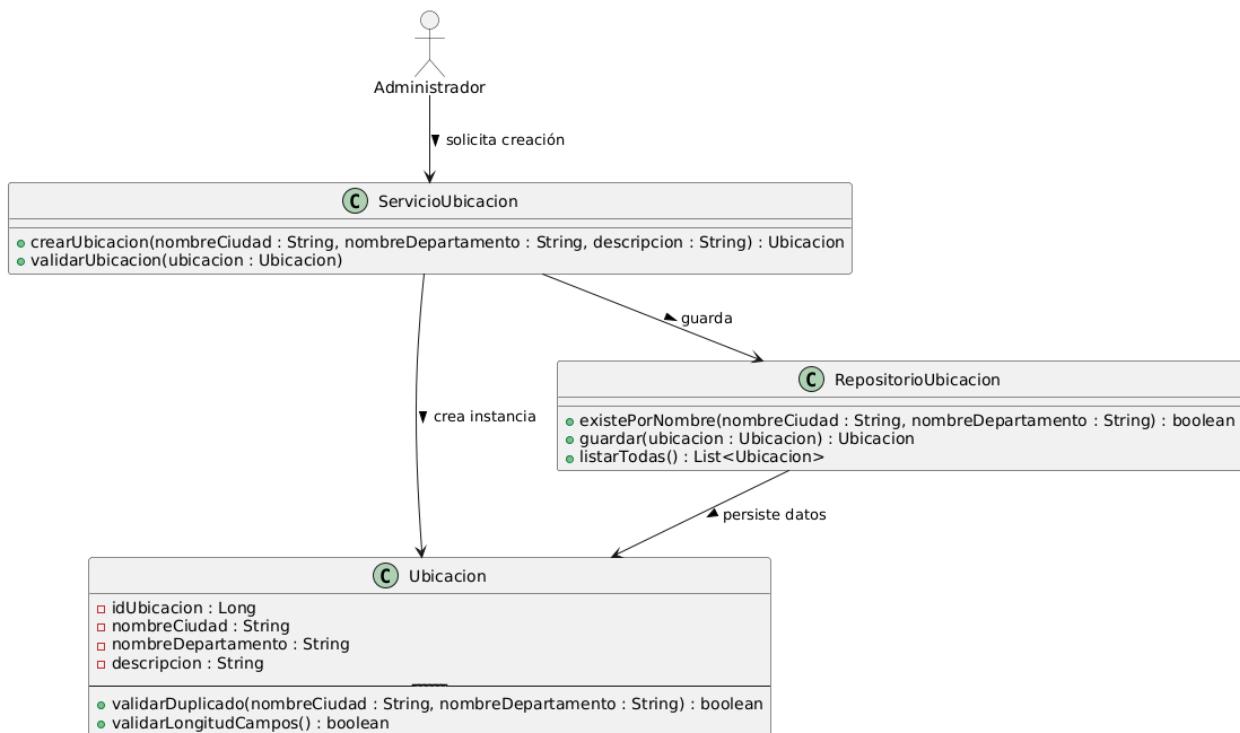
Módulo Inmuebles / Categorías - RF03 Crear categorías de inmuebles



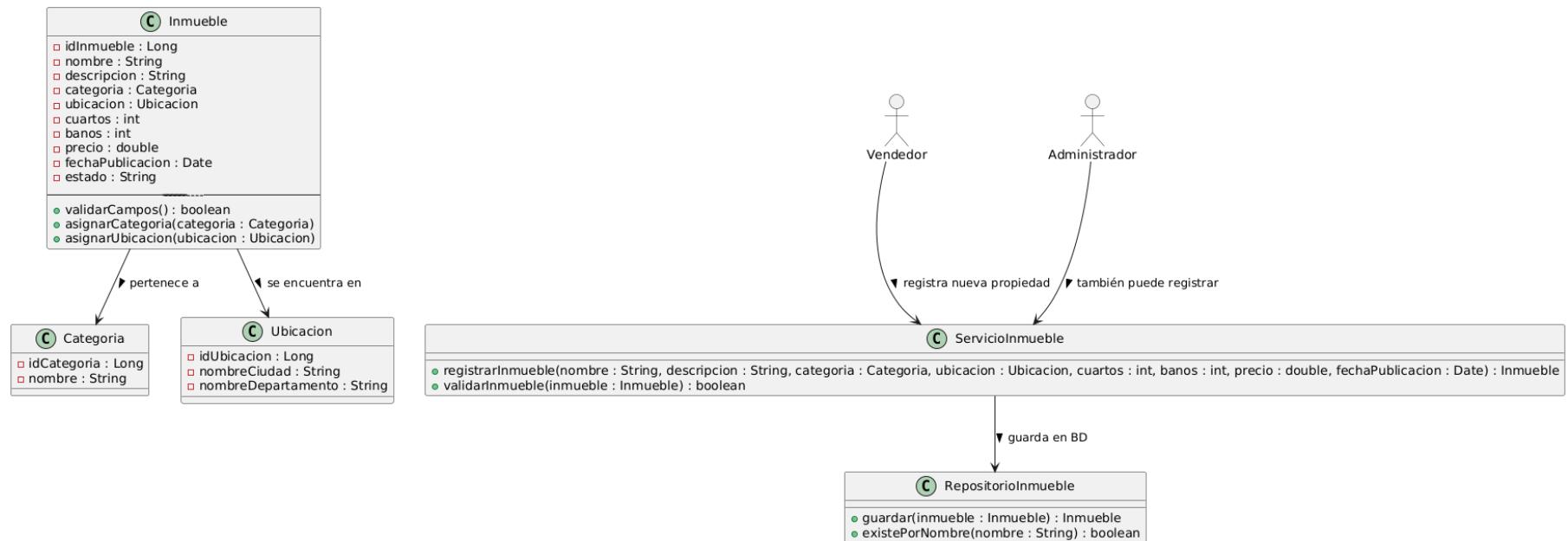
Módulo Inmuebles / Categorías - RF04 Listar categorías de inmuebles



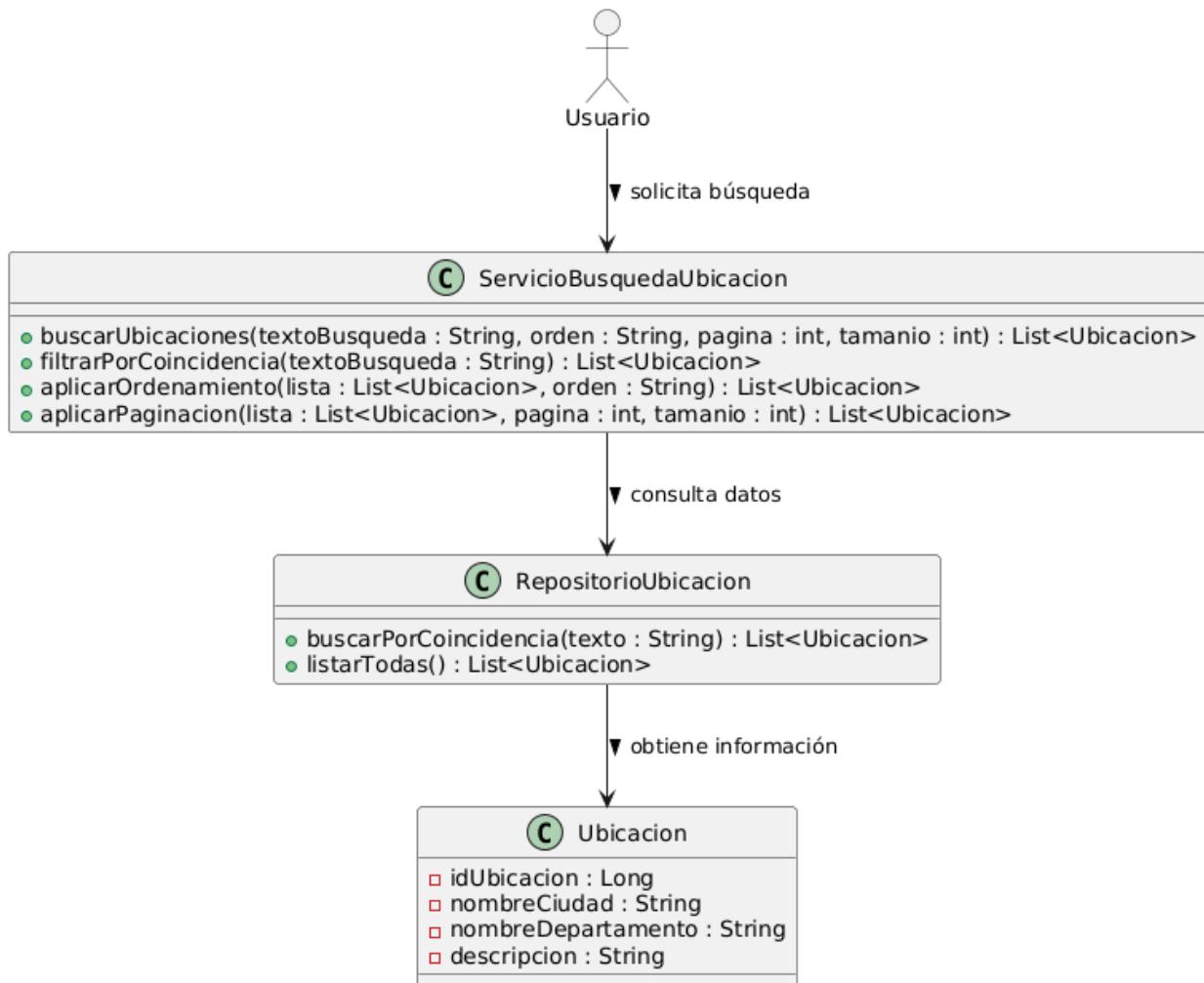
Módulo Inmuebles / Ubicaciones - RF05 Crear ubicaciones



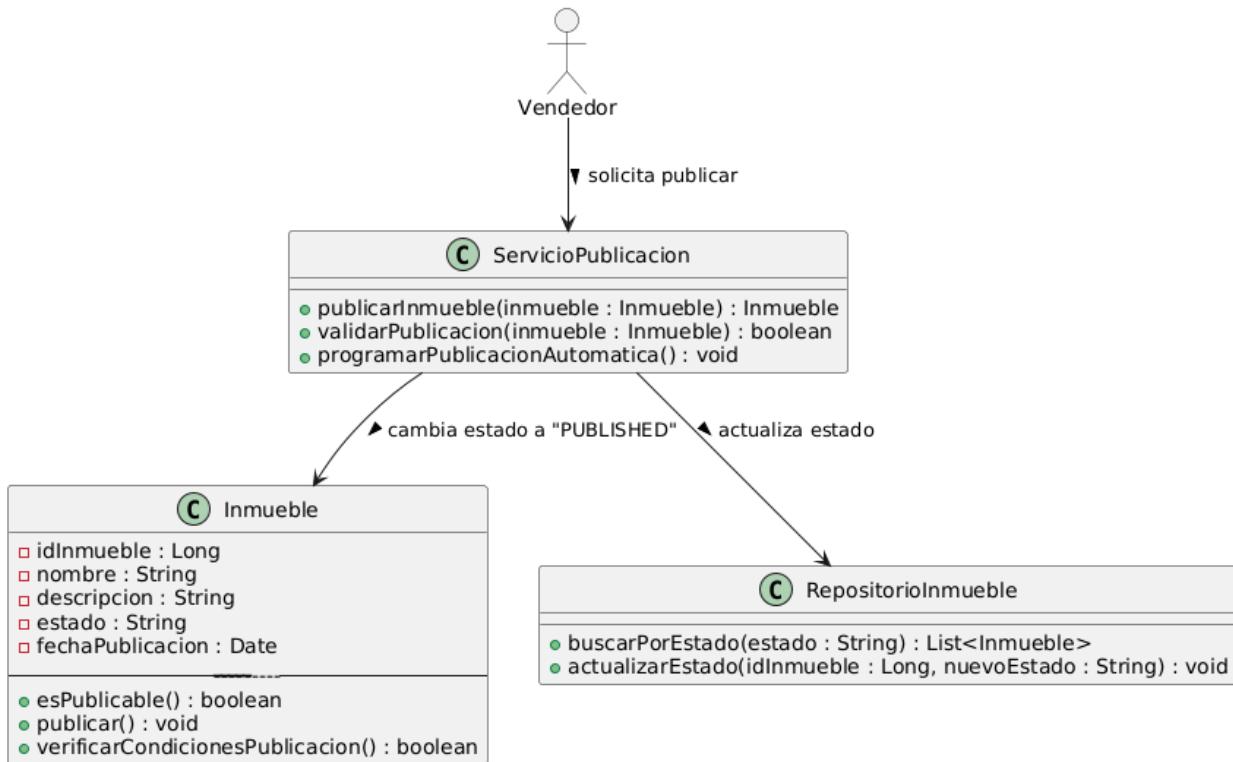
Módulo Inmuebles - RF07 Registrar nuevas propiedades



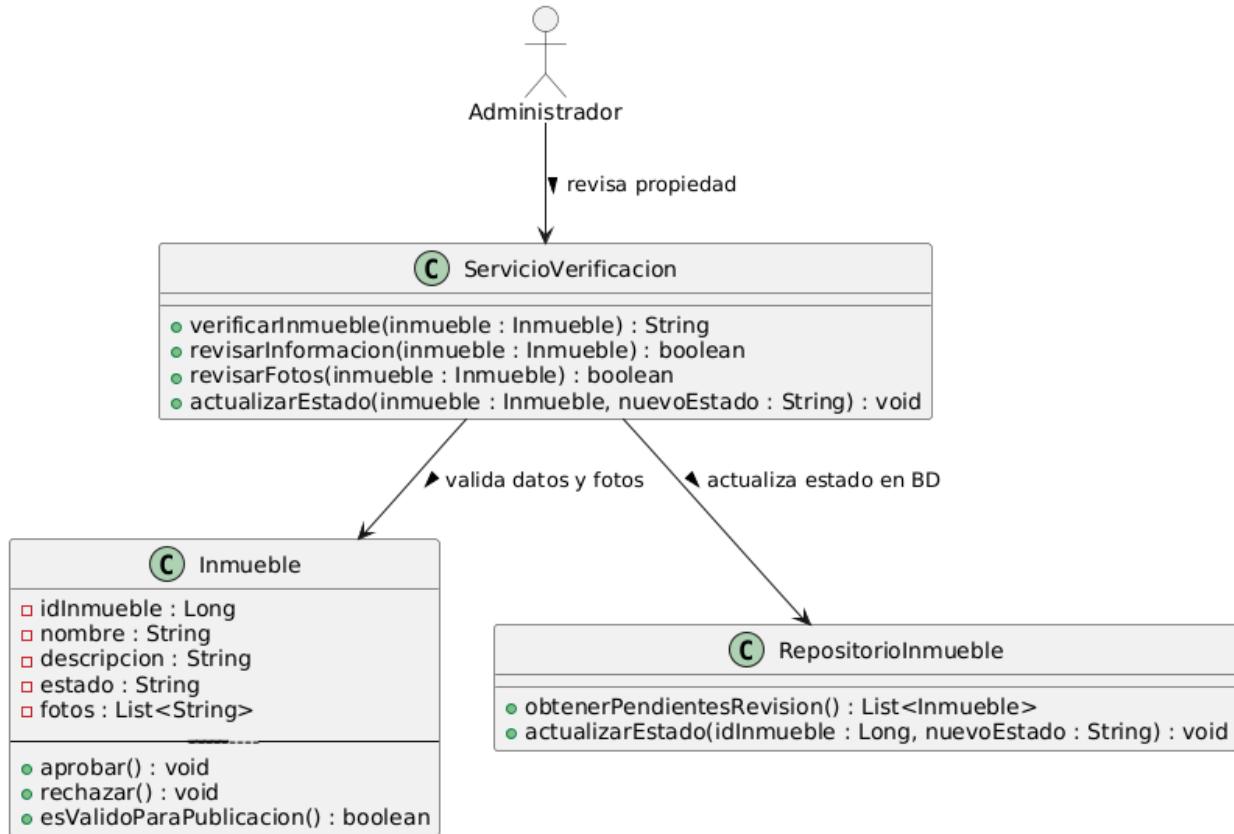
Módulo Inmuebles / Ubicaciones - RF06 Buscar ubicaciones



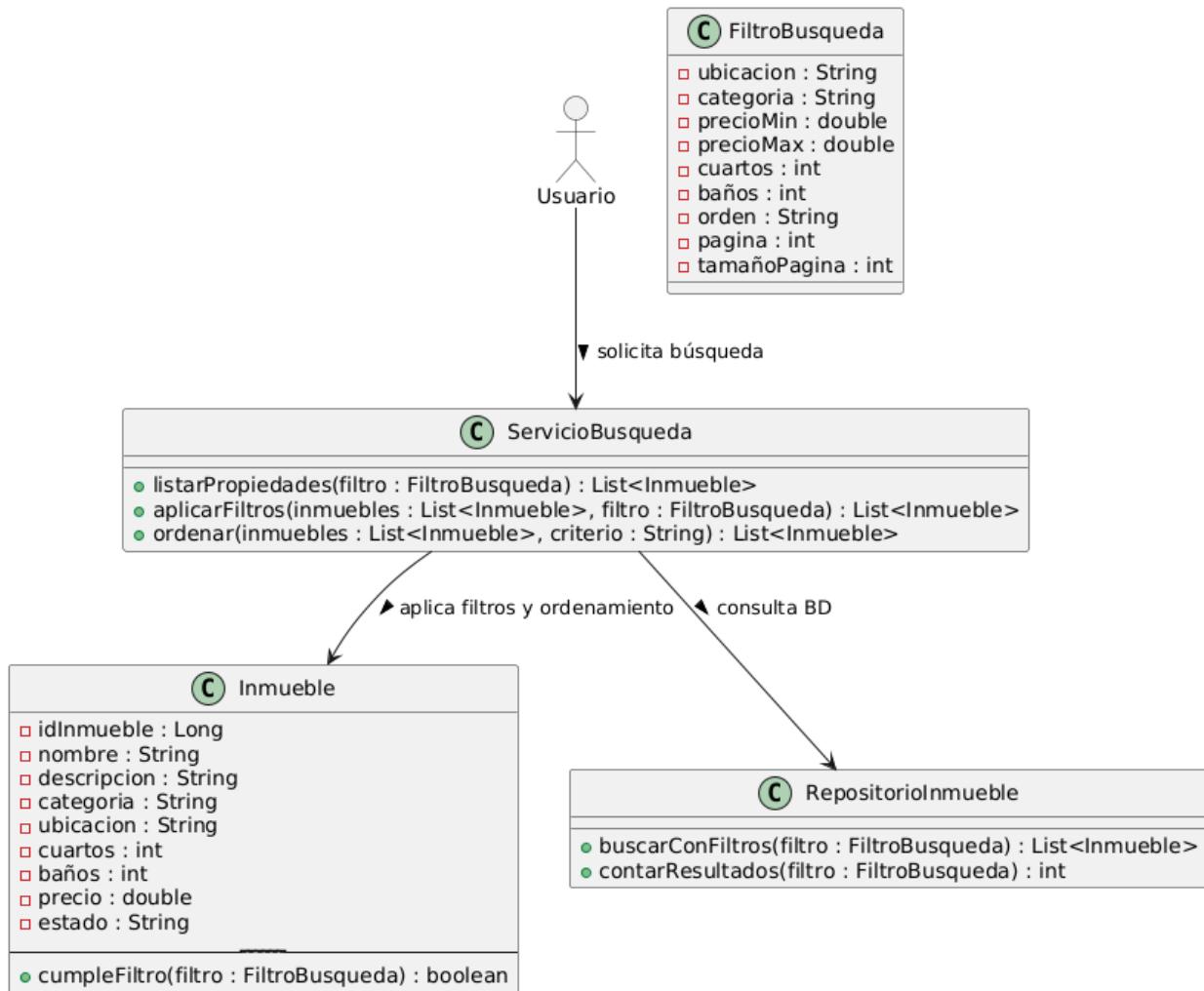
Módulo Inmuebles - RF08 Publicar propiedades



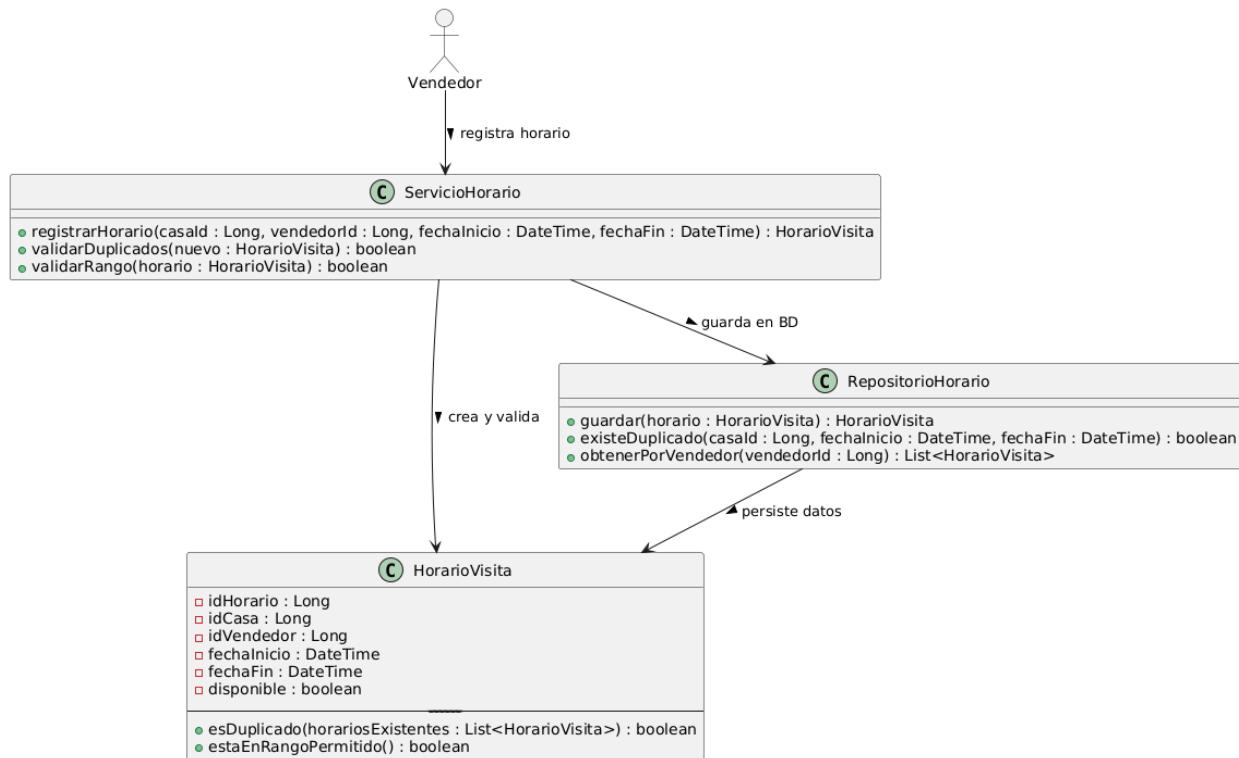
Módulo Control de Calidad - RF09 Verificar propiedades antes de su publicación



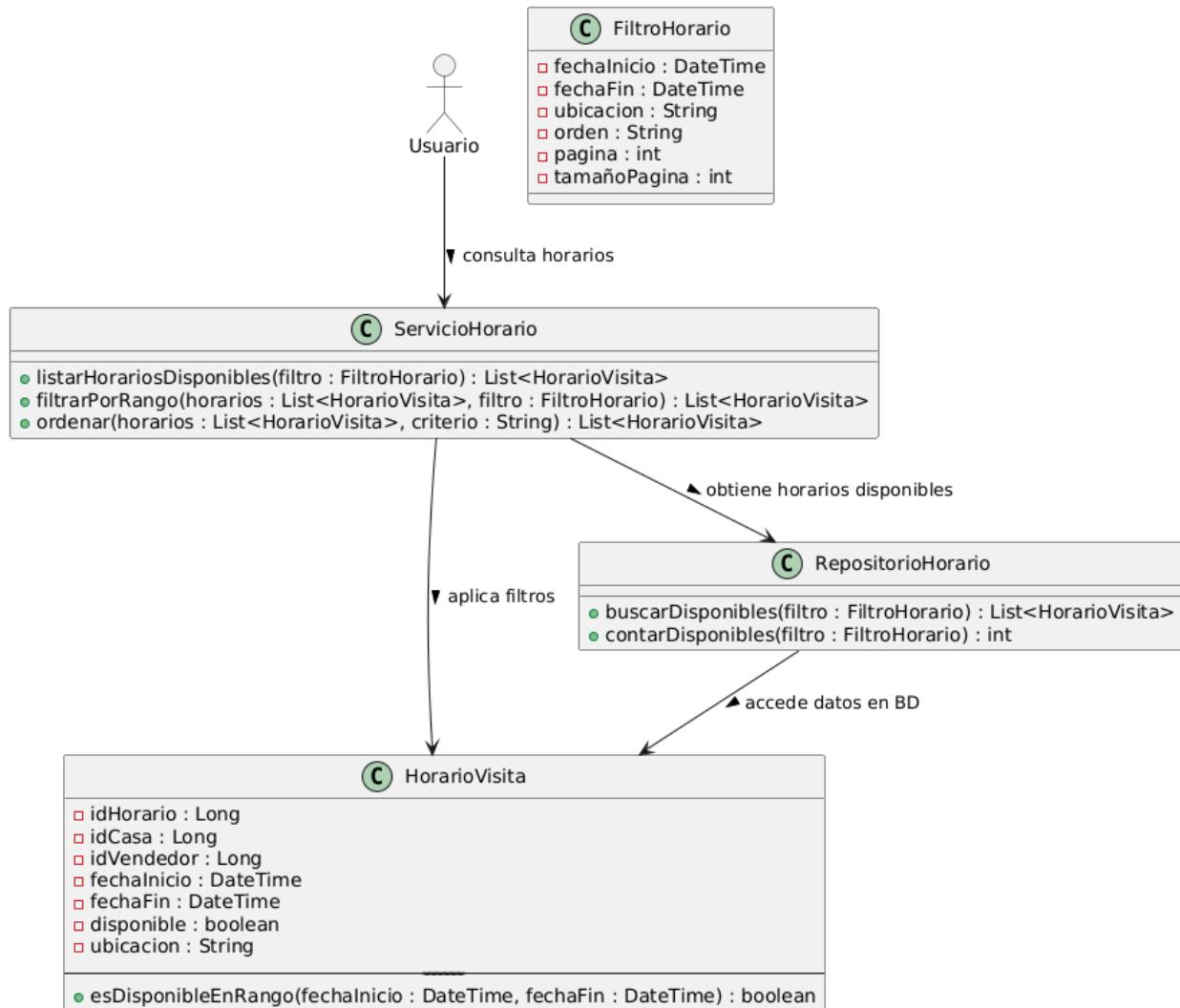
Módulo Inmuebles / Búsqueda - RF10 Listar propiedades



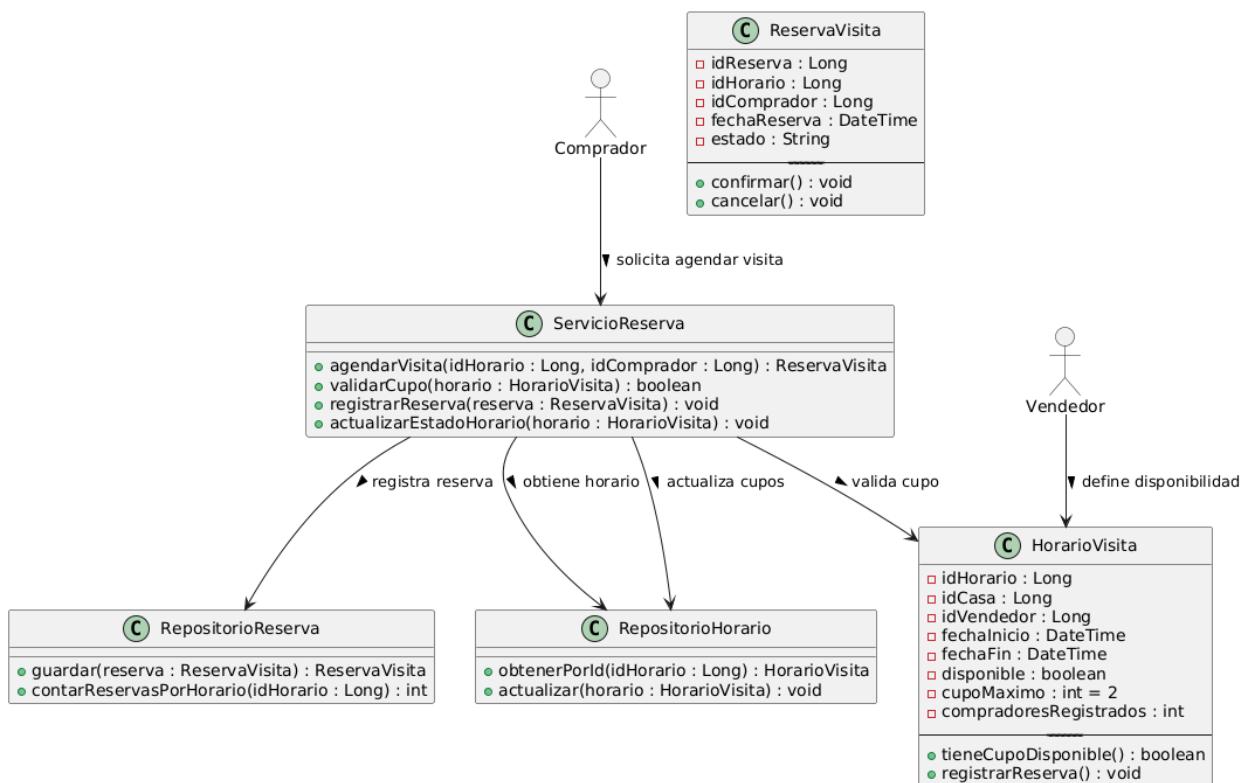
Módulo Reservas / Horarios - RF11 Disponibilizar horarios de visitas



Módulo Reservas / Horarios - RF12 Listar horarios de visitas disponibles



Módulo Reservas / Visitas - RF13 Agendar visitas



¿QUÉ ES UN DIAGRAMA DE CLASES?

Un Diagrama de Clases (DC) es un tipo de diagrama estructural del lenguaje UML (Unified Modeling Language) que muestra la estructura estática del sistema, representando las clases, sus atributos, métodos (operaciones) y las relaciones entre ellas.

En otras palabras, es una fotografía del modelo lógico del sistema, donde se visualizan los objetos que lo componen y cómo interactúan entre sí.

¿POR QUÉ ES IMPORTANTE?

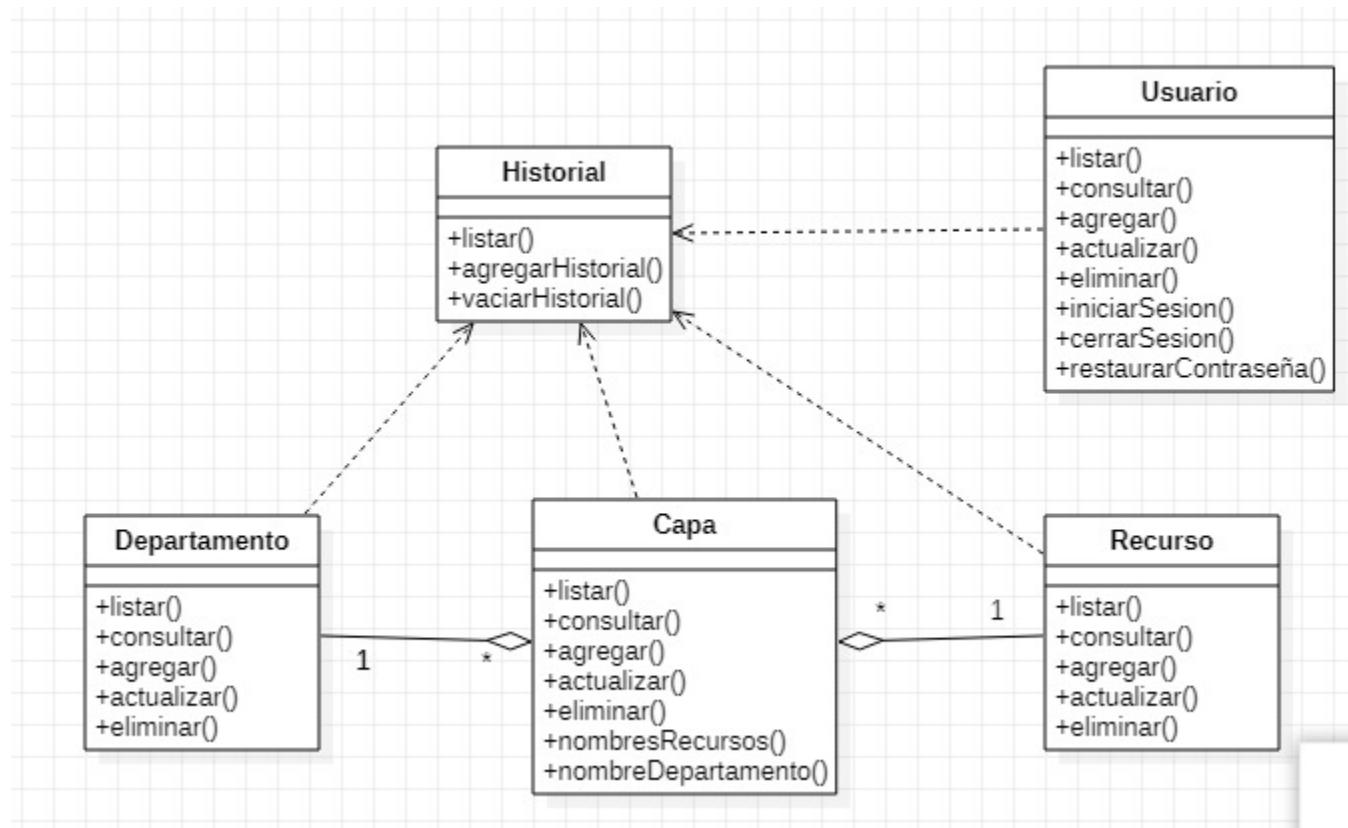
1. El Diagrama de Clases es fundamental en el análisis y diseño de software porque:
2. Permite visualizar la arquitectura del sistema antes de programarlo.
3. Facilita la comunicación entre analistas, desarrolladores y clientes.
4. Identifica responsabilidades y relaciones entre los elementos del dominio.
5. Sirve de base para generar código (por ejemplo, en lenguajes orientados a objetos como Java, C# o Python).
6. Evita ambigüedades y errores de interpretación durante el desarrollo.

En resumen, ayuda a planificar, documentar y mantener sistemas más organizados y coherentes.

PARTES DE UN DIAGRAMA DE CLASES

Cada clase en UML se representa como un **rectángulo dividido en tres secciones**: la primera para el nombre de la clase, la segunda para los atributos y la tercera para los métodos. Además, las relaciones entre clases.

- El signo cuadrado rojo (-) indica **privado**.
- El signo el circulo verde (+) indica **público**.



¿QUÉ ES UNA CARDINALIDAD?

La **cardinalidad** (también llamada **multiplicidad**) indica **cuántos objetos** de una clase pueden estar **relacionados con objetos de otra clase**.

Símbolo	Significado	Ejemplo
1	Uno a uno	Un usuario tiene un perfil
0..1	Cero o uno	Un pedido puede tener o no un cupón
* o 0..*	Muchos	Un cliente puede hacer muchos pedidos
1..*	Uno o más	Un profesor dicta una o más clases

TIPOS DE RELACIONES: ASOCIACIÓN, AGREGACIÓN Y COMPOSICIÓN

1. Asociación

Representa una relación simple entre dos clases.

Por ejemplo, un Cliente realiza un Pedido.

2. Agregación

Es una relación de **todo-parte débil**, donde una clase contiene a otra, pero **la parte puede existir sin el todo**.

Ejemplo:

Un **Departamento** tiene varios **Empleados**, pero si el departamento se elimina, los empleados pueden seguir existiendo.

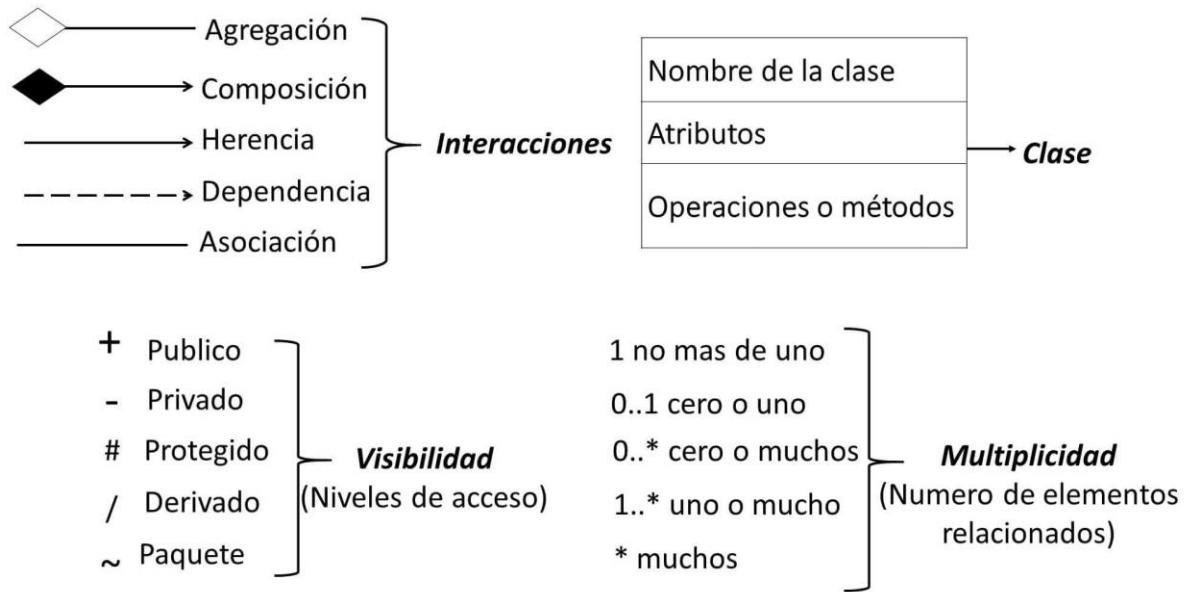
3. Composición

Es una relación de **todo-parte fuerte**, donde si el todo desaparece, las partes también.

Ejemplo:

Una **Casa** tiene **Habitaciones**, pero si se elimina la casa, también desaparecen sus habitaciones.

Elementos y símbolos en los diagramas de clases UML



CONCLUSIÓN

Los diagramas de clases son una herramienta fundamental dentro del modelado orientado a objetos, ya que permiten representar de manera visual y estructurada la arquitectura lógica de

un sistema. A través de ellos, se pueden identificar las entidades principales, sus atributos, operaciones y las relaciones que las conectan, facilitando la comprensión del funcionamiento interno del software antes de su implementación. Además, estos diagramas ayudan a mejorar la comunicación entre analistas, diseñadores y desarrolladores, garantizando coherencia entre los requerimientos funcionales y el diseño técnico. El uso adecuado de elementos como la cardinalidad, la asociación, la composición y la agregación permite reflejar con precisión las dependencias y responsabilidades entre las clases, asegurando un modelo sólido, escalable y mantenable. En síntesis, el diagrama de clases constituye una base esencial para construir sistemas bien estructurados y alineados con los principios de la ingeniería de software.

REFERENCIAS

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