UNIVERSIDAD MAYOR DE

SAN MARCOS

Universidad del Perú, DECANA DE AMÉRICA.

*FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA*

DISEÑO DEL SISTEMA

|  |  |  |  |
| --- | --- | --- | --- |
| Curso | : | Gestión de la Configuración de Software |  |
| Profesor | : | Lenis Rossi Wong Portillo |  |
| Escuela Profesional | : | Ingeniería de Software |  |
| Alumnos | : | Aguilar Salazar, Edwin Ccari | 18200323 |
|  |  | Antúnez Palomino, Kori Xiomara | 18200110 |
|  |  | Gomez Cavero, Mishell | 18200267 |
|  |  | Palomino Gutierrez, Erick Victor | 18200283 |
|  |  | Sandoval Salinas, Anthony Kevin Jerremy | 18200294 |
|  |  | Suarez Herandez, Kenny Joel | 18200102 |
|  |  | Vilca Daza, Diego Percy | 18200128 |

**Lima – Perú**

**Noviembre,2020**

**ÍNDICE**

[**Introducción**](#_jsi9gt7odeye) **3**

[**Representación arquitectónica**](#_szt6l6ruuupo) **4**

[**Metas y limitaciones arquitectónicas**](#_xlmyxx6d8njc) **5**

[**Vistas de caso de uso**](#_4qhoc1vwnq2n) **6**

[**Vista lógica**](#_a7pgydtvysoy) **7**

[**Vista de proceso**](#_xpbtuou6gktu) **8**

[**Vista de implementación**](#_uhwby4jehwik) **9**

# Introducción

Con el objetivo de enrumbar a la organización con respecto al proyecto es necesario realizar un documento donde se detallen los puntos de cómo será construido el sistema.

Este documento es , específicamente, el documento de diseño del sistema donde se detallan las el diseño arquitectónico del sistema, la abstracción de los objetos que interactúan en el sistema y sus asociaciones entre ellos. De esta forma se podrán apoyar no solo los desarrolladores del sistema, sino todo aquel relacionado con la realización de este proyecto.

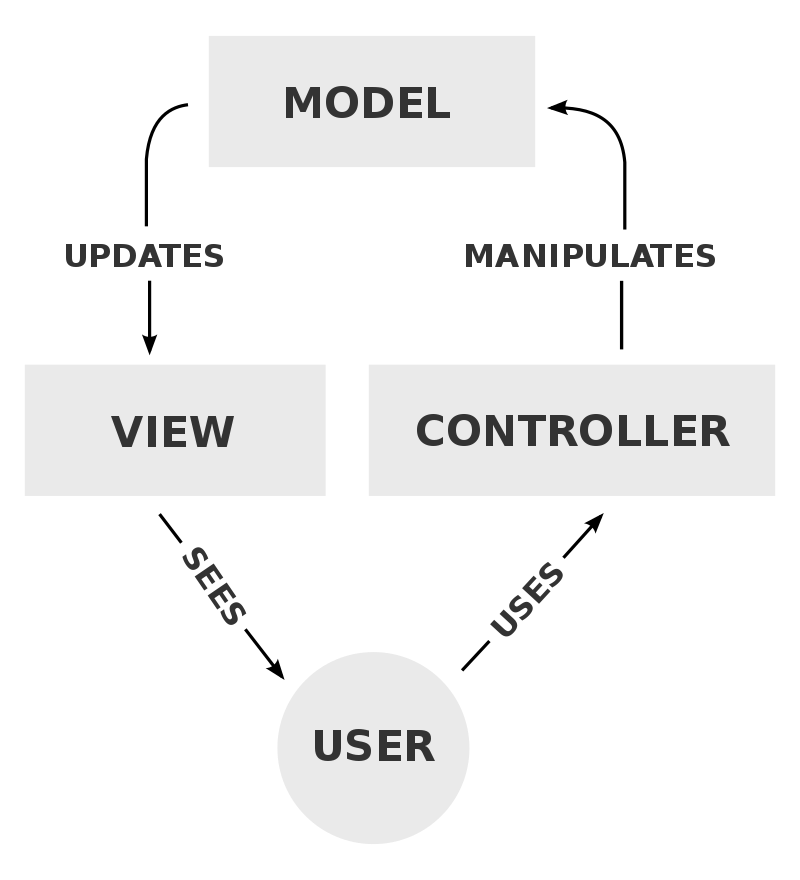
# Representación arquitectónica

En este apartado se hablará de los módulos principales que participarán en el sistema.

Debido a que el sistema es una aplicación web, la arquitectura estará fundamentada en el patrón modelo-vista-controlador. La idea de este patrón es dividir el sistema en tres componentes principales:

**El modelo:** representa a la información con la que el sistema va a operar, es decir, gestiona el acceso directo a la información (en este caso a una base de datos) y actualiza lo mostrado a través del componente “vista”, además las peticiones de acceso o manipulación de información llegan a través del componente “controlador”.

**El controlador:** es el componente encargado de todas las funcionalidades internas de la aplicación. Es el responsable de la comunicación entre la “vista” y el “modelo”, en otras palabras, es el intermediario entre ambos componentes.

**La Vista:** Presenta al modelo de forma que pueda interactuar con el usuario, es decir, es la interfaz de usuario.

# Metas y limitaciones arquitectónicas

Los objetivos del patrón escogido son los de mantener un orden lógico de peticiones, ejecución y extracción de información. De esta forma tenemos como metas, al utilizar este patrón, las siguientes ideas:

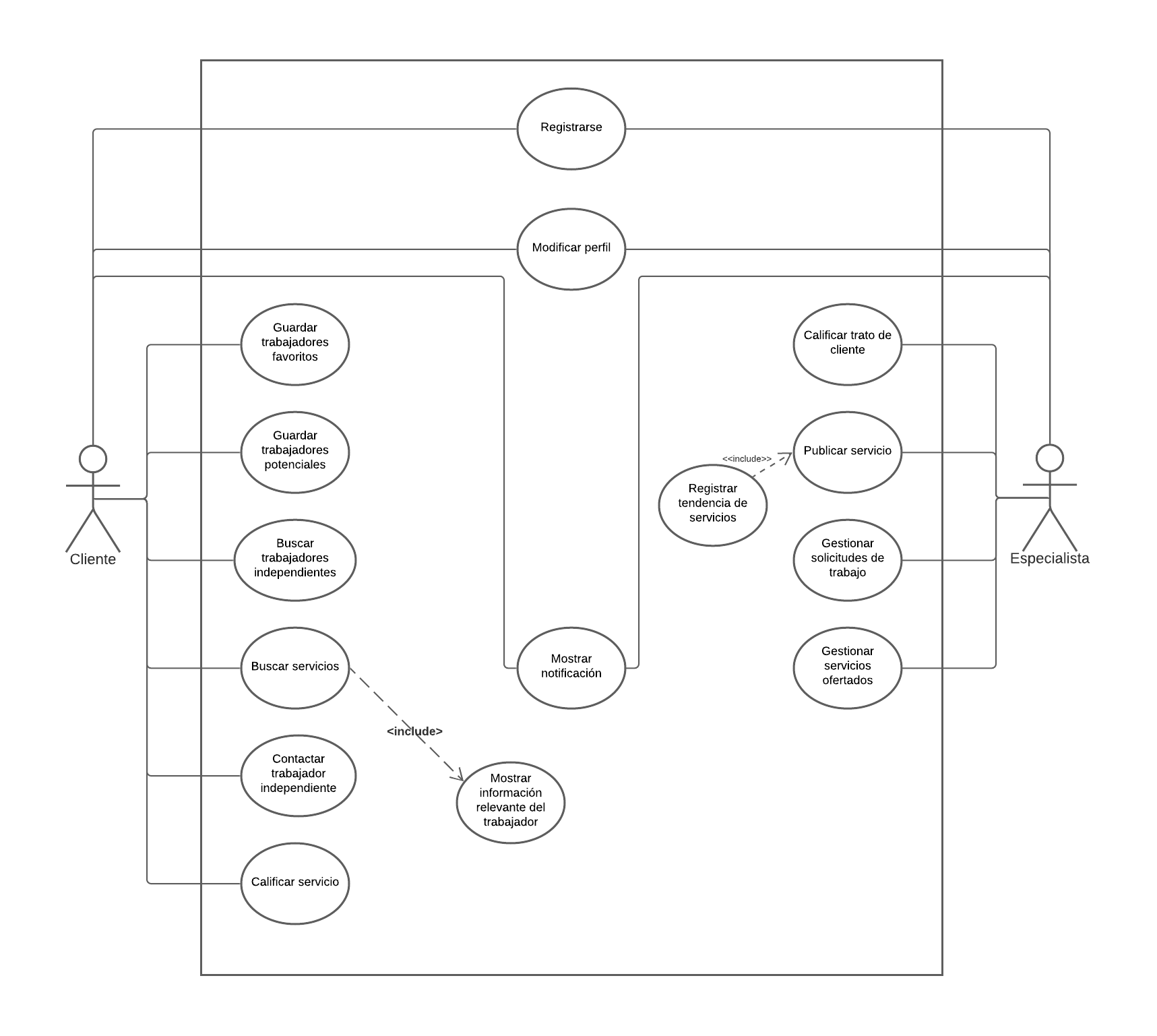
* Al dividir en tres componentes distintos buscamos modularizar el trabajo de desarrollo en 3 áreas de trabajo: diseño de vistas, diseño lógico o algorítmico y diseño relacional o de base de datos. Esto nos permite trabajar mejor de forma separada el proyecto y así lograr la repartición de tareas.
* Otra meta es la de conseguir un mejor manejo de errores a través de este patrón, ya que la separación de módulos nos permite identificarlos de forma más fácil.
* Finalmente se tiene como meta producir un sistema de alta escalabilidad, que es el objetivo de toda organización al ver el éxito de su producto.

Con respecto a las limitaciones que nos da este patrón son las siguientes:

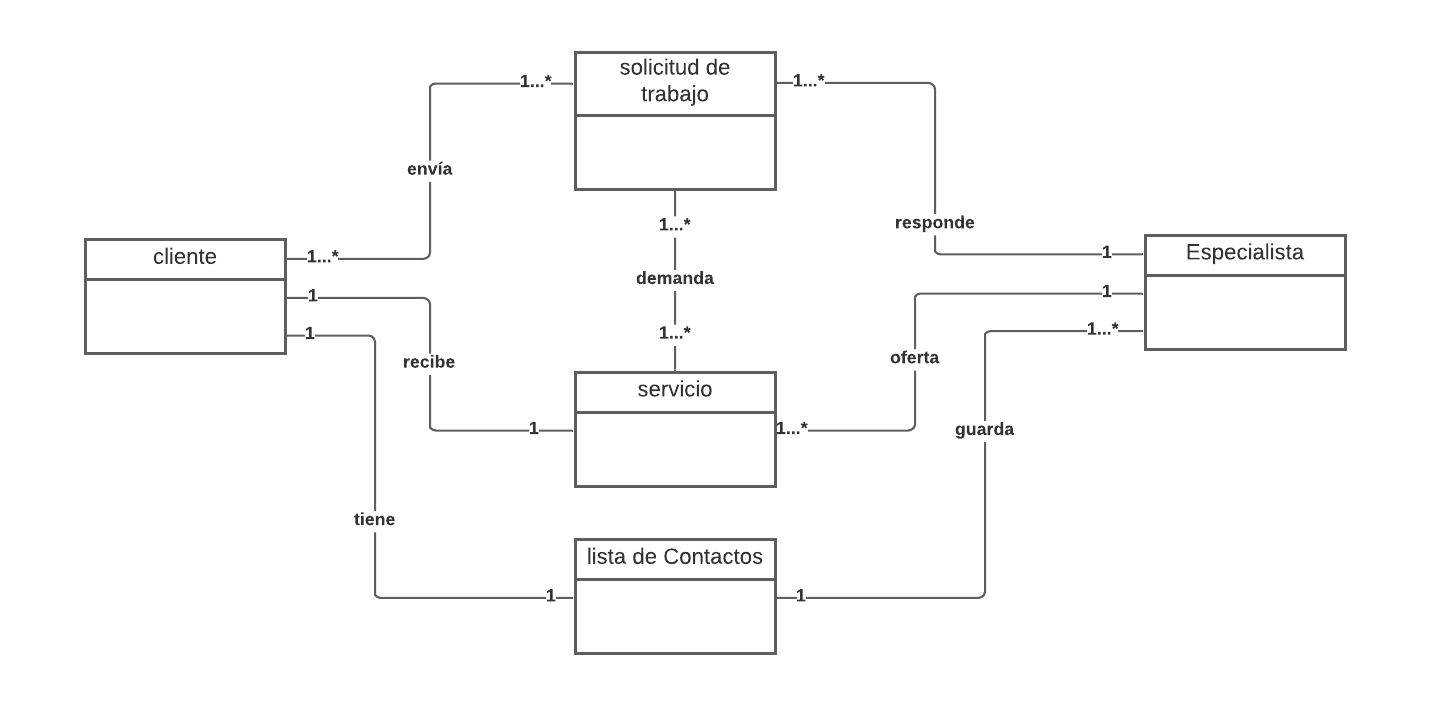
* La cantidad de archivos a mantener y trabajar se ve incrementada considerablemente.
* La separación en capas de este patrón hace más complejo el sistema.

# Modelo de caso de uso

Una vez elicitados los requerimientos del sistema estos serán organizados en casos de uso según el modelo de desarrollo RUP obteniendo así un modelo general de casos de uso del sistema, representado por la siguiente figura:

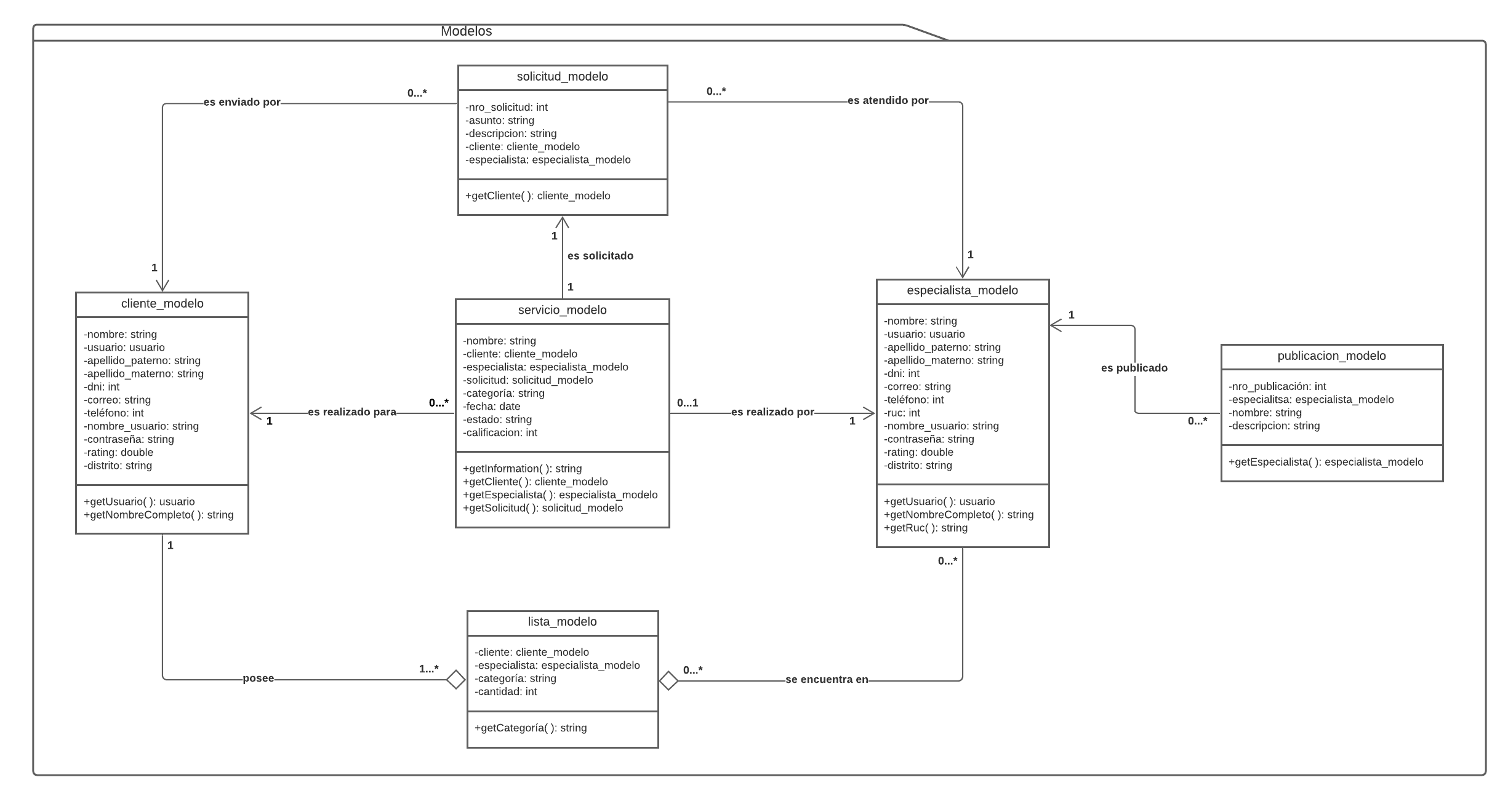
****

# Modelo lógico

Para poder abstraer de la realidad las clases u objetos que interactúan dentro del sistema realizamos un modelo de dominio donde inspeccionamos en el proceso real que cubre nuestro sistema qué entidades, eventos o relaciones existen en dicho proceso real y los diagramamos en el siguiente modelo:

Una vez abstraídos las entidades de la realidad las analizamos y transformamos en clases funcionales para el sistema basadas en el patrón modelo, vista y controlador empaquetadas en dichos módulos:

**Clases modelo:**

****