



# Sistema informático para la Gestión de Conocimiento

## Integrantes:

Ali, Lucas

Casavalle, Leandro

Córdoba, Leandro

Cruz, Sofía

Granero, Javier

Exposición de Proyecto

# Introducción

Diseño, documentación e implementación de una solución para un sistema informático de **Gestión de Conocimiento**.



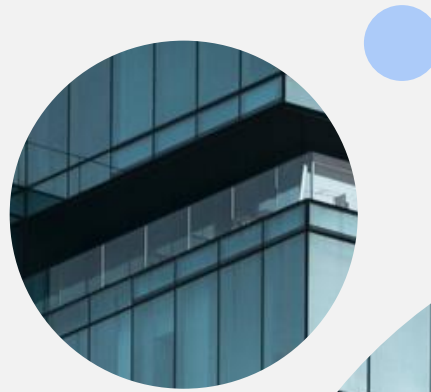
Diseño



Documentación



Implementación







# Diseño de la solución

# Etapas



## Análisis

- **Comprender** a fondo el problema e identificar los **requisitos**.

## Modelado

- Desarrollar **modelos** abstractos del sistema que sirvan como base para la implementación.

## Diseño

- Elaborar un **diseño** detallado que sirva como base para la implementación.

# Identificación de clases



# Requisitos

## Gestión de usuarios



- Registro de usuarios con **datos personales** (nombre, apellido, país), email y contraseña.
- Usuarios pueden realizar **preguntas y respuestas**.
- Valoración con "**me gusta**" a respuestas.
- Usuarios consultores pueden marcar respuestas como **aceptadas**.

# Clase diseñada

USUARIO	
- id: entero	
- <u>autonumerico: entero</u>	
- nombre: string	
- apellido: string	
- paisOrigen: string	
- email: string	
- contraseña: string	
- notificaciones: Lista[string]	
- likesRespuestas: lista[Publicacion]	
<hr/>	
+ getNombre(): string	
+ eliminarCuenta()	
+ contarRespuestasAceptadas(): entero	
+ getId(): entero	
+ crearPregunta(imagen: string, titulo: string, descripcion: string, tags: arreglo[string]): Pregunta	
+ crearRespuesta(pregunta: Pregunta, imagen: string, contenido: string): Respuesta	
+ listarInformacion()	
+ aceptarSolucion(pregunta: Pregunta, idRespuesta: entero)	
+ agregarRespuestaLikeada(respuesta: Respuesta)	
+ pertenece(idBuscado: entero): bool	
+ agregarNotificacion(mensaje: string)	

# Requisitos

## Gestión de preguntas ?

- Creación de preguntas con **título**, **descripción** y opción de adjuntar **imagen**.
- Registro de **fecha** de creación.
- Asociación de **tags** para búsquedas.
- Preguntas **inactivas** después de 6 meses sin respuestas.
- Posibilidad de marcar respuestas y cambiar estado a **solucionada**.
- **Múltiples respuestas** por pregunta.
- **Notificación** al usuario consultor al recibir respuesta.
- Sin notificación si la pregunta está **suspendida**.



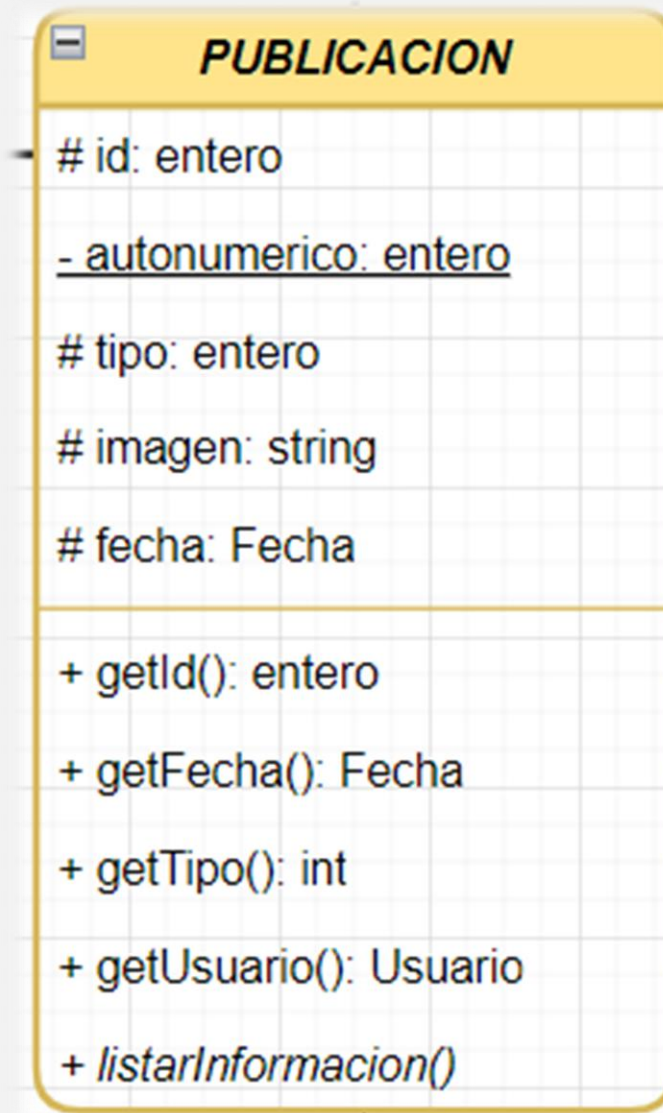
# Requisitos

## Gestión de respuestas

- Creación de respuestas con texto (**contenido**) e **imagen** opcional.
- Registro de **fecha** de creación.
- Respuestas pueden ser marcadas como **aceptadas**.
- Valoración con "**me gusta**".
- Posibilidad de listar respuestas por **cantidad** de "me gusta".

# Clase abstracta diseñada

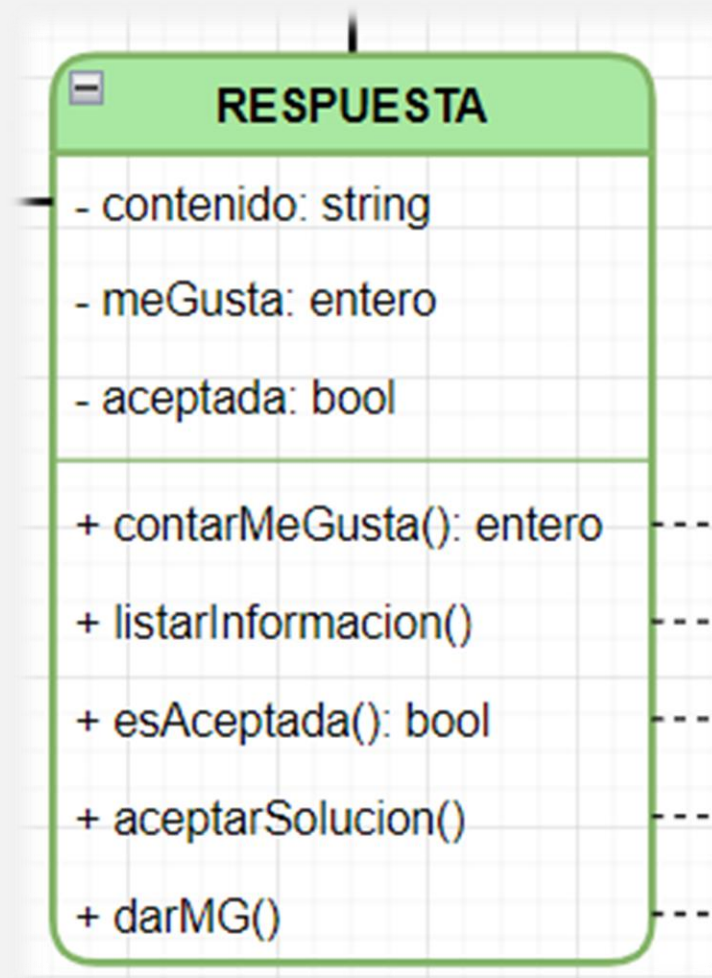
## Herencia



# Clase derivada diseñada



# Clase derivada diseñada





# Requisitos

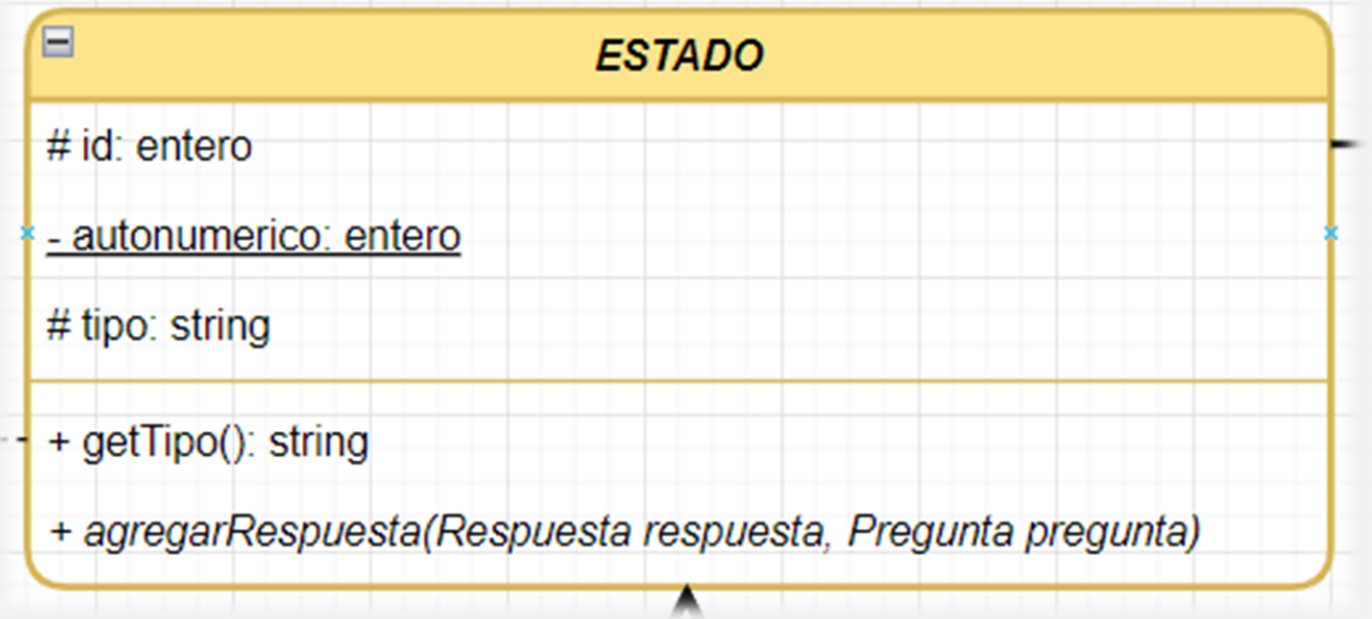
## Estados de las preguntas



- Estados: **activa**, **inactiva**, **solucionada**, **suspendida**.
- Automatización del cambio de estado:
  - **Inactiva**, después de 6 meses sin respuestas.
  - **Activa**, si está inactiva y recibe respuesta.
  - **Suspendida**, si el consultor elimina su cuenta.
  - **Solucionada**, si el consultor marca una respuesta como aceptada.

# Clase abstracta diseñada

## Herencia State Pattern



# Clases derivadas diseñadas



# Requisitos

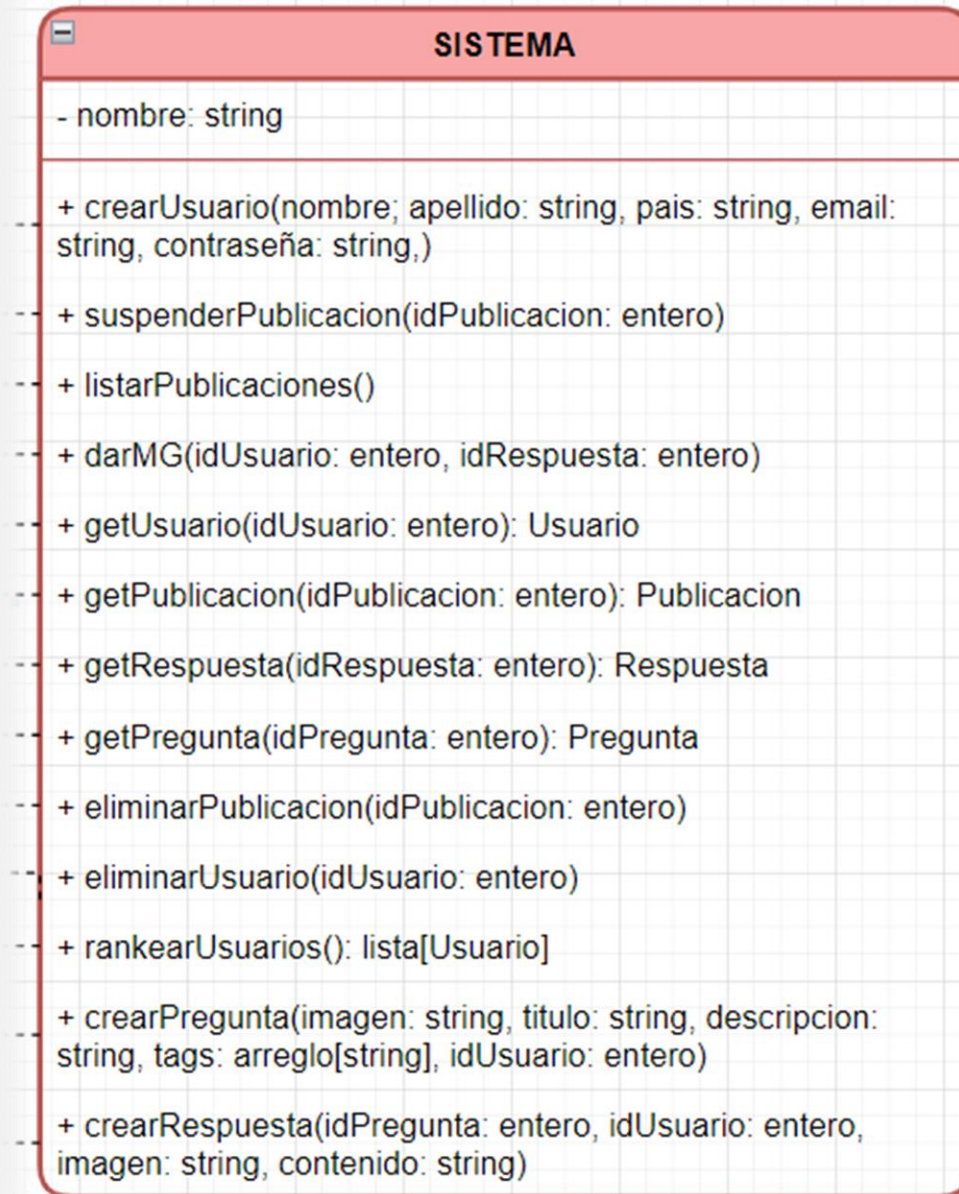
## Interacción, valoración, búsqueda



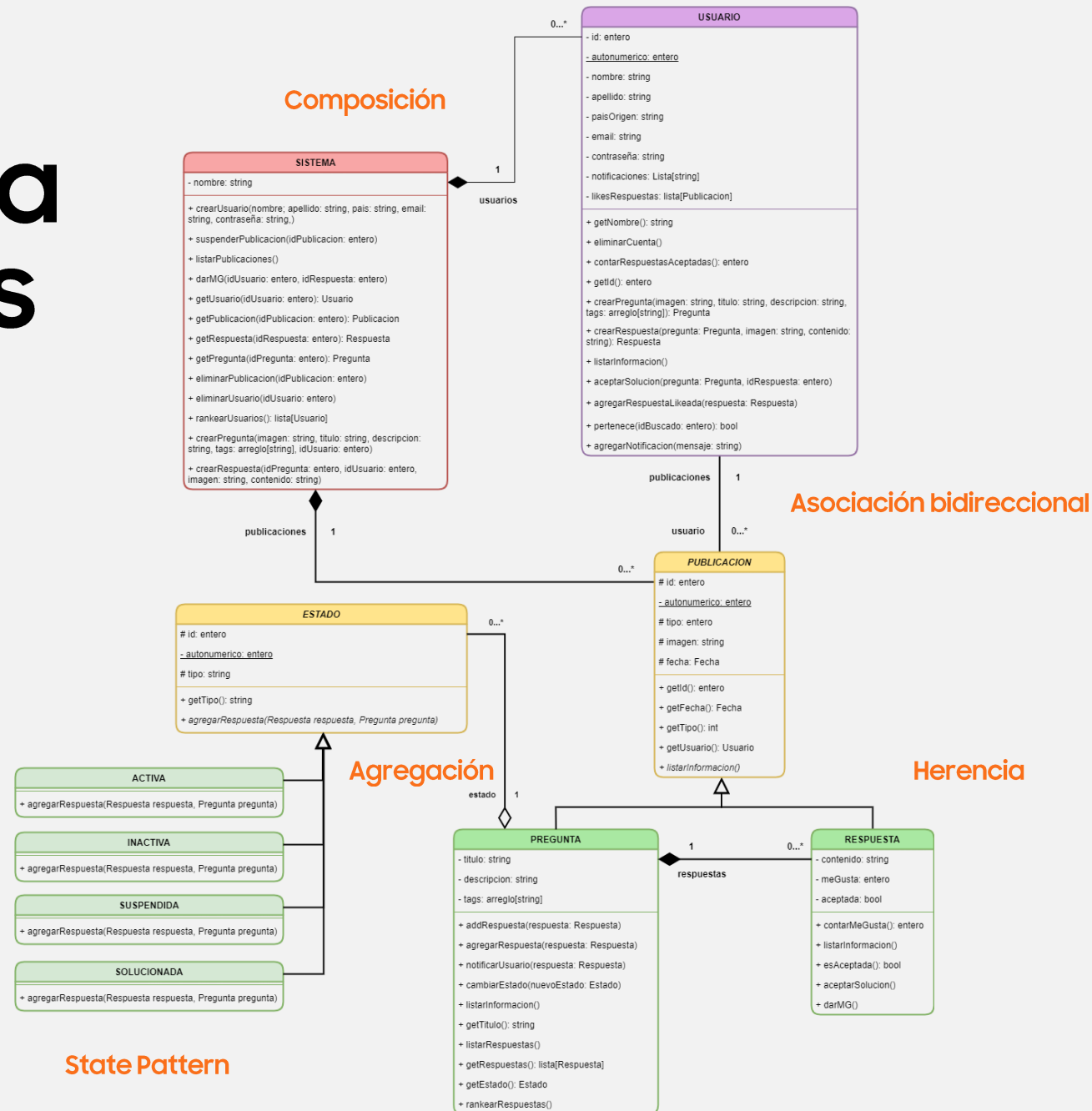
- Usuarios pueden **preguntar** y **responder**.
- **Notificación** automática al consultor cuando alguien responde.
- Preguntas inactivas pueden ser respondidas y vuelven al estado **activo**.
- Usuarios **rankeados** por respuestas **aceptadas**.
- Respuestas **rankeadas** por **cantidad** de "me gusta".
- Respuesta **aceptada** ocupa el primer lugar si la pregunta está solucionada.



# Clase principal diseñada



# Diagrama de clases (UML)





```
cout << Gracias << endl;
```