

**Diseño de Aplicaciones 2**  
Primer Obligatorio



Cristian Palma 208443 Federico Alonso 182999

**Descripción del diseño**

**Grupo N6A**

Repositorio: https://github.com/ORT-DA2/182999\_208443

**Índice**

[1. Descripción del diseño 2](#_Toc83924348)

[1.1 Suposiciones y decisiones de diseño 2](#_Toc83924349)

[1.2 Descripción general del diseño 4](#_Toc83924350)

[1.3 Diagrama general de paquetes 4](#_Toc83924351)

[1.4 Diagrama de componentes 4](#_Toc83924352)

[1.5 Asignación de responsabilidades 4](#_Toc83924353)

[1.6 Mecanismos de inyección de dependencias, fabricas, patrones y principios de diseño 4](#_Toc83924354)

[1.7 Descripción del mecanismos de acceso a datos utilizado 4](#_Toc83924355)

[1.8 Descripción del manejo de excepciones 4](#_Toc83924356)

[1.9 Diagrama de secuencia de … 4](#_Toc83924357)

# Descripción del diseño

## Suposiciones y decisiones de diseño

Utilizando como base la letra del obligatorio y las consultas del foro se tomaron las siguientes decisiones:

1. Para los usuarios se creó una clase abstracta **Usuario**, y cada uno de los roles existentes (administrador, desarrollador, tester) corresponden a una clase que heredan de Usuario. Los atributos de la clase usuario son:
2. Id
3. Nombre
4. Apellido
5. Nombre de usuario
6. Contraseña
7. Email
8. Token
9. El usuario administrador se cargó directamente en la base de datos y para los usuarios desarrollador y tester se hizo la funcionalidad necesaria para darlos de alta utilizando una clase con esa responsabilidad llamada **GestorUsuario**.
10. Para los proyectos se creó una clase **Proyecto** con los siguientes atributos:
11. Id
12. Nombre
13. Lista de Incidentes
14. Lista de Desarrolladores
15. Lista de Testers
16. Para las funcionalidades necesarias para el mantenimiento de los proyectos, se creó una clase llamada **GestorProyecto**.
17. Para dar de alta y/o actualizar los desarrolladores y/o testers correspondientes a un proyecto, se debe enviar todo el listado en una sola acción, es decir por ejemplo que si tenemos 5 desarrolladores asociados al proyecto y deseamos agregar 1 desarrollador más, debemos incluir los que ya se encontraban antes asociados. Se piensa que a futuro cuando se tenga un front end, en la funcionalidad de update, los que ya se encuentren asociados al proyecto van a estar en la vista del usuario ya seleccionados. **//NO SERIA NECESARIO ELIMINAR // BUGS ASOCIADOS?**
18. Para registrar incidentes se creó una clase Incidente con los siguientes atributos:
19. Id
20. Nombre
21. Nombre Proyecto
22. Descripción
23. Versión
24. Estado del Incidente (puede ser activo o resuelto)
25. Desarrollador Id (para guardar cual desarrollador resolvió el incidente)
26. Para resolver el mantenimiento de los incidentes, se creó la clase **GestorIncidente**.
27. Se creó una clase **Validaciones**, que sirve para validar atributos de diferentes clases con las siguientes validaciones:
28. Validar largos de textos
29. Validar passwords
30. Validar emails
31. No se puede agregar a un proyecto un tester o desarrollador que previamente no haya sido dado de alta en la BD.
32. Cuando un desarrollador cambia el estado de un incidente a resuelto, automáticamente se la asigna a dicho desarrollador el atributo Desarrollador Id**. //FUNCIONALIDAD ESPECIFICA PARA CAMBIAR ESTADO DE BUG Y NO DENTRO DEL UPDATE**

## Descripción general del diseño

## Diagrama general de paquetes

## Diagrama de componentes

## Asignación de responsabilidades

## Mecanismos de inyección de dependencias, fabricas, patrones y principios de diseño

## Descripción de los mecanismos de acceso a datos utilizado

## Descripción del manejo de excepciones

## Diagrama de secuencia de …