CIFP Nº1 CEUTA

Proyecto Final de Grado

Gestor REI (Recursos E Inventario)

Álvaro García Salvatierra

2º Desarrollo Aplicaciones Web

CURSO 2024/2025

**Resumen:**

La idea principal de este proyecto es crear un sistema de inventariado, tareas, gestión de usuarios al mismo tiempo, la idea de este proyecto llega desde la poca optimización de estos procesos a nivel local en varias instituciones publicas. Este proyecto ayudaría a optimizar los recursos y la agilización de las incidencias.

**Palabras Clave:** Gestor, Inventario, Tareas, Inventariado, Técnicos, Usuarios, Gestión Administrativa, Incidencias

**Abstract:**

The main idea of this proyect was to create a Inventory system that could assign tasks to technicians and manage users all at the same time. This whole idea comes from the lack of optimization in public institutions. This proyect would help optimize the resources and overall facilitate troubleshooting.

**Keywords:** Gestor, Inventory, Tasks, Technician, Users, Administrative Gestion, Incidences, Troubleshooting, Manager, Manage

**Indice**

1. **Introducción**
2. **Objeto del proyecto**
3. **Objetivos**
4. **Alternativas**
5. **Analisis DAFO**
6. **Requisitos Funcionales**
7. **Requisitos No Funcionales**
8. **Casos de uso**
9. **Diagrama de Gantt**
10. **Diseño de la base de datosç**
11. **Diseño de interfaces**
12. **Desarrollo en entorno servidor**
13. **Desarrollo web en entorno cliente**
14. **Conclusión**

**Introducción**

La idea de este proyecto proviene de la falta de optimización de recursos y tratamiento de incidencias y otros problemas en las instituciones publicas que hemos notado a lo largo del tiempo.

Gracias a este proyecto tenemos la idea de poder crear un programa con el que agilizar un poco este proceso, así ayudando a que las incidencias se resuelvan mas fácilmente y tener un inventariado limpio y organizado.

**Objeto del proyecto**

El “Gestor REI” es un sistema de gestión utilizado principalmente para levantar incidencias, gestionar inventario y poder recoger datos sobre las tareas técnicas del trabajo (como el rendimiento de cada técnico en el trabajo).

Gracias a este sistema cualquier usuario puede levantar una incidencia y asignarla a sus empleados/subordinados o cualquier técnico de la institución.

**Objetivos**

* Agilizar proceso de creación de incidencias.
* Crear objetos en inventario.
* Manejar inventariado general.
* Crear un sistema de jerarquías entre usuarios para distinguir jefes y empleados.
* Crear un sistema de asignación para asignar incidencias a los técnicos.

# **ALTERNATIVAS**

La alternativa principal es GLPI es un sistema bastante completo.

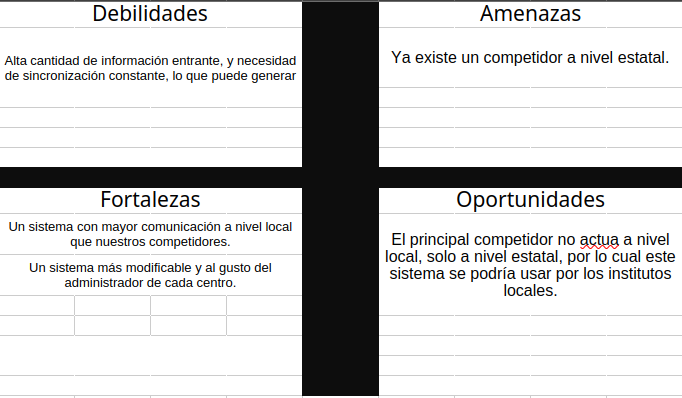
**Ventajas:**

* Nuestro sistema es más modificable en su enteridad.
* Nuestro sistema incluye un sistema de jerarquías para declarar que usuario es jefe de que técnico.
* Nuestro sistema trabaja a nivel local, por lo cual los administradores del sistema a nivel local pueden tener libre albedrío para modificar lo que quieran cuando quieran
* Recogida de datos de técnicos. Gracias a estos datos podemos ver el rendimiento de los técnicos.

**Desventajas:**

* Nuestro un sistema basado en permisos, por lo cual tiene más restricciones a la hora de asignar incidencias.
* Nuestro sistema no tiene distintos tipos de objetos, sino que usa una plantilla a la hora de crearlos.

# **Análisis DAFO**



# **REQUISITOS FUNCIONALES**

**RF-1 Gestión de usuarios**

* **RF1.1** Registro Usuarios
* **RF1.2** Dar baja usuarios Usuarios
* **RF1.3** Modificar datos del usuario
* **RF1.4** Autenticar la sesión del usuario
* **RF1.5** Cierre de sesión
* **RF1.6** Sistema de privilegios
* **RF1.7** Panel de administrador
* **RF1.8** Gestión de los datos del los usuarios en BD
* **RF1.9** Crear jerarquía varios usuarios

**RF-2 Tareas**

* **RF2.1** Registrar tareas
* **RF2.2** Eliminar tareas
* **RF2.3** Cambiar estados tarea
* **RF2.4** Modificar y responder a la tarea
* **RF2.5** Asignar tarea a usuario
* **RF2.6** Dar nombre a la tarea
* **RF2.7** Dar una descripción a la tarea
* **RF2.8** Dar un tiempo estimado de la tarea

**RF-3 Institución**

* **RF3.1** Crear instituciones (o centros)
* **RF3.2** Dar de baja instituciones

**RF-4 Inventario**

* **RF4.1** Asignar Inventario a Institución

**RF-5 Gestión Objetos**

* **RF5.1** Crear Objetos
* **RF5.2** Dar de baja Objetos
* **RF5.3** Dar estado a objeto (En inventario, de baja, desactivado, etc..)
* **RF5.4** Dar nombre a Objetos
* **RF5.5** Dar descripción a objeto
* **RF5.6** En caso de avería dar descripción de la propia

# **REQUISITOS NO FUNCIONALES**

* **RNF1.1** Crear Promedios de los datos del usuario (tareas completadas, etc)
* **RNF1.2** Gestión de expresiones regulares.
* **RNF1.2** Gestión seguridad del usuario.
* **RFN2.1** Crear sistema de respuesta de asignado a creador

# **CASOS DE USO**

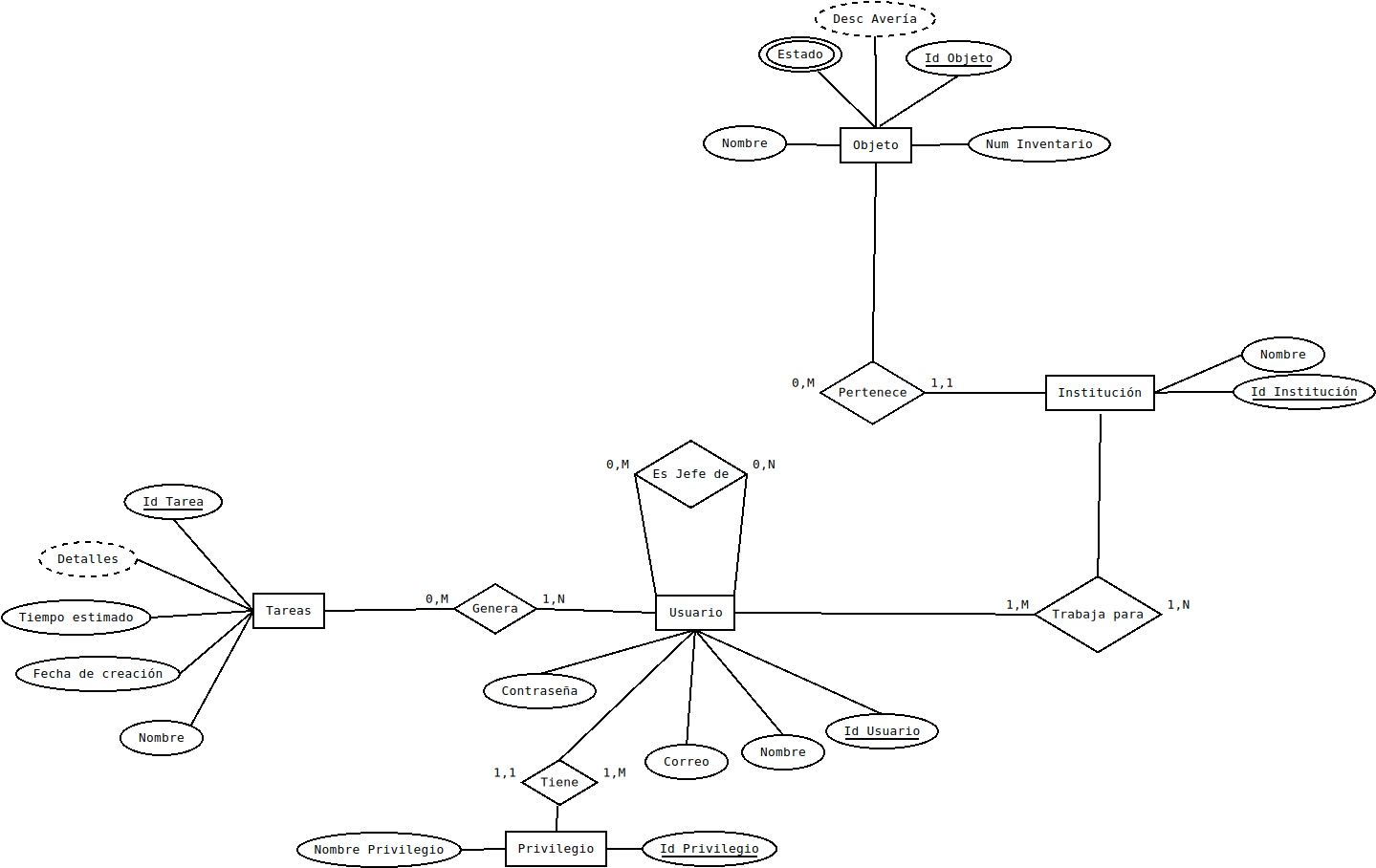
(Diagrama UML)

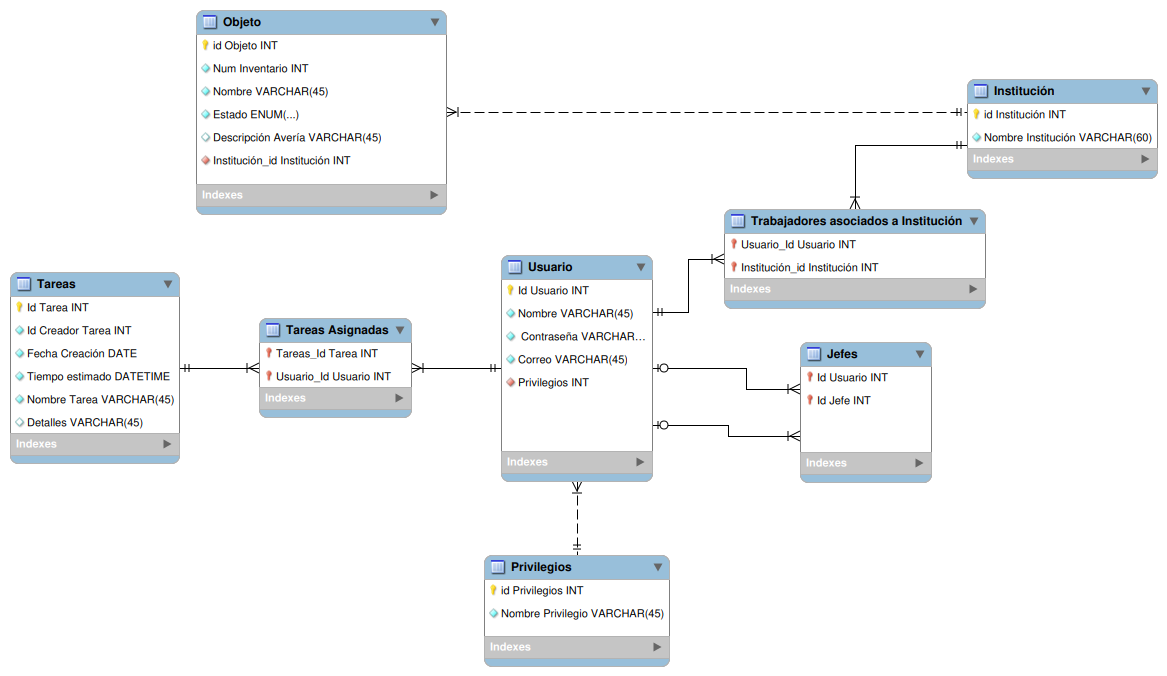
(Diagrama de Clases)

(Diagrama de Flujo de datos)

# **DIAGRAMA DE GANTT**

# **DISEÑO DE LA BASE DE DATOS**





# **DISEÑO DE INTERFACES**

(Paleta de colores)

(Tipografía)

(Iconos y otros elementos)

(Wireframes)

(Mockups)

# **DESARROLLO EN ENTORNO SERVIDOR**

(USO POO)

(USO PDO)

(USO MVC)

(Modelos Planteados)

(Controlador frontal y controladores)

(USO DAO)

(Uso de clases Helpers: Seguridad y validaciones)

(VISTAS)

(Partials y/o plantillas)

(Estructura del proyecto)

(USO API)

(USO COOKIES)

(CONTROL DE PERFILES DE USUARIO Y SESIONES)

(CONTROL DE EXCEPCIONES)

(PHPUNIT)

(EXPLICACIÓN DE FLUJO EN CONTROLADORES)

(¿Uso de composer?)

(¿Librerias?)

(¿Uso de autoload?)

(¿Uso de Ajax?)

# **DESARROLLO WEB EN ENTORNO CLIENTE**

(COOKIES)

(VALIDACIÓN DE DATOS)

(DOM)

(TRY CATCH)

(EVENTOS)

(PÁGINA MAIN)

(AJAX)

(JQUERY??)

# **CONCLUSIÓN**