Desarrollo de Sistemas de información para la empresa Lavamatic la italiana

MANUAL TÉCNICO

DE IMPLANTACIÓN

Versión: 0001

Norma IEEE

Fecha: 05/03/2024

VERSION 01.

| *Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso, protegidas es la Ley 23 de 1982, conocida como la "Ley de Derechos de Autor”. Colombiana.* |
| --- |

| **SENA** | **CENTRO DE SERVICIOS FINANCIEROS**  COORDINACION DE TECNOLOGÍAS DE LA INFORMACIÓN. | | |
| --- | --- | --- | --- |
| **Proyecto** | **JSJV** | | |
| **Entregable** | Manual de Usuario | | |
| **Autor** | TEAM SCRUM (Nº1 ). Tecnólogo en Análisis y Desarrollo de Software | | |
| **Versión/Edición** | 0001 | **Fecha Versión** | 05/03/2024 |
| **Aprobado por** | NOMBRE INSTRUCTOR EVALUADOR | **Fecha Aprobación** | DD/MM/AAAA |
| **ADSO FICHA Nº** | **2558723** | **Nº Total de Páginas** | 5 |

REGISTRO DE CAMBIOS

| **Versión** | **Causa del Cambio** | **Responsable del Cambio** | **Fecha del Cambio** |
| --- | --- | --- | --- |
| 0001 | Versión inicial | Valentina Sanchez | 05/03/2024 |
| 0002 | N/A | N/A | N/A |
| 0003 | N/A | N/A | N/A |

CONTROL DE DISTRIBUCIÓN

| **Nombre y Apellidos (TEAM SCRUM)** |
| --- |
| Jaider Steven Aponte |
| Julian Santiago Millan |
| Valentina Sanchez Caicedo |
|  |
|  |

**CONTENIDO**

INTRODUCCIÓN

OBJETIVO

Desarrollar un software (***JSJV)***la facilidad para el manejo de los procesos como de gestión(cambiar el gestión) de empleados, gestión de inventarios, gestión de actividades y gestión de servicios, reemplazando el sistema de información actual para la optimización de los procesos productivos de la empresa.

**1.REQUERIMIENTOS TÉCNICOS**

1.1 REQUERIMIENTOS MÍNIMOS DE HARDWARE

* Procesador: i7-(9)
* RAM: 16
* Almacenamiento: 2 TB
* Red: Estrella o Malla

1.2 REQUERIMIENTOS MÍNIMOS DE SOFTWARE

* **Sistema Operativo:** Windows 10.
* **Navegador Web**: Google Chrome.
* **Software de Desarrollo:** Visual Studio Code.
* **Base de Datos:** MySQL.
* **Control de versiones**: Git.

**2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

2.1 PYTHON:

Para la implementación del sistema virtual JSJV, se utilizará el lenguaje de programación python en la versión 3.11.3 y el framework de Django.

2.2 MYSQL:

En la instalación de MYSQL utilizaremos la aplicación XAMPP en su versión 3.3.0.

2.3 APACHE:

De igual manera usaremos XAMPP para la instalación de APACHE.

3. INSTALACIÓN DE WEB SERVER (o servidor) (PANTALLAZOS)

4. CONFIGURACIÓN DEL APLICATIVO

4.1.1 Reemplazar archivo controller (Como se usó el modelo vista controlador)

1. Identificar la vista en views.py en este se alojan los “controller” del sistema.
2. Crea una nueva función de vista con la lógica actualizada.
3. Actualiza las URLs en urls.py para enlazar la nueva vista..
4. Realizar pruebas y depuración para asegurar su funcionamiento.

4.2.1 Configurar la base de datos

1. Configuración: En el archivo settings.py del proyecto se configura la conexión a la base de datos.
2. Implementar base de datos: Aparte en la aplicación principal se va a crear un archivo llamado db.py el cual se va a configurar las rutas de conexión a la base de datos:

MYSQL = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'jsjv2',

'USER': 'root',

'PASSWORD': '',

'HOST': 'localhost',

'PORT': '3306'

}

}

1. Definir Modelos: En las aplicaciones que creamos, se definen los modelos utilizando la clase models.Model. Cada modelo representa una tabla en la base de datos y define sus campos y relaciones.
2. Migraciones: Generar migraciones para crear las tablas correspondientes en la base de datos.Ejecutando python manage.py makemigrations para crear las migraciones y python manage.py migrate para aplicarlas y crear las tablas en la base de datos.
3. Administrar Datos: Utilizamos el admin de django (admin.py) para gestionar los datos de los modelos. Registrando los modelos podemos hacer método MVC desde el panel de administración.

4.3.1 Configurar El Módulo Administración

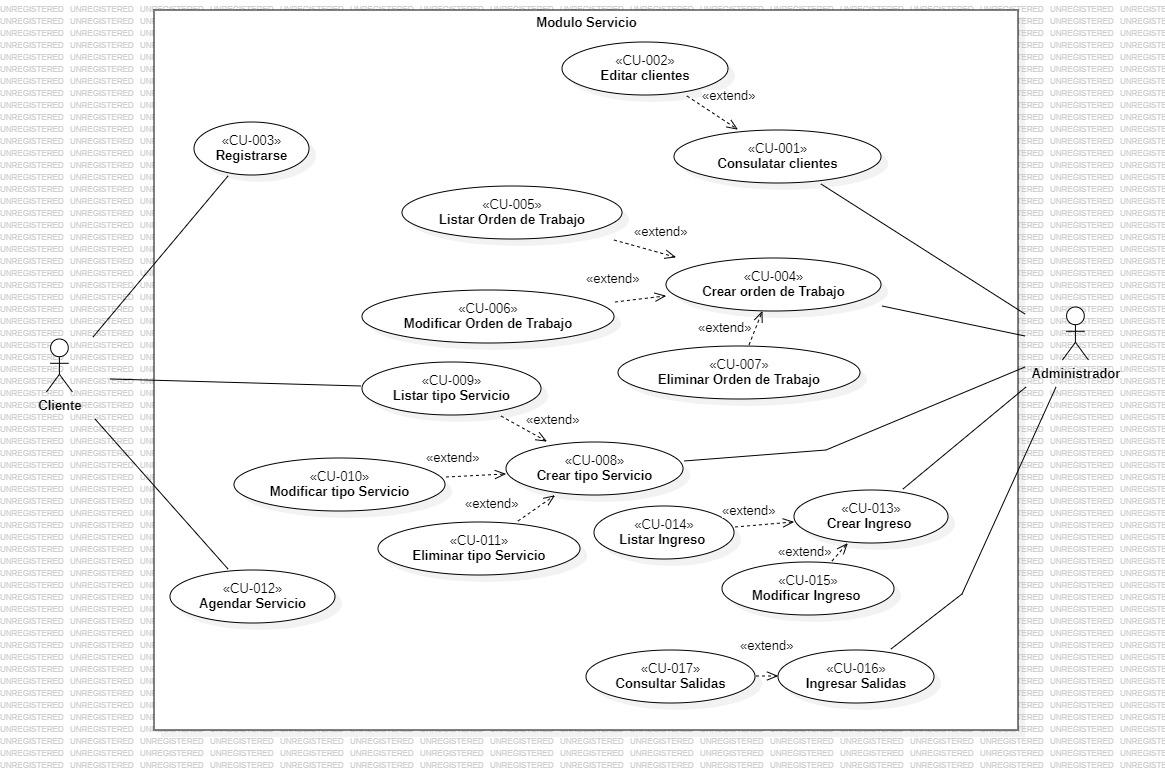
1. Crea un superusuario con python manage.py createsuperuser.
2. En el archivo admin.py de la aplicación django, registrar los modelos que se deben administrar en el panel de administrador.

4.4.1 Servidor activo

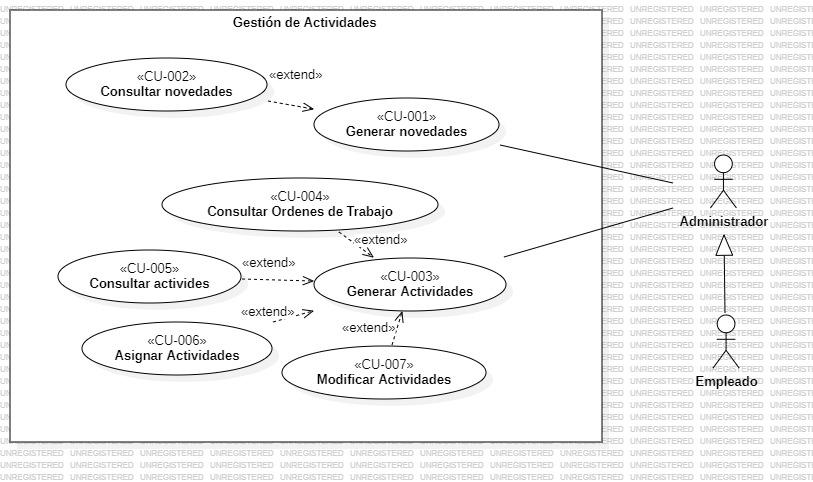
1. Ejecutar el comando python manage.py runserver en la terminal para iniciar el servidor de desarrollo de Django. Por defecto, el servidor se ejecutará en <http://127.0.0.1:8000/>.
2. Para acceder al panel de administrador copia el link y pegalo en un navegador web al final de este pones /admin para que te lleva al panel del administrador([http://127.0.0.1:8000/admin/.)](http://127.0.0.1:8000/admin/.Se) Se inicia sesión como superAdmin para acceder al panel de administración y gestionar los modelos registrados.

5. CASOS DE USO

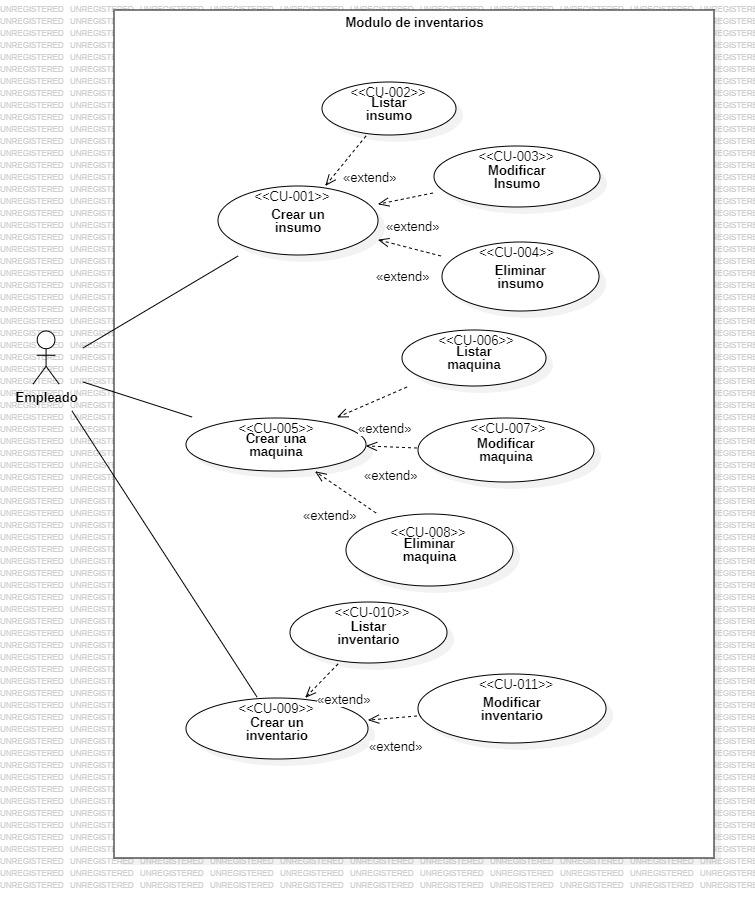
* Diagrama de casos de uso **Servicios**



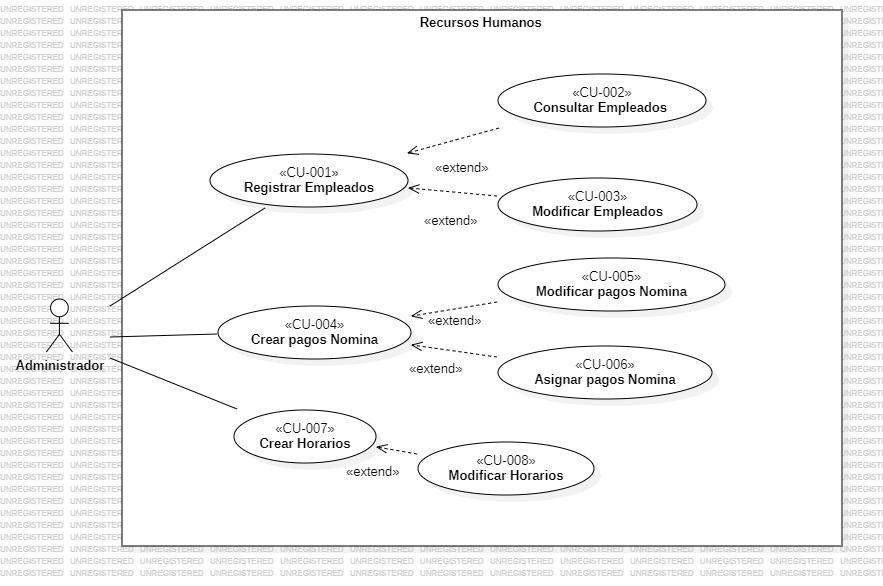
* Diagrama de casos de uso **Actividades**

****

* Diagrama de casos de uso **Inventarios**

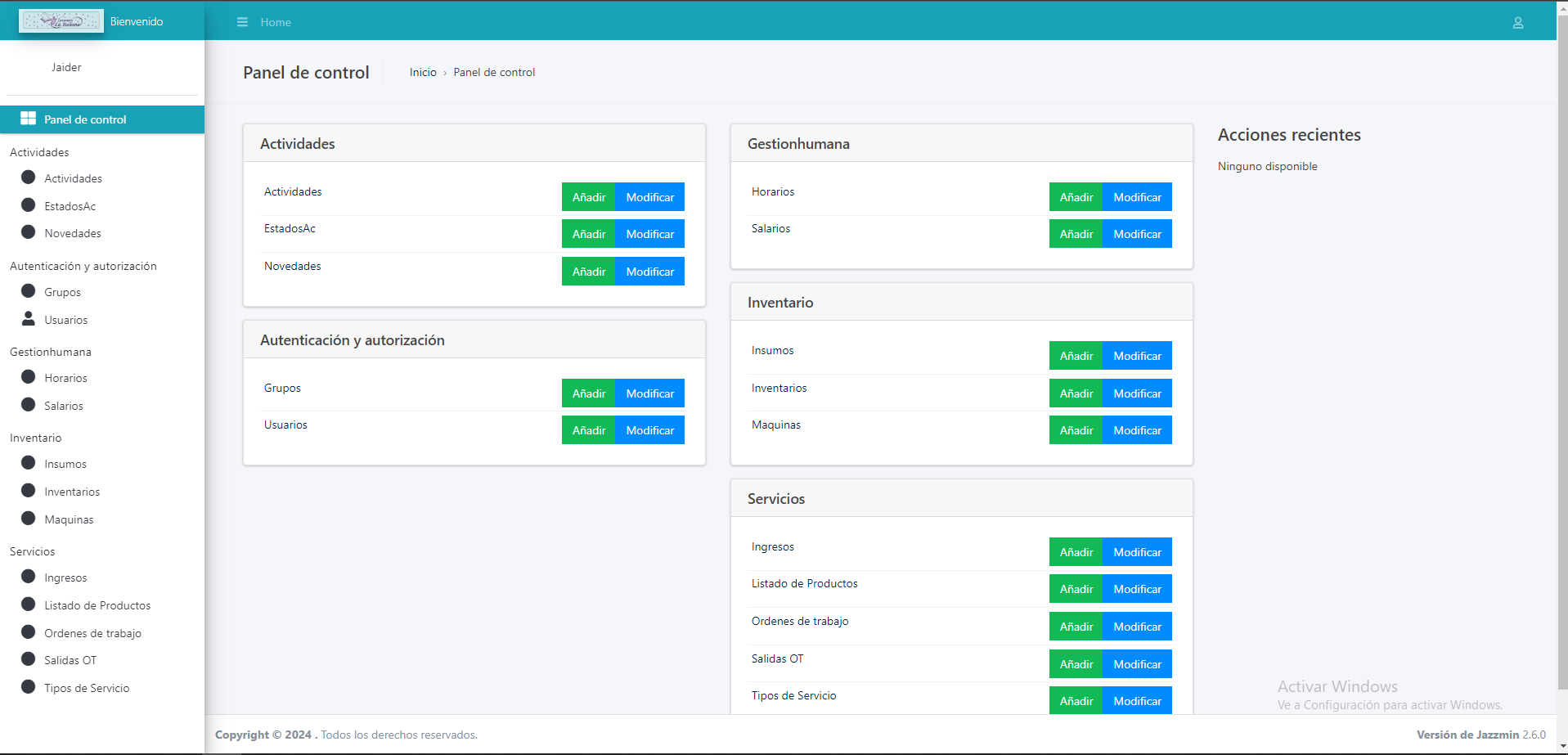
****

* Diagrama de casos de uso **Gestión Humana**

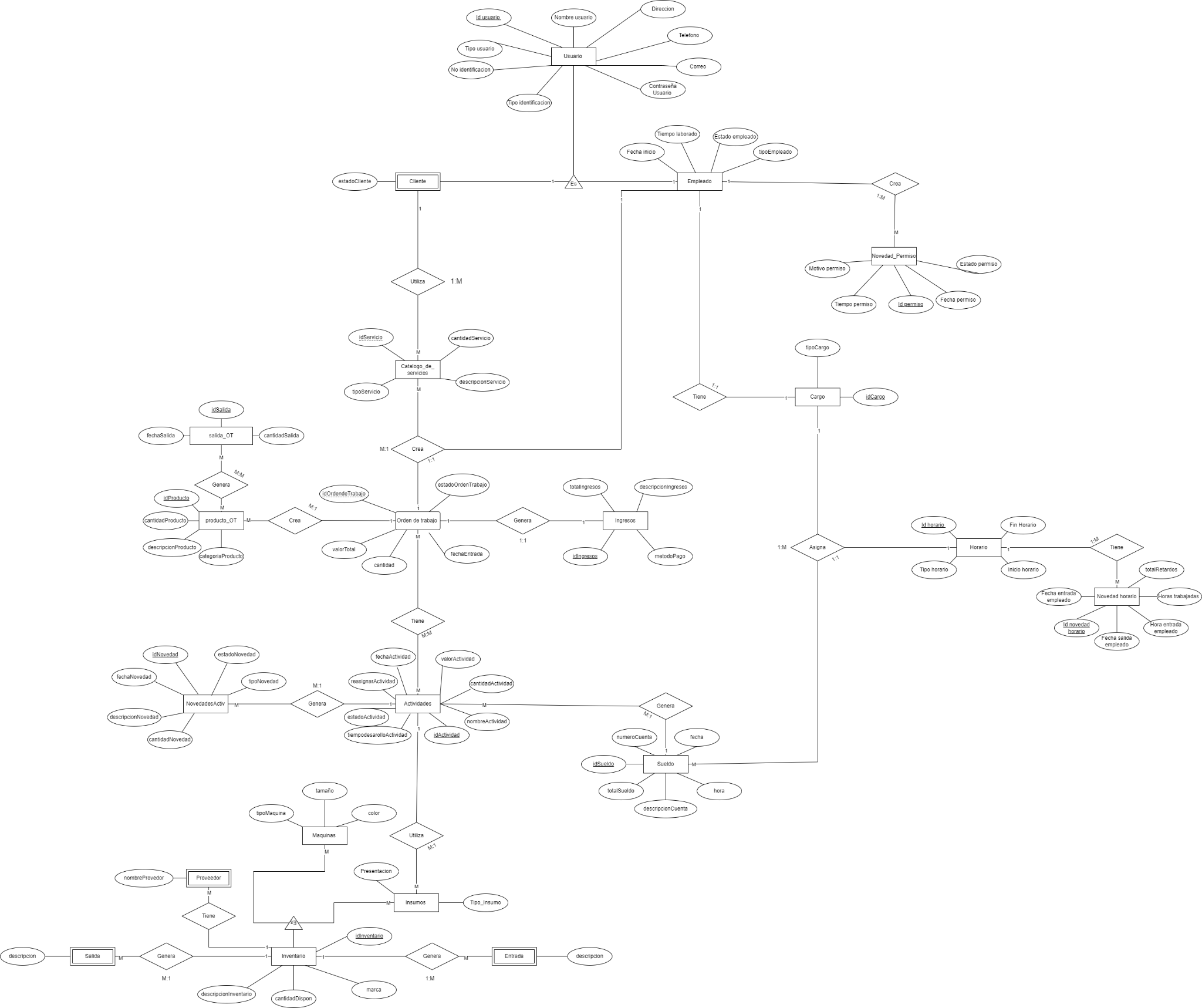


6. MODULO DE ADMINISTRACION

Imagen vista Administrador



7. MODELO ENTIDAD RELACIÓN



8. DICCIONARIO DE DATOS DEL MODELO ENTIDAD RELACIÓN

Imagen Diccionario de Datos: Usuarios



Imagen Diccionario de datos: Servicios

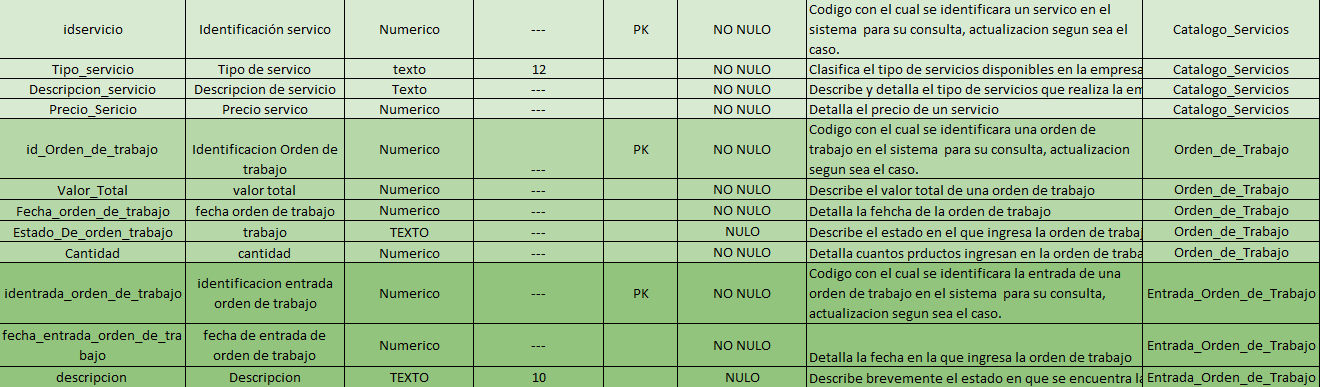


Imagen Diccionario de datos: Servicios

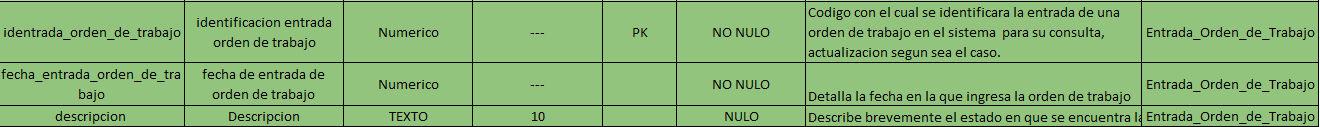


Imagen Diccionario de datos: Salida Y productos

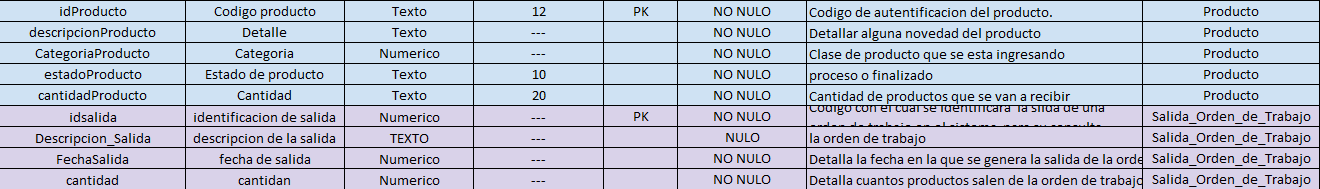


Imagen Diccionario de Datos: Inventario

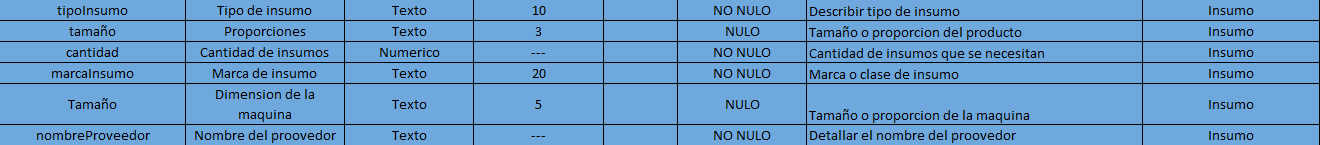


Imagen Diccionario de datos : Inventario

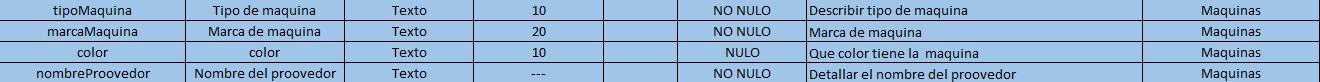


Imagen Diccionario de datos : Inventario



Imagen Diccionario de datos: Actividades



Imagen Diccionario de datos : Facturacion

