**JSJV**

**Documento de Especificación de Arquitectura**

Realizado por:

-Kevin Santiago Godoy

-Julian Santiago Millan

-Valentina Sanchez Caicedo

-Jaider Steven Aponte Parra

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
| <Escriba aquí> | 03 | Kevin Santiago Godoy  Julian Santiago Millan  Valentina Sanchez Caicedo  Jaider Steven Aponte Parra | <Escriba aquí> | <Escriba aquí> |

**Contenido**

[1.](#_heading=h.3znysh7) Documento de Arquitectura de Software 4

[1.1.](#_heading=h.2et92p0) Introducción 4

[1.2.](#_heading=h.tyjcwt) Propósito 4

[1.3.](#_heading=h.3dy6vkm) Alcance 4

[1.4.](#_heading=h.1t3h5sf) Referencias 5

[1.5.](#_heading=h.4d34og8) Definiciones acrónimos y abreviaciones 5

[2.](#_heading=h.2s8eyo1) Generalidades del Proyecto 5

[2.1.](#_heading=h.17dp8vu) Problema a Resolver 5

[2.2.](#_heading=h.3rdcrjn) Descripción General del Sistema a Desarrollar 5

[2.3.](#_heading=h.26in1rg) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_heading=h.lnxbz9) Vistas de la arquitectura 6

[3.1.](#_heading=h.35nkun2) Vista de Casos de Uso 6

[3.2.](#_heading=h.1ksv4uv) Vista de Procesos 6

[3.3.](#_heading=h.44sinio) Vista Lógica 6

[3.4.](#_heading=h.z337ya) Vista de Implementación 6

[3.5.](#_heading=h.3j2qqm3) Vista de Despliegue 6

[4.](#_heading=h.1y810tw) Arquitectura en capas 6

[5.](#_heading=h.4i7ojhp) Vista de Datos 7

[5.1.](#_heading=h.2xcytpi) Modelo Relacional 7

[6.](#_heading=h.1ci93xb) Definición de Interfaces de Usuario 7

[7.](#_heading=h.3whwml4) Características Generales de Calidad 7

[7.1.](#_heading=h.2bn6wsx) Tamaño y performance 7

[7.2.](#_heading=h.1pxezwc) Calidad 7

[7.3.](#_heading=h.49x2ik5) Usabilidad 7

[7.4.](#_heading=h.2p2csry) Eficiencia 7

[7.5.](#_heading=h.147n2zr) Seguridad 7

[7.6.](#_heading=h.3o7alnk) Confiabilidad 7

[7.7.](#_heading=h.23ckvvd) Mantenimiento 7

[7.8.](#_heading=h.ihv636) Estándares 7

# Documento de Arquitectura de Software

## Introducción

El propósito de este documento es el de analizar y definir las necesidades de la empresa Lavamatic la Italiana, se centrará en cambiar el antiguo sistema Droop Soft y adicionar nuevas funcionalidades que facilitarán las actividades productivas y administrativas de la empresa.

## Propósito (Describir el propósito de este documento)

El documento presenta la descripción de la arquitectura para el Sistema JSJV, en base al Modelo de Vistas “4+1”, que es de utilidad para los stakeholders (Interesados) debido al rol que desempeña cada uno en el desarrollo del sistema.

La arquitectura presentada corresponde a la solución propuesta para el sistema, proporcionando distintas vistas de la arquitectura según los diferentes puntos de vista de los involucrados en el desarrollo del aplicativo. El documento se compone de diagramas y diseños detallados, brindando un contexto y una descripción de las interacciones entre los usuarios y el sistema.

El informe está estructurado de acuerdo con la plantilla para la documentación de arquitecturas de software.

## Alcance (Describir el alcance del documento)

El documento contiene la representación del proyecto JSJV, mediante el diseño de diagramas dado el análisis previo de los requisitos del sistema para darle solución a una problemática presentada por un cliente.

El documento está organizado mediante tres ideas principales:

* Características generales del diseño.
* Los requisitos funcionales y no funcionales para el diseño.
* Los modelos y vistas que lo representan.

Los modelos implementados para la representación del sistema, permiten analizar el funcionamiento del sistema desde distintos puntos de vista, de forma que podemos implementar el “Modelo de arquitectura 4+1”. Gracias a esto, se puede llegar a implementar *el análisis de los requisitos, la implementación del diseño de solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.*

Cada uno de los modelos fueron implementados en la herramienta de modelado UML (Lenguaje de modelado unificado) mediante la herramienta …………...

## Referencias

1. Documento de Especificación de Requerimientos no funcionales.

<https://docs.google.com/document/d/1XLbWB_23NMGc40RpYam5BBRShJbG2GIE/edit>

1. Documento de Visión del Proyecto.

<https://docs.google.com/document/d/1nNE5rb8b1aShOu6-fhubuHyklvjSNEWO9r3vzZQhqm8/edit>

1. Plan de Proyecto del Sistema

## Definiciones acrónimos y abreviaciones

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA**: colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

# Generalidades del Proyecto

## Problema a Resolver

## La empresa *lavamatić la italiana* ubicada en la localidad de Keneddy , en el barrio San Andres 2,cuenta actualmente con un sistema de información llamado *Droop Soft,* el cual está desactualizado, Esto conlleva a problemáticas tales como :

* Deficiencia en la recepción de pedidos y servicios.
* Demora en la asignación de actividades a los empleados.
* Ausencia de inventario en el sistema Droop Soft.
* Mala organización de los horarios laborales.

## Descripción General del Sistema a Desarrollar

El sistema de información ***JSJV*** facilitará los procesos para las áreas de gestión humana, gestión de inventarios, gestión de actividades y gestión de servicios, reemplazando el sistema de información actual para la optimización y automatización de las funciones manuales observadas en la compañía especializada en el área de lavado de prendas y muebles.

No solo actualizará las funciones de servicios, también implementará funciones en la gestión humana, actividades e inventario.

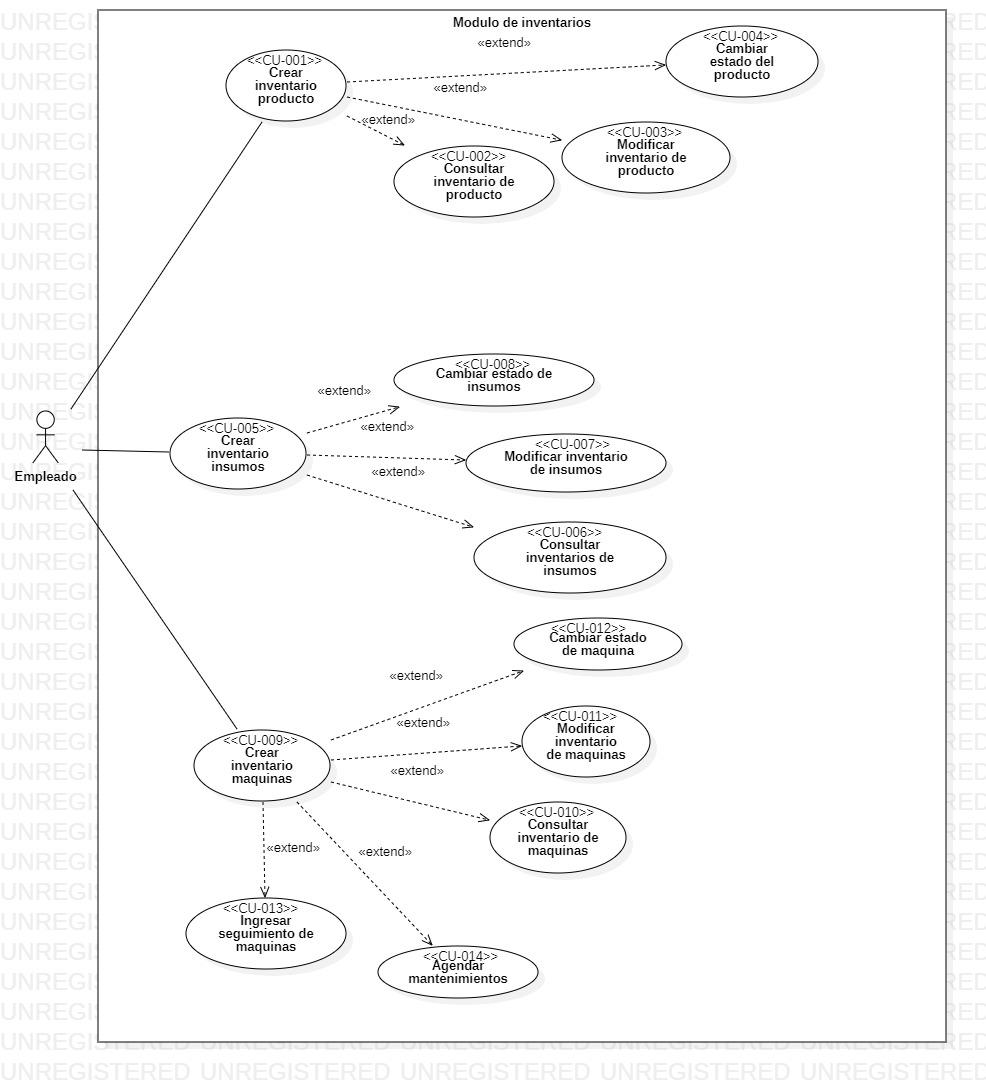
## Identificación de los Stakeholders y sus responsabilidades

| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **Caso de Uso** |
| --- | --- | --- | --- |
| Administrador | Es el usuario encargado de gestión dentro del sistema. | * Registra y almacena los datos del cliente en la solicitud de un servicio. * Registrar e implementar inventario. * Es el encargado de asignar, modificar y registrar los empleados y sus horarios. * Es el encargado de manipular el estado de los empleados dentro de la empresa. * Es el encargado de la asignación de actividades en la empresa. | * CU\_registrar clientes * CU crear órdenes de trabajo. * CU crear inventario de máquinas. * CU crear inventario de insumos. * CU crear inventario de productos. * CU modificar horarios de trabajo. * CU modificar empleados. * CU Asignar Actividades. |
| Empleado | Es el usuario que interactúa en el sistema para tener un control en sus horarios, actividades y en las novedades correspondientes. | * Es el encargado de asignar, modificar el estado de las máquinas en la empresa. * Registrar y modificar los inventarios de insumos. * Registrar y modificar los inventarios de productos. * Crea novedades ocurridas durante el proceso de un servicio. * Registrar su horario de entrada y salida. | * CU\_modificar inventario de productos. * CU\_modificar el inventario de insumos. * CU\_modificar el inventario de productos. * CU\_Generar Novedades. * CU\_Registar hora de ingreso y salida. |
| Cliente | Es el usuario que consulta en el sistema el catálogo, hace pedidos de productos según su preferencia. | * Solicita un servicio. * Agenda un servicio y lo puede cancelar de igual forma. | * CU\_consulta tipo de servicio. * CU\_Agendar un servicio. * CU\_Cancelar un servicio. |

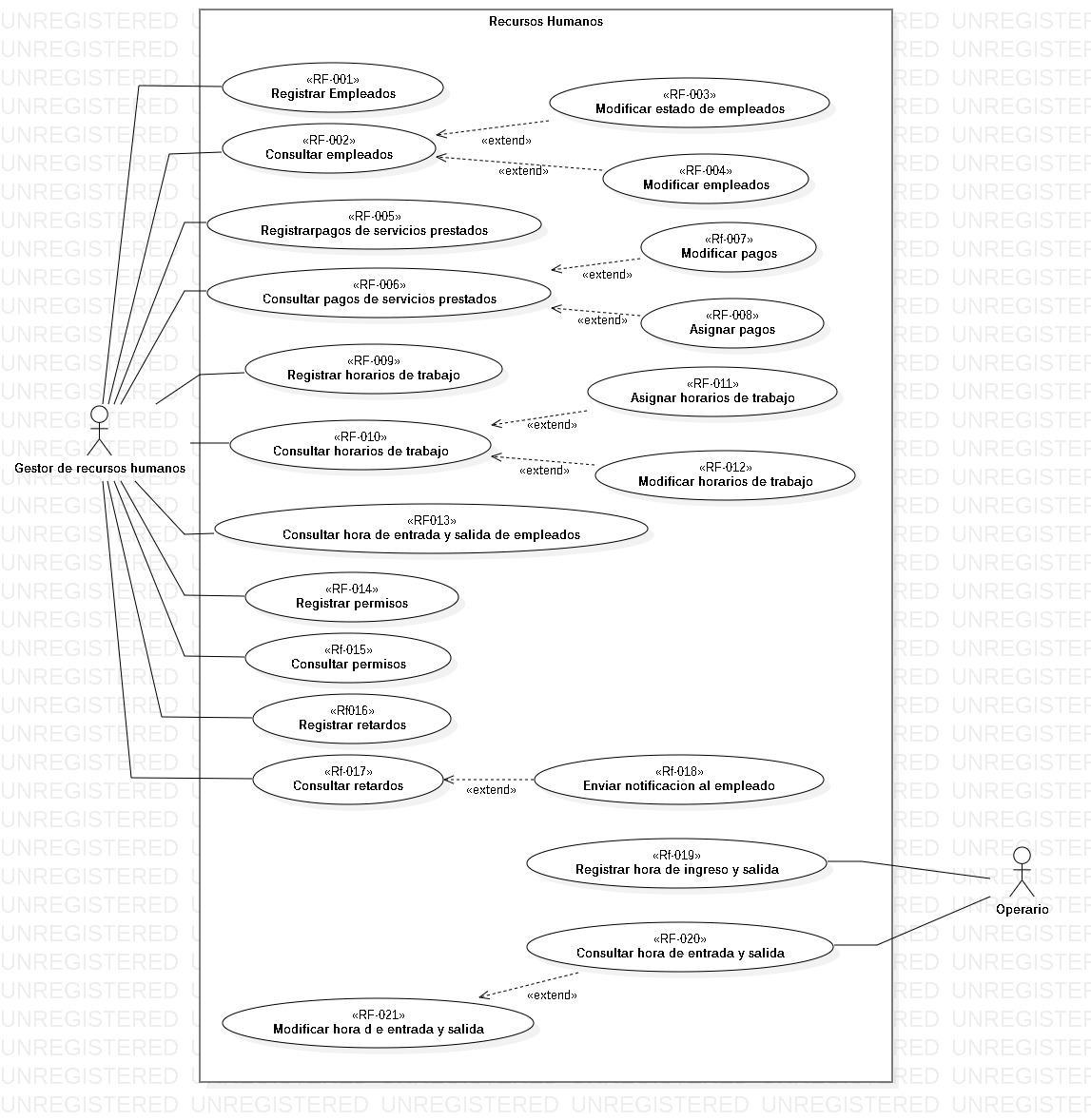
# Vistas de la arquitectura

## Vista de Casos de Uso

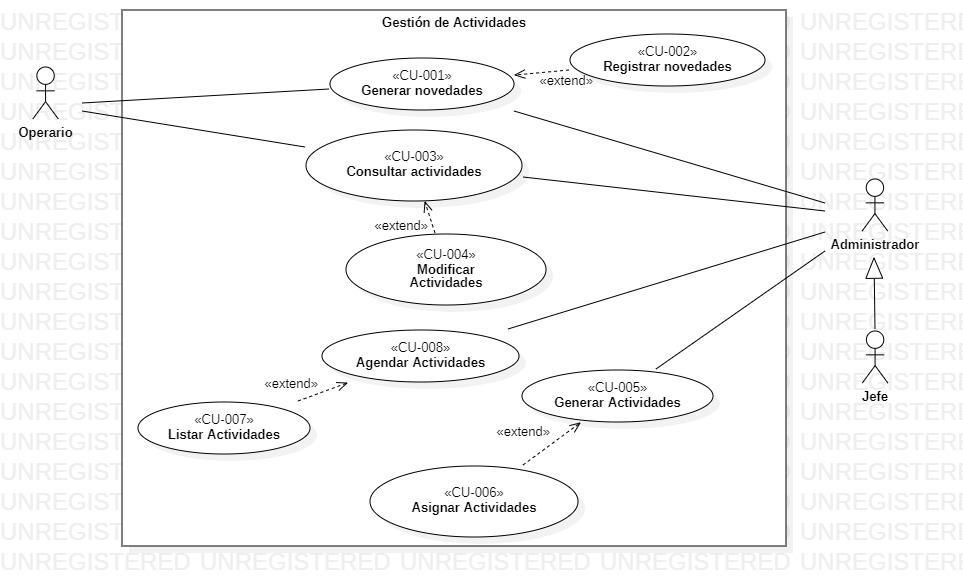
**Inventario**



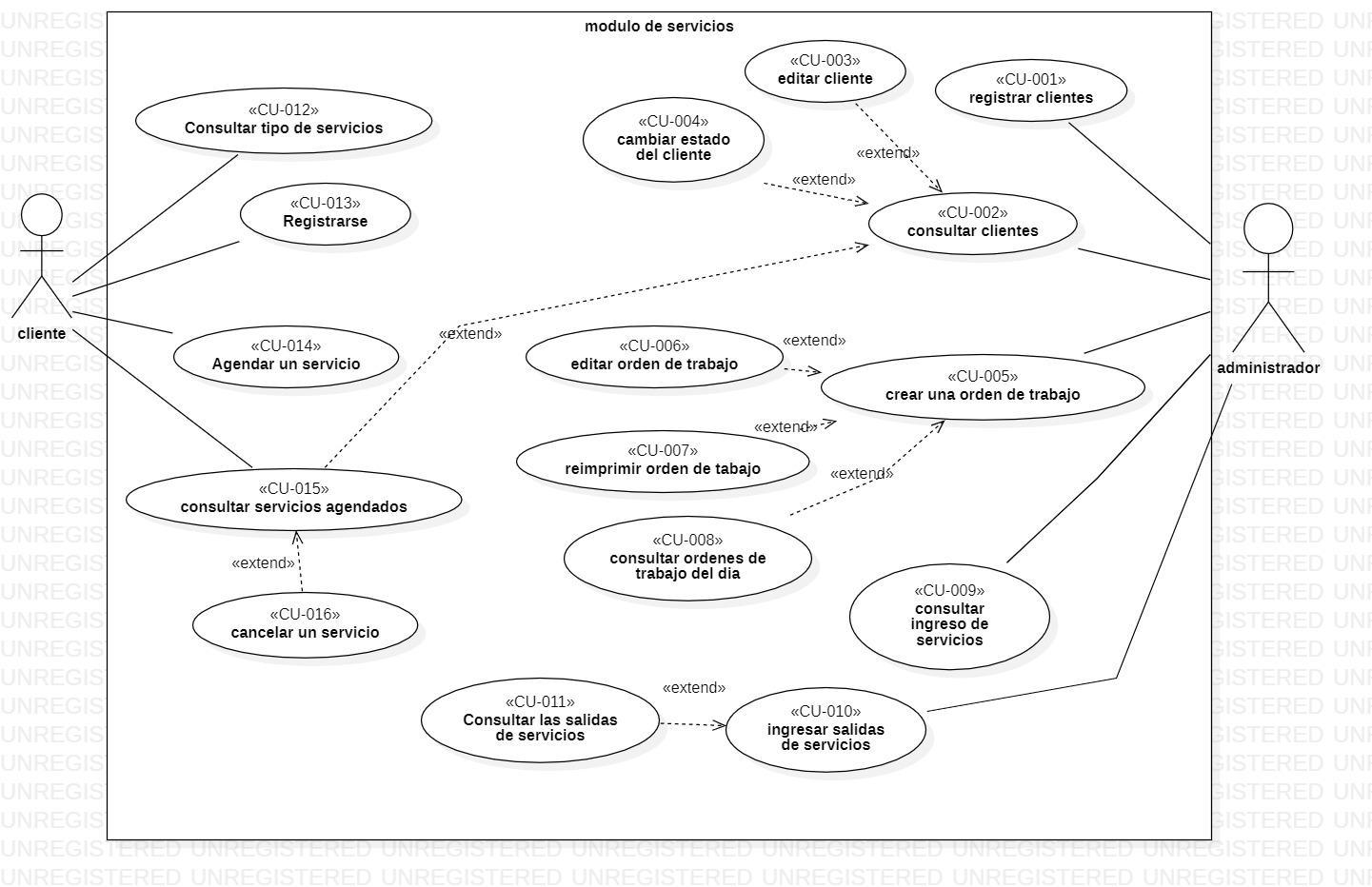
**Gestion Empleados**

****

**Actividades**

****

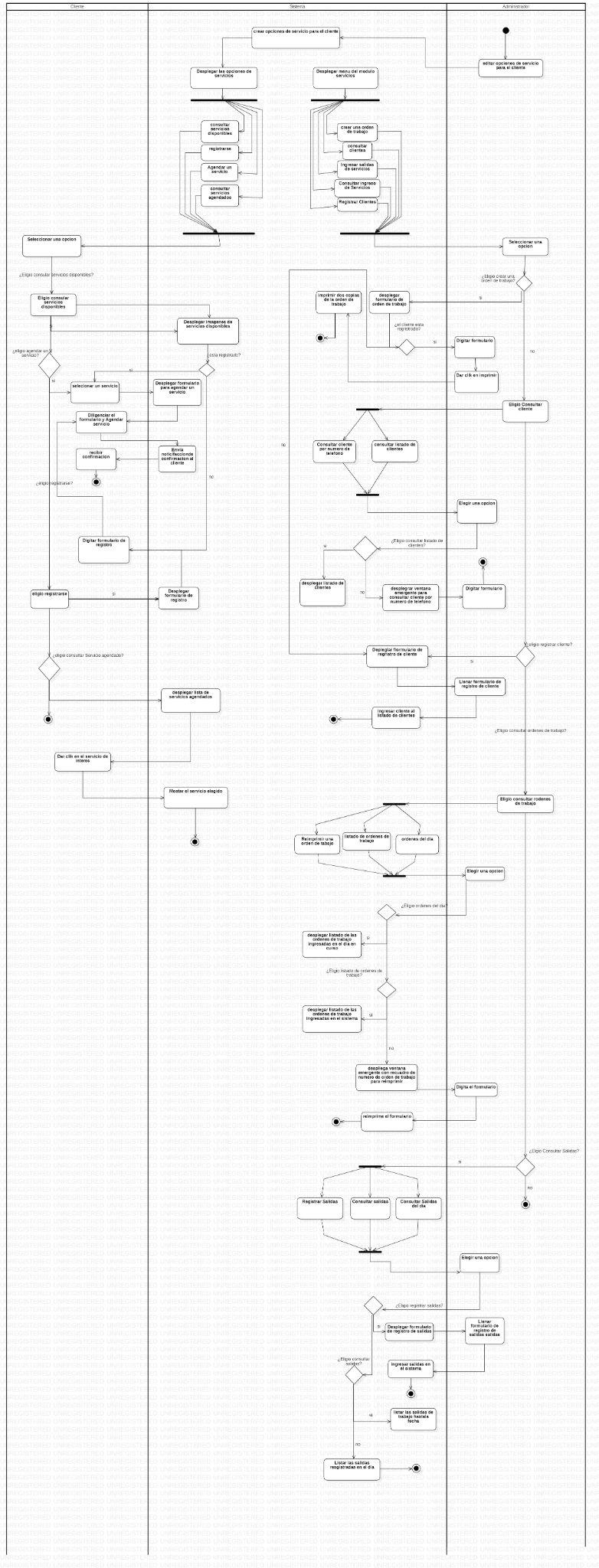
**Servicios**



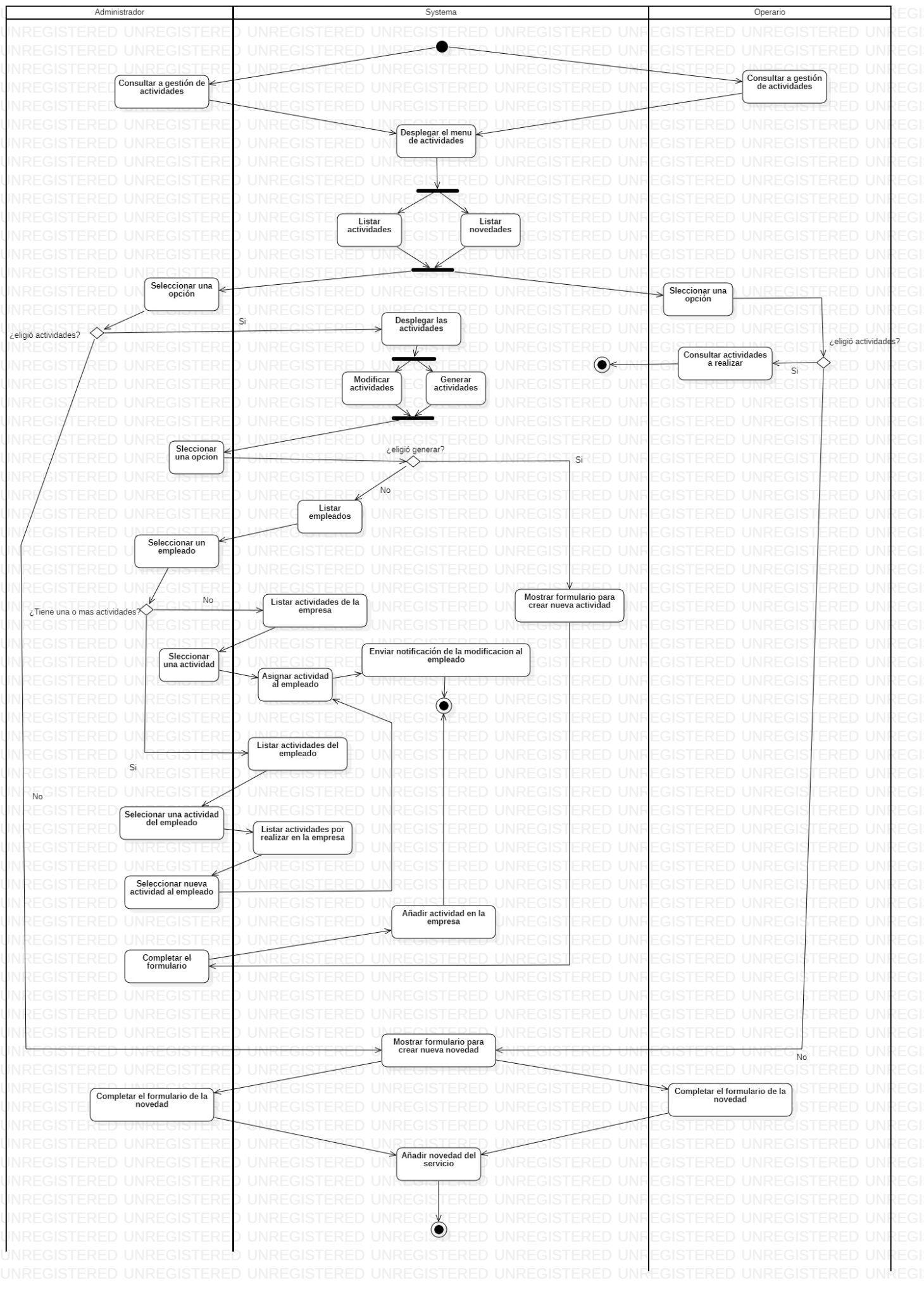
## Vista de Procesos

### Diagrama de Actividades

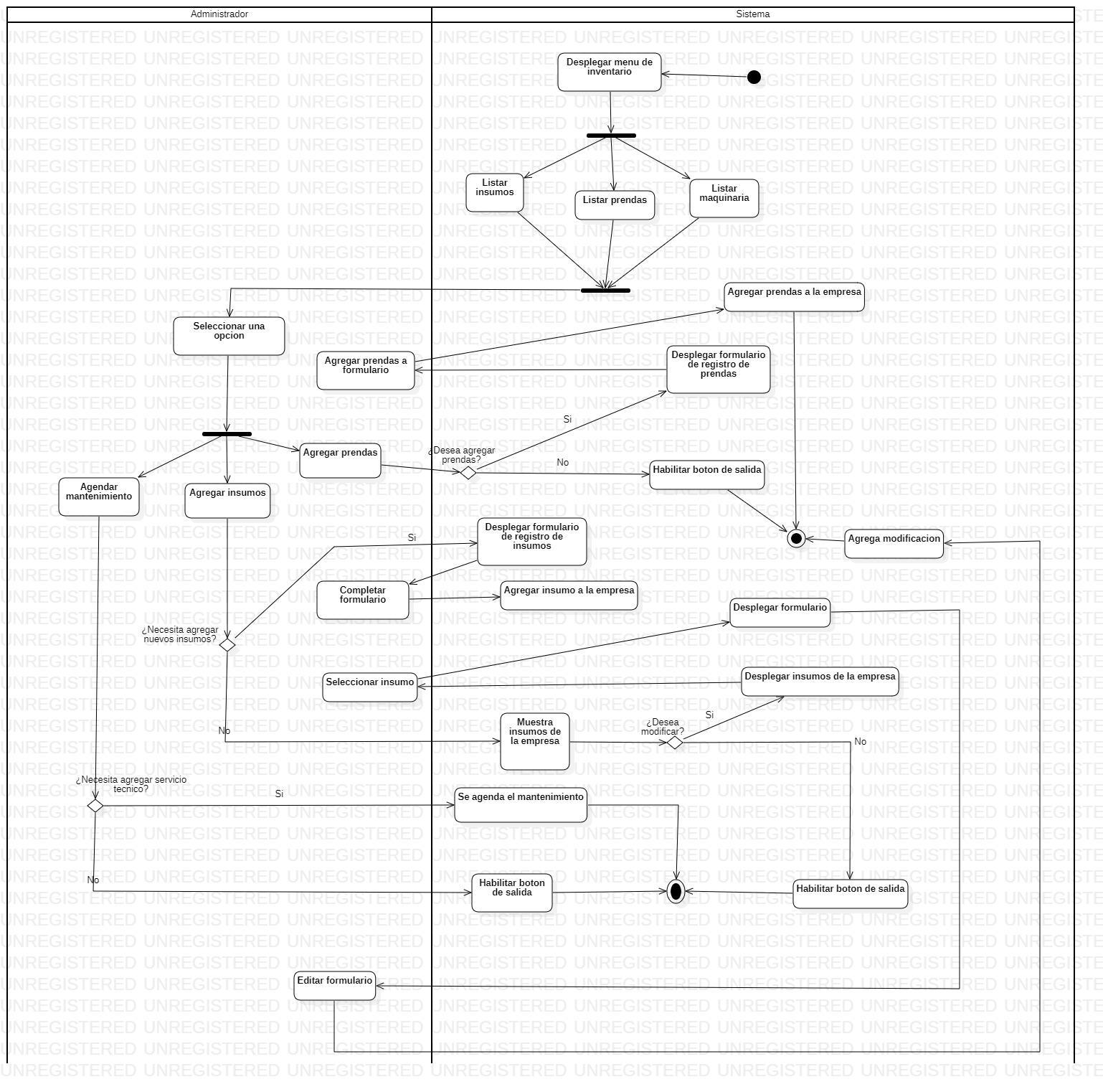
Servicios



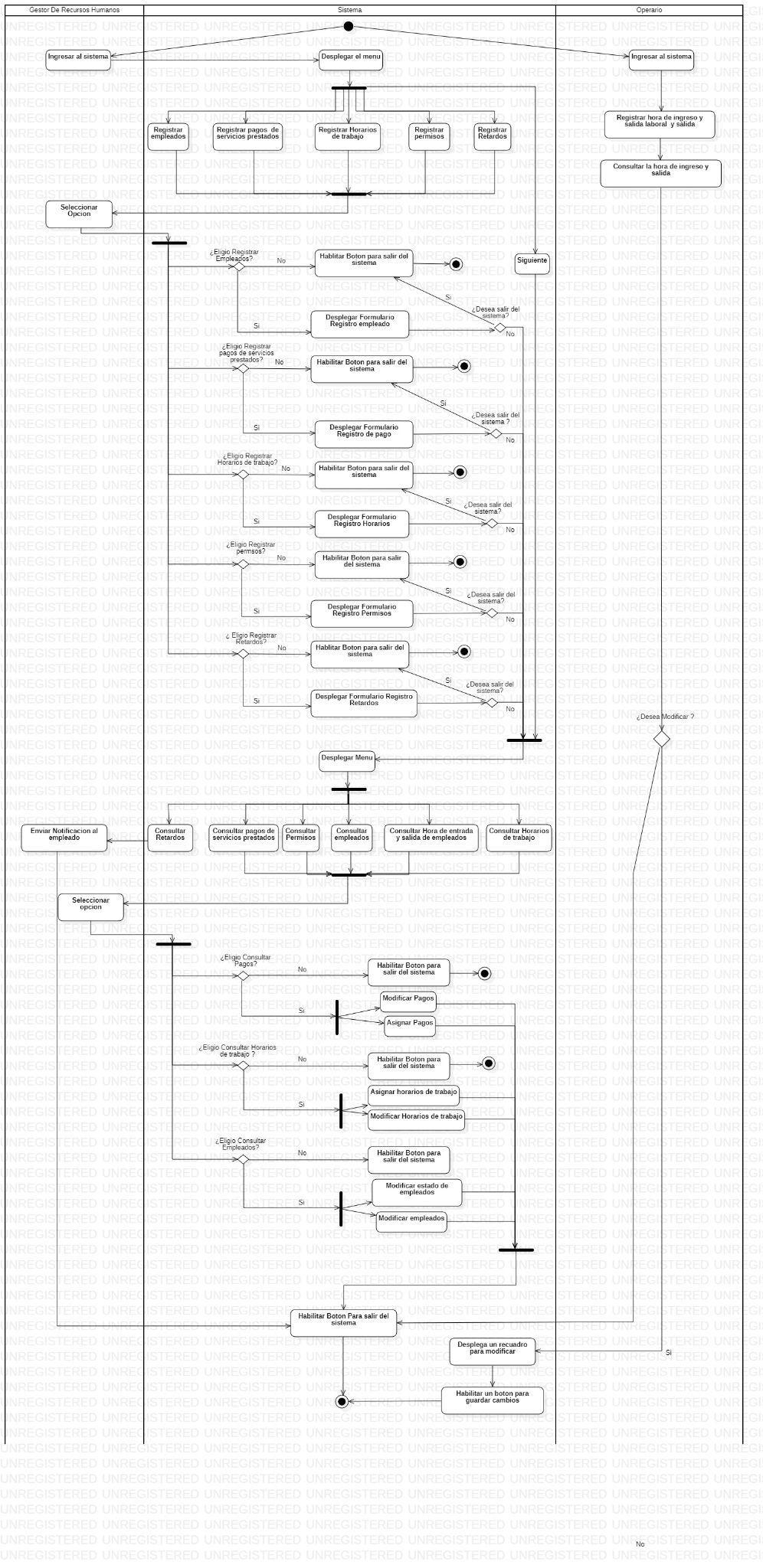
Actividades



Inventarios

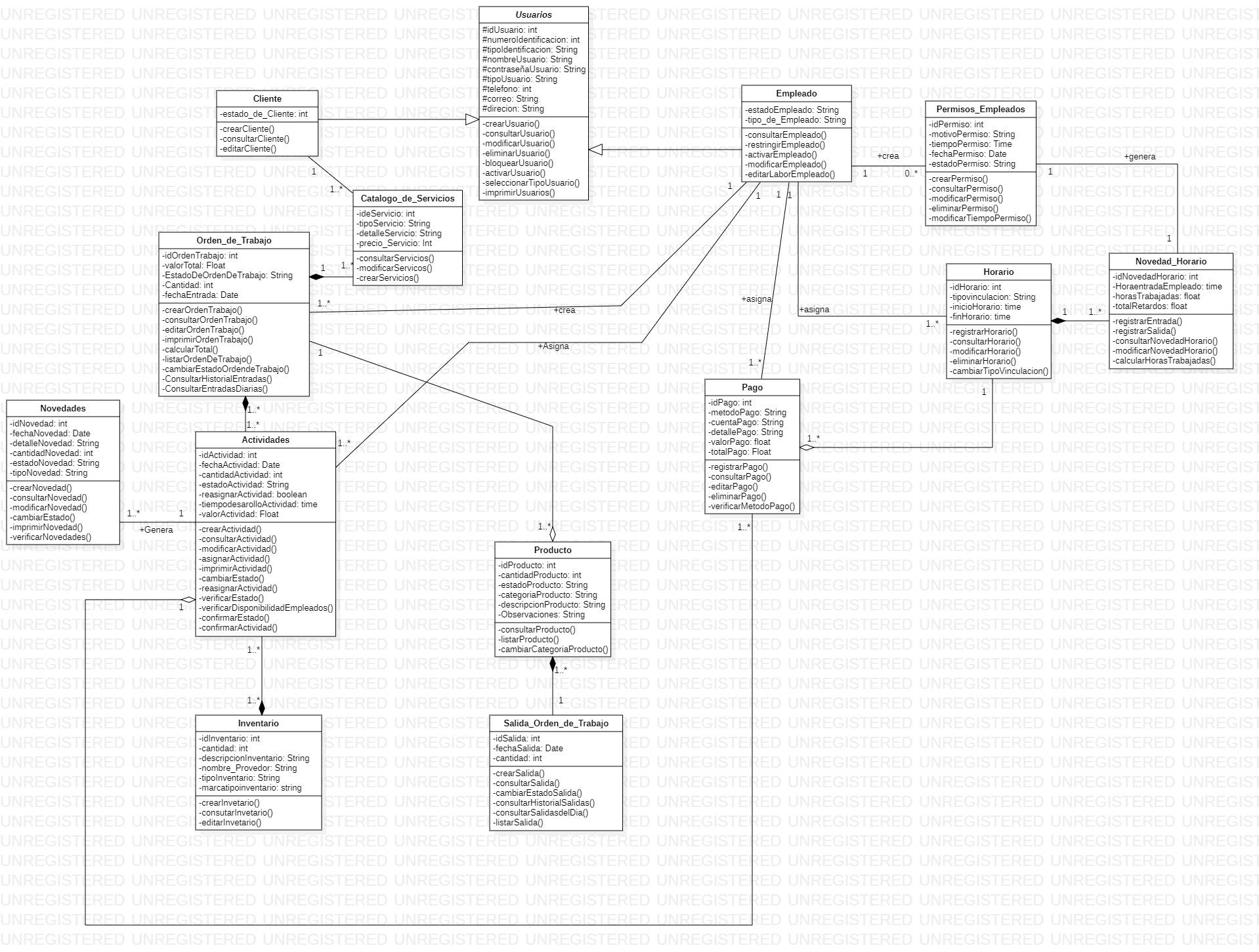
****

**Gestion de Empleados**

****

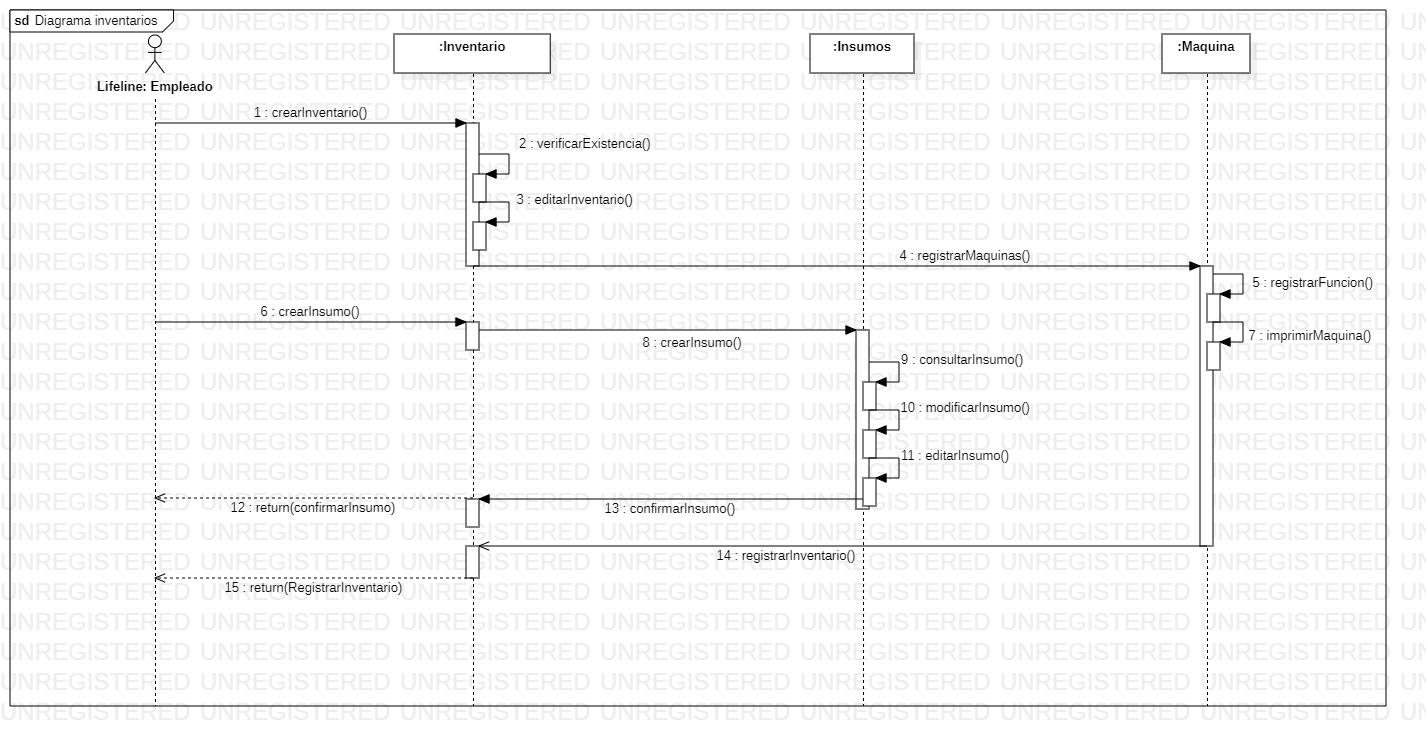
## Vista Lógica

### Diagramas – Clases

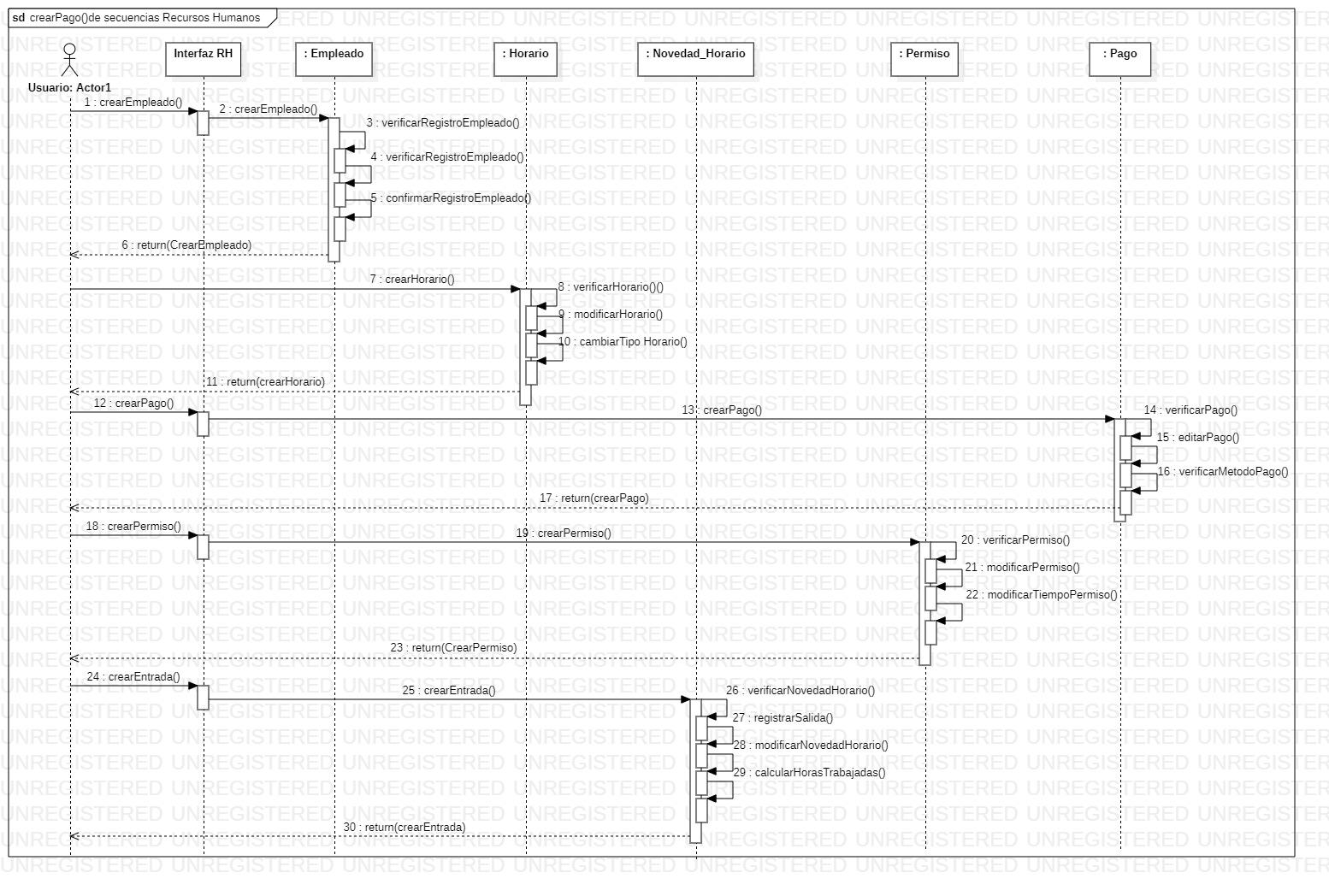


### Diagramas - Secuencia

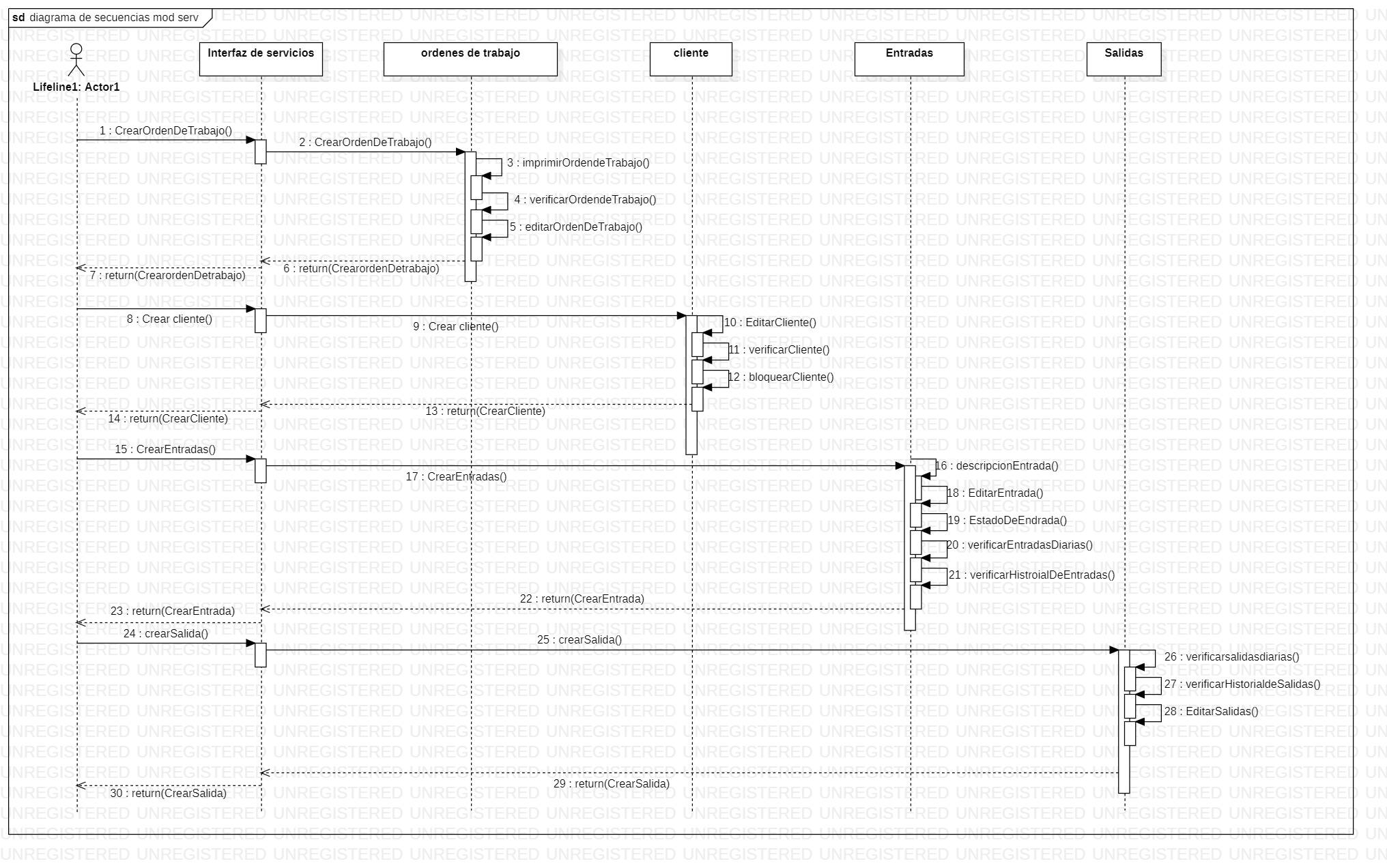
Inventarios



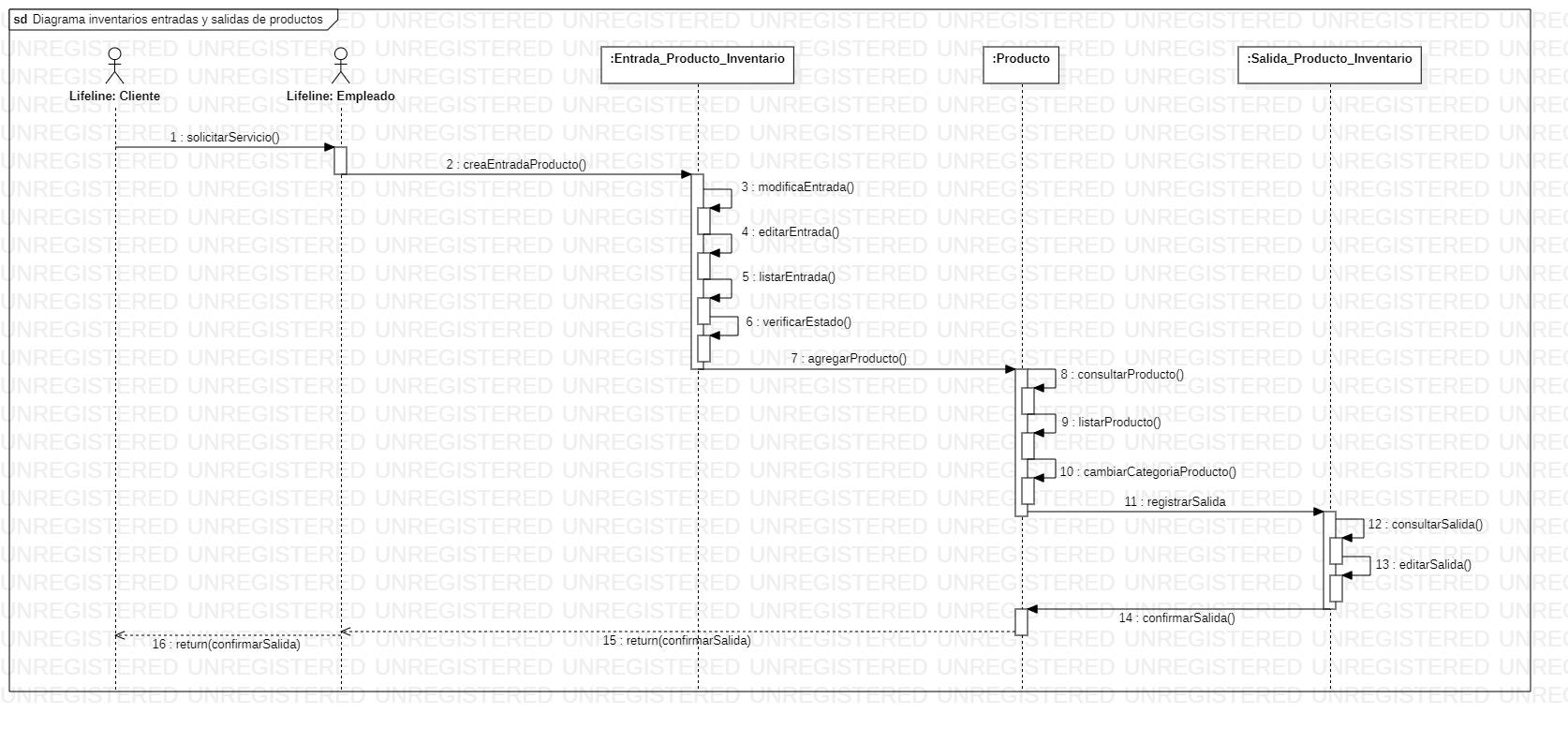
Gestion Empleados:



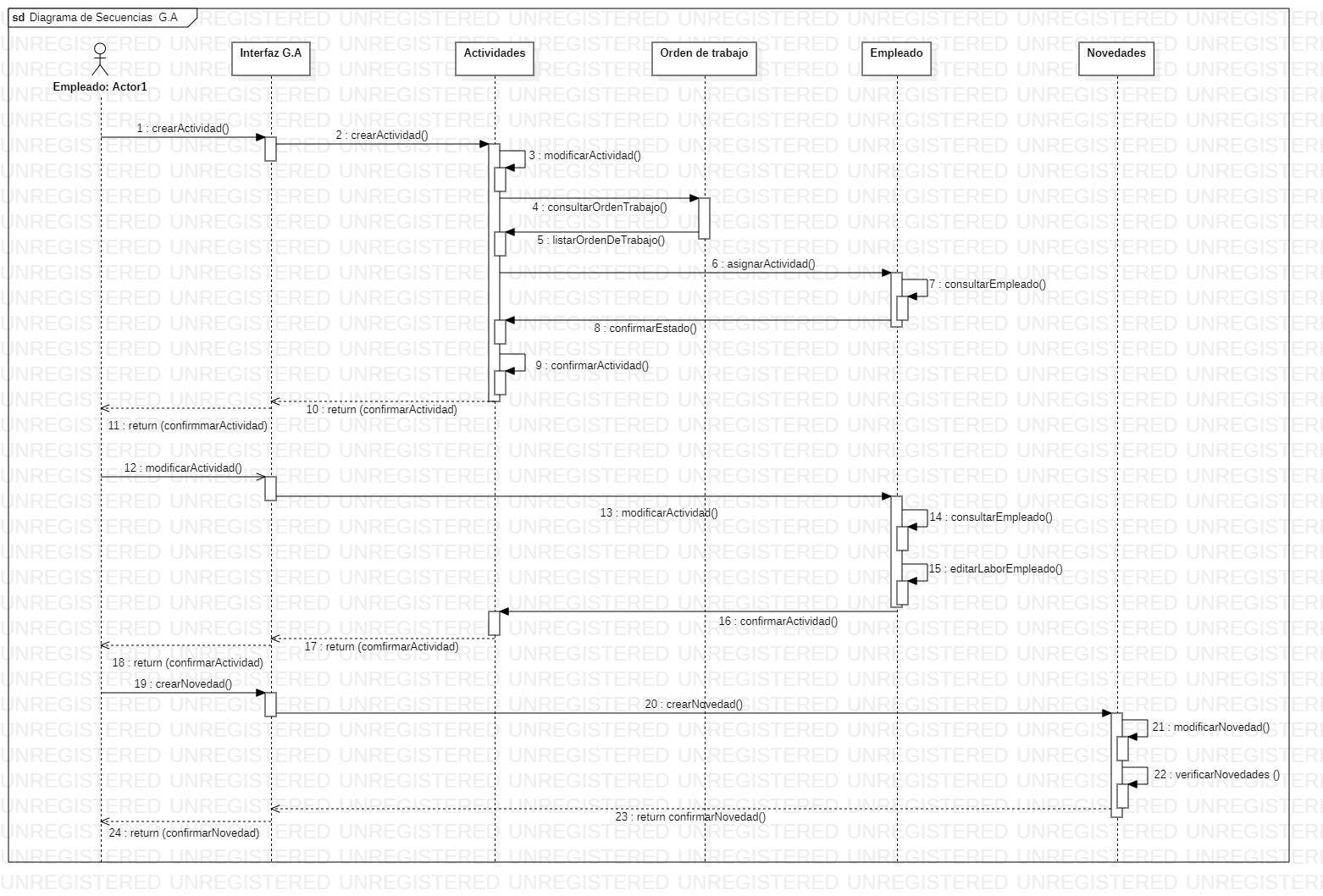
Servicios:



Entradas y salidas OT:

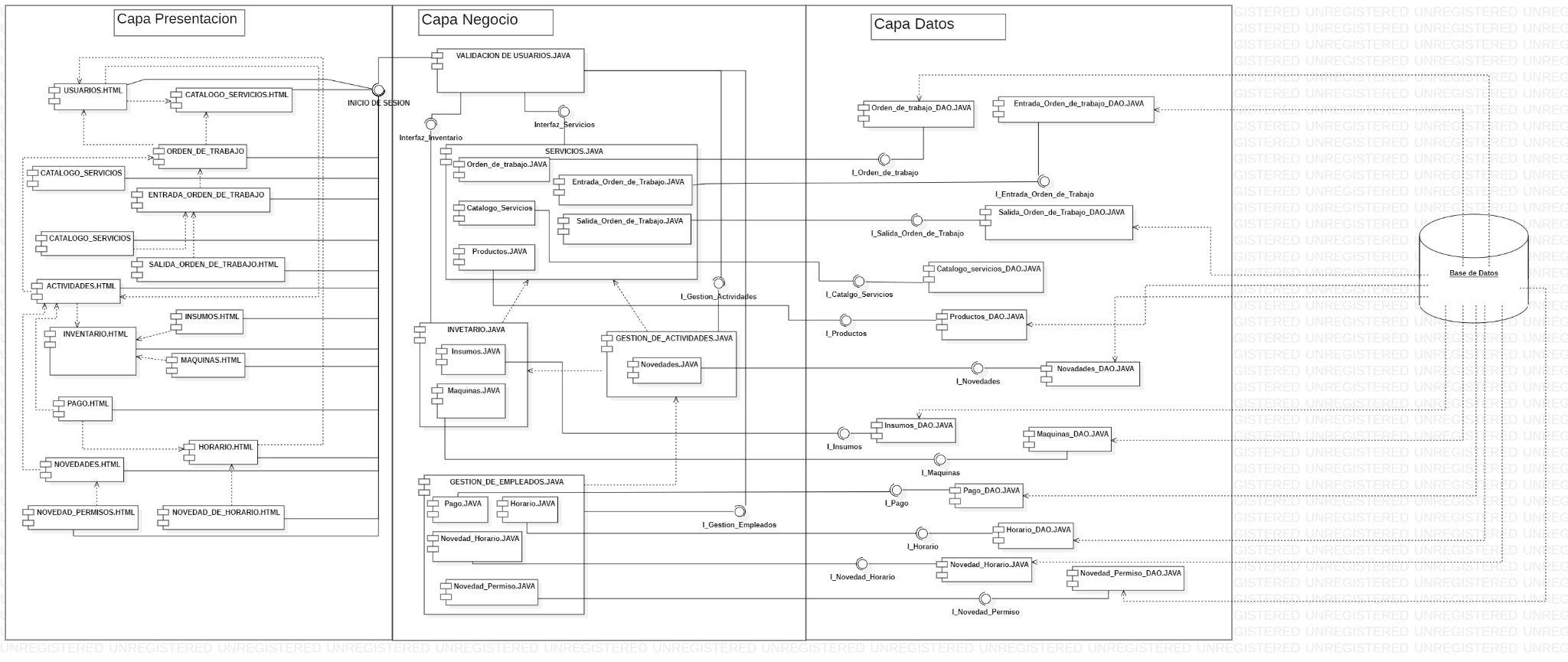


Actividades:

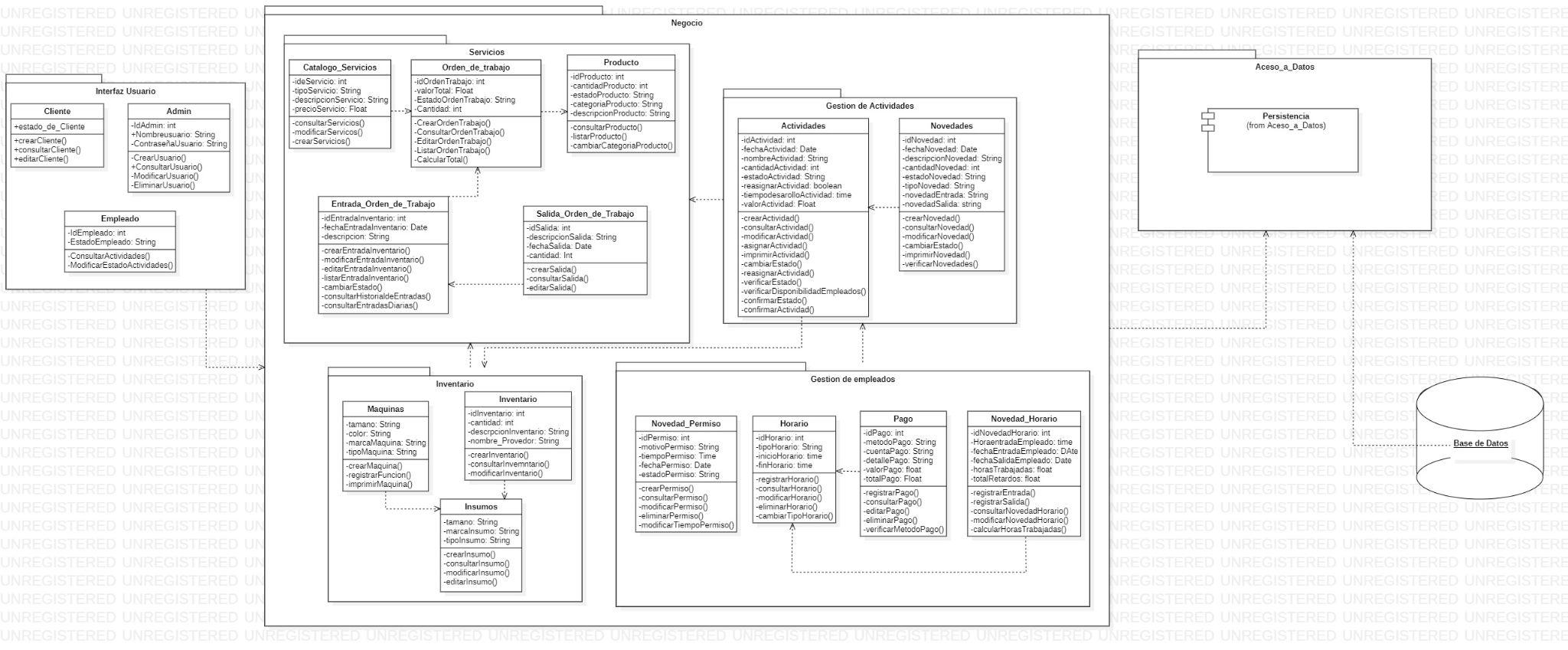


## Vista de Implementación

### Diagrama de Componentes



### Diagrama de Paquetes

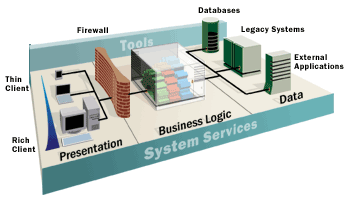


## Vista de Despliegue

### Diagrama de despliegue



# Arquitectura en capas (capas, plataforma)

Se usó la arquitectura en tres capas para poder observar a profundidad todas las funciones y necesidades que tendría el sistema en sus ámbitos que son el software y el hardware.

# Vista de Datos

## Modelo Relacional

# Características Generales de Calidad

## Tamaño y performance

* Tiempo de respuesta en el acceso a la Base de Datos:

-En la base de datos se espera un tiempo de respuesta aproximado de dos segundos.

* Tiempo de respuesta de transacciones:

-Se espera que de igual manera su tiempo de respuesta sea aproximadamente de dos segundos.

* Espacio en disco para el cliente:

-Se espera que el cliente tenga un espacio de un terabyte en el disco duro.

* Espacio en disco para el servidor de Base de datos:

-Se espera que la base de datos cuente con un espacio en el disco duro aproximado de diez gigabytes.

## Usabilidad

El sistema de información cuenta con un programa fácil de manejar, esto lo lleva a tener en cuenta al usuario por que tiene como factor facilidad de aprendizaje.

## Eficiencia

El sistema de información tiene una capacidad de realizar tareas de manera rápida y efectiva, utilizando los recursos de la computadora de manera efectiva, sin retrasos o bloqueos

## Seguridad

Se tendrá en cuenta la gestión de contraseñas para garantizar que las contraseñas y los datos confidenciales de los usuarios estén seguros y protegidos contra posibles amenazas cibernéticas. igualmente implementaremos medidas de protección y prevención para evitar la exposición o el acceso no autorizado a datos confidenciales o información crítica del usuario. Esto puede incluir medidas de seguridad técnicas, como encriptación, autenticación y verificación de identidad, así como políticas de seguridad y procedimientos de gestión de riesgos.

## Confiabilidad

El software es capaz de manejar grandes volúmenes de tráfico y transacciones de manera confiable, sin caídas del sistema o pérdida de datos,el programa tendrá la capacidad de funcionar correctamente y de manera consistente en todo momento, incluso en situaciones imprevistas o de alta demanda.

## Mantenimiento

El software es capaz de corregir errores o fallos en el software. Si los usuarios informan problemas o errores en el programa, el equipo de mantenimiento realiza pruebas y análisis para identificar la causa del problema y corregirlo mediante la actualización del software,actualizar, reparar y optimizar el software existente para garantizar que siga siendo funcional y eficiente. Esto puede incluir actualizaciones de seguridad, corrección de errores, mejoras de rendimiento y compatibilidad, y la adición de nuevas características o funcionalidades.

## Estándares

Las normas ISO 25000, también conocidas como SQuaRE (Software product Quality Requirements and Evaluation), son un conjunto de normas internacionales que garantizan un marco para la evaluación y la gestión de la calidad del software. A continuación, se presentan algunos pasos que pueden ayudar a aplicar las normas ISO 25000 en la calidad del software:

1. Identificar los requisitos de calidad del software: En primer lugar, es importante identificar los requisitos de calidad del software que son relevantes para su proyecto. Los requisitos pueden incluir aspectos como la funcionalidad, la eficiencia, la usabilidad, la confianza, la seguridad y la compatibilidad.
2. Seleccione las métricas de calidad adecuadas: A continuación, debe seleccionar las métricas de calidad adecuadas para medir el grado en que se cumplen los requisitos de calidad. Las métricas pueden incluir.

La norma ISO 9001 es un conjunto de estándares internacionales que establece los requisitos para un sistema de gestión de calidad (SGC) en una organización. A continuación, se presentan algunos pasos que pueden ayudar a aplicar la norma ISO 9001 en la calidad del software:

1. Identificar los requisitos de calidad del software: En primer lugar, es importante identificar los requisitos de calidad del software que son relevantes para su proyecto. Los requisitos pueden incluir aspectos como la funcionalidad, la eficiencia, la usabilidad, la confianza, la seguridad y la compatibilidad.
2. Establecer un sistema de gestión de calidad: Para cumplir con los requisitos de la norma

Las normas ISO 27000 son un conjunto de estándares internacionales que refuerzan las mejores prácticas para la gestión de la seguridad de la información en una organización. A continuación, se presentan algunos pasos que pueden ayudar a aplicar las normas ISO 27000 en la calidad de software:

1. Identificar los requisitos de seguridad de la información del software: En primer lugar, es importante identificar los requisitos de seguridad de la información que son relevantes para su proyecto de software. Los requisitos pueden incluir aspectos como la confidencialidad, integridad y disponibilidad de la información.
2. Realizar una evaluación de riesgos: Para establecer una gestión efectiva de la seguridad de la información en el software, es necesario llevar a cabo una evaluación de riesgos para identificar las amenazas y vulnerabilidades del sistema.
3. Definir las políticas y procedimientos de seguridad: A partir de los resultados de la evaluación de riesgos, se deben definir las políticas y procedimientos de seguridad de la información que se aplicarán en el software.
4. Establecer controles de seguridad: Para proteger la información del software, se deben establecer controles de seguridad apropiados, tales como la autenticación, la gestión de acceso, la encriptación y la monitorización de seguridad.
5. Realizar una auditoría interna: Es importante realizar una auditoría interna periódica para evaluar el cumplimiento de las políticas y procedimientos de seguridad del software, y realizar mejoras si es necesario.