

Documento de Arquitectura Del Software de TecnoHelp

2020

Documento oficial arquitectónico del proyecto TecnoHelp, que tiene como fin brindar una vista macro de la estructura del proyecto que sirva como base para el entendimiento de los deberes del equipo y de la estructura interna para el usuario final.

Control de Cambios

Creación del documento				
Autor	TECNOHELP – TECNO PLAZA		Fecha	13 de marzo del 2020
Revisado por	Nombre	Ana Milena Rojas	Fecha	Seleccione una Fecha
	Cargo	Product Owner		
Aprobado por	Nombre		Fecha	Seleccione una Fecha
	Cargo			
Versión	Descripción	Autor	Aprobado Por	Fecha Aprobación
1.0	Creación del documento	TECNOHELP – TECNO PLAZA		Seleccione una Fecha
				Seleccione una Fecha

TABLA DE CONTENIDO

1.	Introducción	4
1.1.	Propósito	4
1.2.	Alcance	4
1.3.	Definiciones, siglas y abreviaturas	4
1.4.	Referencias	5
1.5.	Vista Global	5
2.	Macro Arquitectura	6
2.1.	Modelo Multicapa	6
3.	Vista Física	8
4.	Vista Funcional o Lógica	9
4.1.	Diagrama de componentes.	9
5.	Vista de Despliegue	10
5.1.	Diagrama de Despliegue.	10
6.	Vista de Procesos	11
6.1.	Diagrama de Procesos.	11

1. Introducción

1.1. Propósito

El propósito de este documento es definir y dar a conocer la Arquitectura de Software, la cual va a ser utilizada para el desarrollo del proyecto “Software de servicio al cliente para la empresa Tiendas Tecno Plaza SAS”. Se utilizan las vistas de proceso y física, para describir la estructura y funcionalidad, así como la distribución física, la descripción y la interacción del sistema.

Este documento va dirigido al equipo de desarrollo, con el fin de proporcionar las bases arquitecturales, además de una visión global y comprensible del diseño, sobre las cuales el proyecto se define para ser llevado a cabo, con el documento podrán comprender la organización de procesos del sistema y ver reflejados la aplicación de requisitos no funcionales tales como la mantenibilidad, el performance y la disponibilidad.

1.2. Alcance

Este documento describe como se va a desarrollar el sistema que permitirá a los colaboradores de la empresa Tiendas Tecno Plaza SAS, gestionar el proceso de garantía, servicio técnico, soporte y pólizas de servicio, logrando así que los colaboradores tengan una herramienta que permita agilizar los procesos que realizan en su día a día. La aplicación es Web y por tal motivo funcionaria en computadores que tengan un navegador Web.

En este documento se especifican las restricciones, los atributos de calidad, las vistas arquitecturales y los patrones de diseño, que se implementarán por parte del equipo de desarrollo, teniendo como referencia el Product Backlog.

1.3. Definiciones, siglas y abreviaturas

DAS: Documento de arquitectura de software.

AngularJS: Es un framework de JavaScript de código abierto.

SQL Server: Software gestor de base de datos.

Sprint Boot: herramienta que nace con la finalidad de simplificar el desarrollo de aplicaciones basadas en el framework Spring Core.

Front-End: término utilizado en desarrollo de aplicaciones para determinar la parte del software que interactúa con los usuarios.

Back-End: término utilizado en desarrollo de aplicaciones para determinar la parte del software que procesa la información del Front-End.

Product Backlog: en los modelos de desarrollo ágil corresponde a el listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.

Product Owner: en los modelos de desarrollo ágil corresponde a la persona interesada en el proyecto (cliente).

Modelo Multicapa (Modelo n-tier): Es un modelo de arquitectura de software que separa los datos de una aplicación, interfaz de usuario y la lógica del negocio.

JWT (JSON Web Token): es un estándar abierto basado en JSON para la creación de tokens de acceso que permiten la propagación de identidad y privilegios.

Diagrama Informal: Diagrama que no surge de ningún modelo preestablecido.

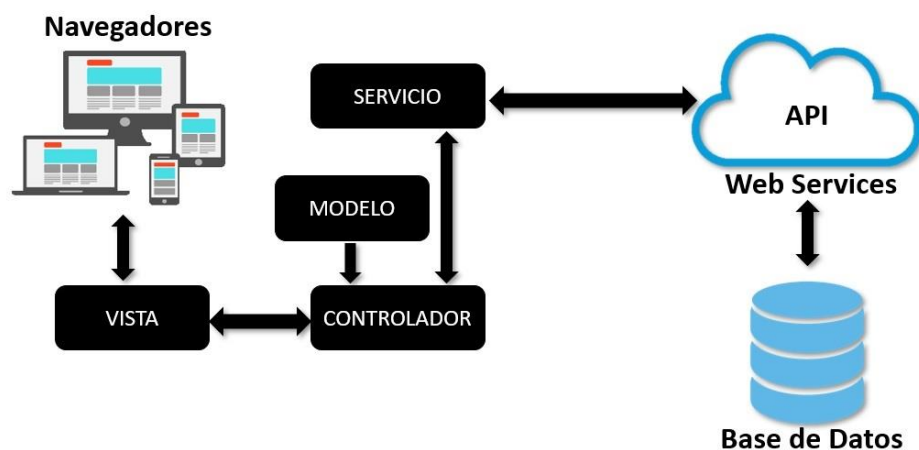
1.4. Referencias

Documento	Versión	Fecha de la versión
Product Backlog	1.0	13/03/2020

1.5. Vista Global

En el documento se encontrará una macro arquitectura, con especificaciones de las vistas usadas en el proyecto, a continuación, una lista de metas y restricciones arquitectónicas que mostrarán los atributos de calidad observables y no observables que tiene el equipo del área de servicio técnico, garantías y póliza de servicio de Tiendas Tecno Plaza SAS para con el proyecto, por último, las diferentes vistas con sus respectivos diagramas, que nos ayudarán a encontrar la comprensión deseada.

Modelo Multicapa



2. Macro Arquitectura

La macro arquitectura que comprende el sistema actual se basa en 3 vistas claves, la vista lógica, física y de procesos que comprenderán los siguientes diagramas.

- **Vista lógica:** Diagrama de componentes, debido a la facilidad que tiene junto con su estructura el permitir un análisis completo y a fondo de las diversas etapas de desarrollo de un aplicativo software, expresando las relaciones que se obtendrán las distintas dependencias de un componente.
- **Vista Física:** Diagrama informal, el cual permite mostrar la interacción física necesaria y usada entre los dispositivos que incorpora el proyecto. Diagrama de Despliegue, ya que muestra la arquitectura del sistema desde el punto de vista de la distribución y debido a esto permitirá la toma de decisiones sobre la conectividad de red de los clientes a los servidores.
- **Vista de Procesos:** Diagrama de actividades, Por su capacidad para ilustrar las acciones internas de un proceso, capturar la especificación de un caso de uso o historia de usuario y reflejarlos en los flujos de proceso del negocio.

2.1. Modelo Multicapa

Denominado también 3-tier, es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario en varios componentes distintos. Donde el principal objetivo es dividir el software en capas cumpliendo cada capa en una función específica siendo así un sistema modular, ofreciendo una mejor mantenibilidad, alta escalabilidad, y flexibilidad al cambio.

2.2. Metas y Restricciones Arquitectónicas

Atributos de Calidad “Observables”			
Atributo de Calidad	Descripción	Tácticas /Patrón de Arquitectura	Donde se aplica
Confidencialidad Autorizador (subcategoría seguridad)	Capacidad del software de asegurar que cualquiera de sus características y/o contenidos estén enmascarados u ocultos de entidades no autorizadas. Asignar los tipos de acceso a los usuarios según sus roles en el sistema	RBAC (rol based access control): Basado en el principio de acceso con el menor privilegio: Un rol sólo concede los mínimos privilegios requeridos por un individuo para realizar un trabajo o acción.	Se aplica en el sistema al momento de identificar el rol en el logueo de la aplicación, asignando los privilegios de uso (autorización) de una manera ágil y flexible.
Funcionalidad - subcategoría Interoperabilidad	Permitir estandarización de componentes que fueron desarrollados separadamente para luego integrarlos	Programación por Capas	separar el Diagrama de Clases usando el patrón de desarrollo multicapa

Performance	Tiempo que requiere el sistema para responder a un evento o estímulo (tiempo de respuesta), o bien el número de eventos procesados en un intervalo de tiempo (throughput).	Singleton: garantiza que el control total de la aplicación la tiene el programador y no el cliente y que el mismo puede recurrir a usar un TemplateMetod si es necesario que garantice la reutilización del código y la adaptación a nuevas necesidades.	Se aplica en la creación de los objetos y en los Constructores del código del programa en los microservicios.
--------------------	--	---	---

Atributos de Calidad “No observables”			
Atributo	Descripción	Tácticas /Patrón de Arquitectura	Donde se aplica
Modificabilidad	Es la habilidad de realizar cambios futuros al sistema.	Patrón: GRAS Planteamiento modular del sistema, división por capas lógicas de software con responsabilidades definidas de cada capa y protocolos de comunicación entre ellas utilizado en la construcción de los microservicios.	Crea un puente de comunicación entre el protocolo HTTP síncrono y las aplicaciones existentes con interfaz que controlan la información entrante y saliente que garantiza la granularidad del sistema y la no pérdida de la información.
Mantenibilidad	Es la capacidad de someter a un sistema a reparaciones y evolución Capacidad de modificar el sistema de manera rápida y a bajo costo.	Patrón: Capas de software Mediante este patrón permite que las modificaciones a los servicios web de integración sean más sencillas, dado que van a puntos específicos del código. Estandarización en el nombramiento de tablas, módulos, clases, métodos de clase y variables a nivel de diseño y programación.	Pretende a disminuir la alta cohesión de componentes para garantizar que todo el sistema es más dinámico y adaptable a las necesidades de expansión según lo requiera el sistema.
Portabilidad	Capacidad de transferencia del producto de software de un	Responsive design: garantiza adaptar tu diseño al tamaño de pantalla del dispositivo.	Brindarle al usuario la Una forma de poder ingresar a la página desde un PC con

	entorno a otro. El cual puede ser de tipo organizacional, hardware o software		diferentes resoluciones de pantalla.
--	---	--	--------------------------------------

3. Vista Física

Diagrama informal de la vista física: Es un diagrama que describe las características físicas de los elementos que componen la arquitectura del proyecto.

El Diagrama informal de la vista física contempla el uso del aplicativo por parte del usuario que inicialmente ingresa al navegador desde un dispositivo ya sea computador de escritorio hasta dispositivos móviles, el usuario ingresara a la página Web del software Soporte Tecno Plaza que está alojado en un servidor de aplicaciones Internet Information Services 10.0 de igual forma el servidor tiene como gestor de bases de datos SQL Server Standard 2019.

Los atributos de calidad presentes son:

La escalabilidad es la capacidad de hacer crecer un sistema cuando se incrementa la carga de trabajo (el número de peticiones). Cada máquina tiene una capacidad finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones. e incorpora el patrón de diseño multicapa ya que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario, de este modo cada vez que se requiera implementar un nuevo módulo o funcionalidad al aplicativo se manipulara por sus capas.

Portabilidad: El aplicativo web Mesa de Ayuda es portable ya que implementa el patrón de diseño Responsive Design y se puede adaptar a diferentes dispositivos con múltiples tamaños de pantalla de computadores.

Notación del diagrama informal físico:

Usuarios: Persona que habitualmente usa el servicio.

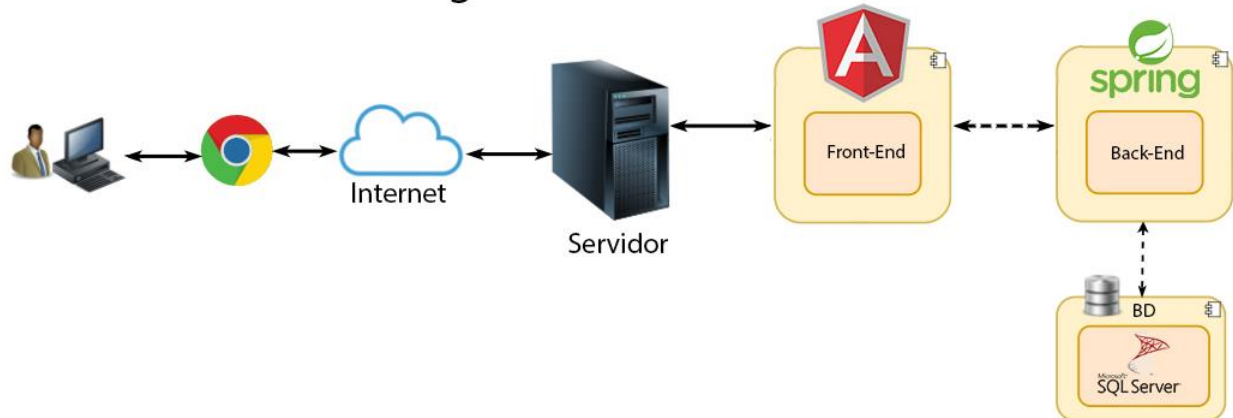
Computador (PC o portátil): Dispositivo electrónico capaz de procesar datos que llegan desde el exterior (entradas) y proporcionan un resultado (salida). El cual tiene conexión a red permanente o intermitente a una red de datos.

Internet: Unión de todas las redes y computadoras distribuidas por todo el mundo, en otras palabras, es una red global en que se conjuntan todas las redes que utilizan el protocolo TCP/IP y que son compatibles entre sí.

Servidor: Es un ordenador encargado de suministrar información a una serie de clientes, que puede ser tanto personas como otros dispositivos conectados a él. la información que se puede transferir es múltiple y variada, desde archivos de texto, imagen, vídeo y programas informáticos, base de datos, entre otros.

Aplicativo web: Son aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet mediante un navegador.

Diagrama Informal Fisico



4. Vista Funcional o Lógica

4.1. Diagrama de componentes.

El siguiente diagrama representa el marco de trabajo de la relación entre las diferentes dependencias del framework AngularJS y los componentes que se encuentran dentro de esta arquitectura para la funcionalidad del aplicativo web. En este diagrama se ve reflejado que el sistema es multicapa representado por la capa de presentación, capa de lógica del negocio, y capa de acceso a datos, donde al ser un sistema multicapa nos permite realizar de manera óptima y eficiente la mantenibilidad ya que se puede dar soporte o atención a un punto específico sin necesidad de tocar el resto del sistema, la modificabilidad del sistema permitiendo incluir componentes en una línea de tiempo sin que afecte lo que ya existe, como también permite mayor funcionalidad.

En la capa de presentación se utiliza el framework de diseño Bootstrap y plantillas Angular CLI permitiendo hacer un diseño web responsivo y portable para diferentes plataformas de computadores y que tengan un navegador Web.

En la capa de lógica de negocio se utiliza un ruteador que redirecciona a un controlador donde éste a su vez tiene unas funciones específicas de acuerdo con la ruta invocada, dentro del controlador tenemos funciones propias del marco de trabajo y la lógica de procesos que ejecuta el sistema.

En la capa de acceso a datos del Spring Boot se crea el proceso de microservicios con acceso directo a la base de datos. En las entidades se implementa el patrón de diseño Singleton permitiendo mayor performance y concurrencia en las peticiones a los servicios de la aplicación dando prioridad a una petición.

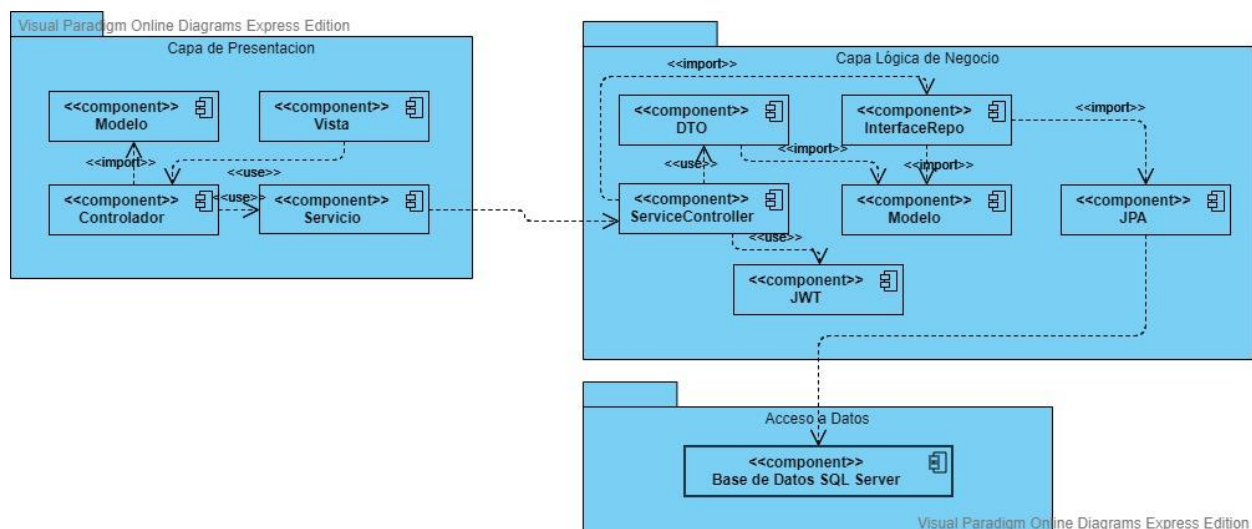
Los atributos de calidad presentes son:

Mantenibilidad propende a disminuir la alta cohesión de componentes para garantizar que todo el sistema sea más dinámico y adaptable a las necesidades de expansión según lo requiera. Además de dedicar recursos necesarios a cada una de las capas y permitir un procesamiento en paralelo de las responsabilidades de estas, permitiendo que sea manejable en caso una posible

expansión del sistema mediante el patrón multicapa el cual tendrá efecto en todo el aplicativo web.

Modificabilidad se aplica en la separación de las tareas, permitiendo que la aplicación sea ligera y entendible, además de que se puede realizar cambios en una parte de la aplicación sin afectar los demás componentes mediante el patrón multicapa.

Funcionalidad se aplica en todo el aplicativo la estandarización de componentes que se desarrollaron de forma separados por capas para ser integrados con otros, y así tener subcategorías y la interoperabilidad del aplicativo.



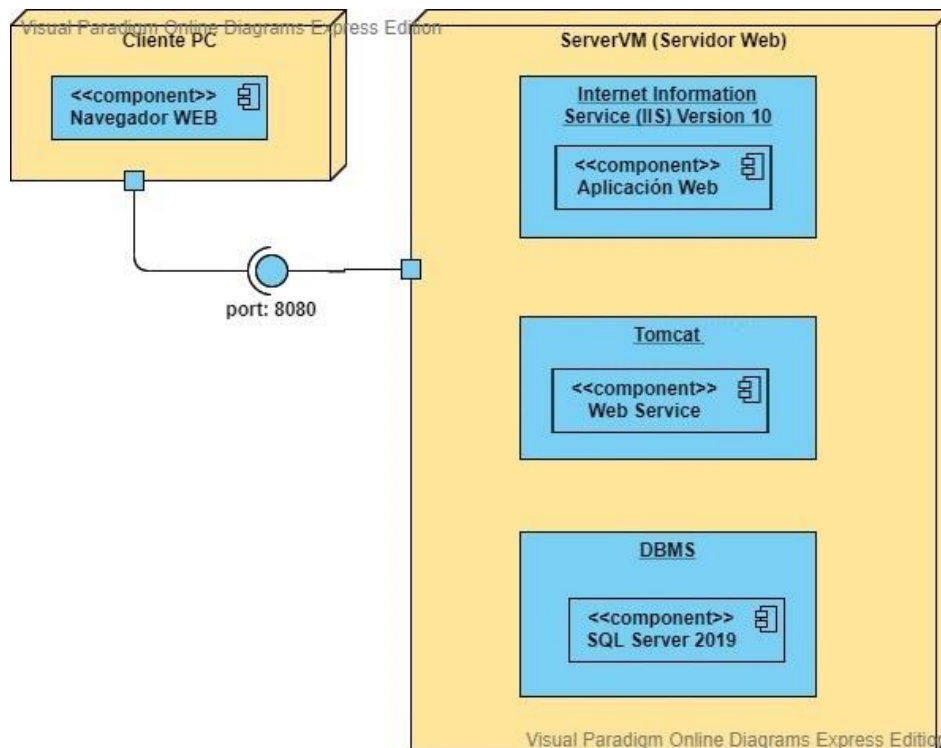
5. Vista de Despliegue

5.1. Diagrama de Despliegue.

Este diagrama describe las configuraciones de redes físicas, sobre las cuales el software será desarrollado. Se indican los nodos físicos que interactúan con la aplicación Web tales como un nodo para el dispositivo donde se accede la aplicación web en un PC con acceso a Internet y un navegador. Éste envía las peticiones por HTTP para mayor seguridad de información accediendo al nodo del Web Server donde aloja el Internet Information Service (IIS) HTTP Server y El DBMS con el componente de base de datos SQL Server Standard 2019, dentro del Internet Information Service (IIS) tenemos la funcionalidad de la aplicación dividida por capas tales como la capa de presentación, lógica del negocio y acceso a datos, permitiéndonos tener mejor mantenibilidad, modificabilidad y funcionalidad. Por otra parte, la lógica de negocios integra las interfaces de Apis la cual sería los Backend de la lógica del negocio creada por microservicios.

Los atributos de calidad presentes son:

Modificabilidad se aplica en la separación de las tareas, permitiendo que la aplicación sea ligera y entendible, además de que se puede realizar cambios en una parte de la aplicación sin afectar los demás componentes mediante el patrón multicapa.



6. Vista de Procesos

6.1. Diagrama de Procesos.

En esta vista se muestran los procesos que hay en el sistema y la forma en la que se comunican estos procesos; es decir, se representa desde la perspectiva de un integrador de sistemas, el flujo de trabajo paso a paso de negocio y operacionales de los componentes que conforman el sistema. Para completar la documentación de esta vista se puede incluir el diagrama de actividad de UML.

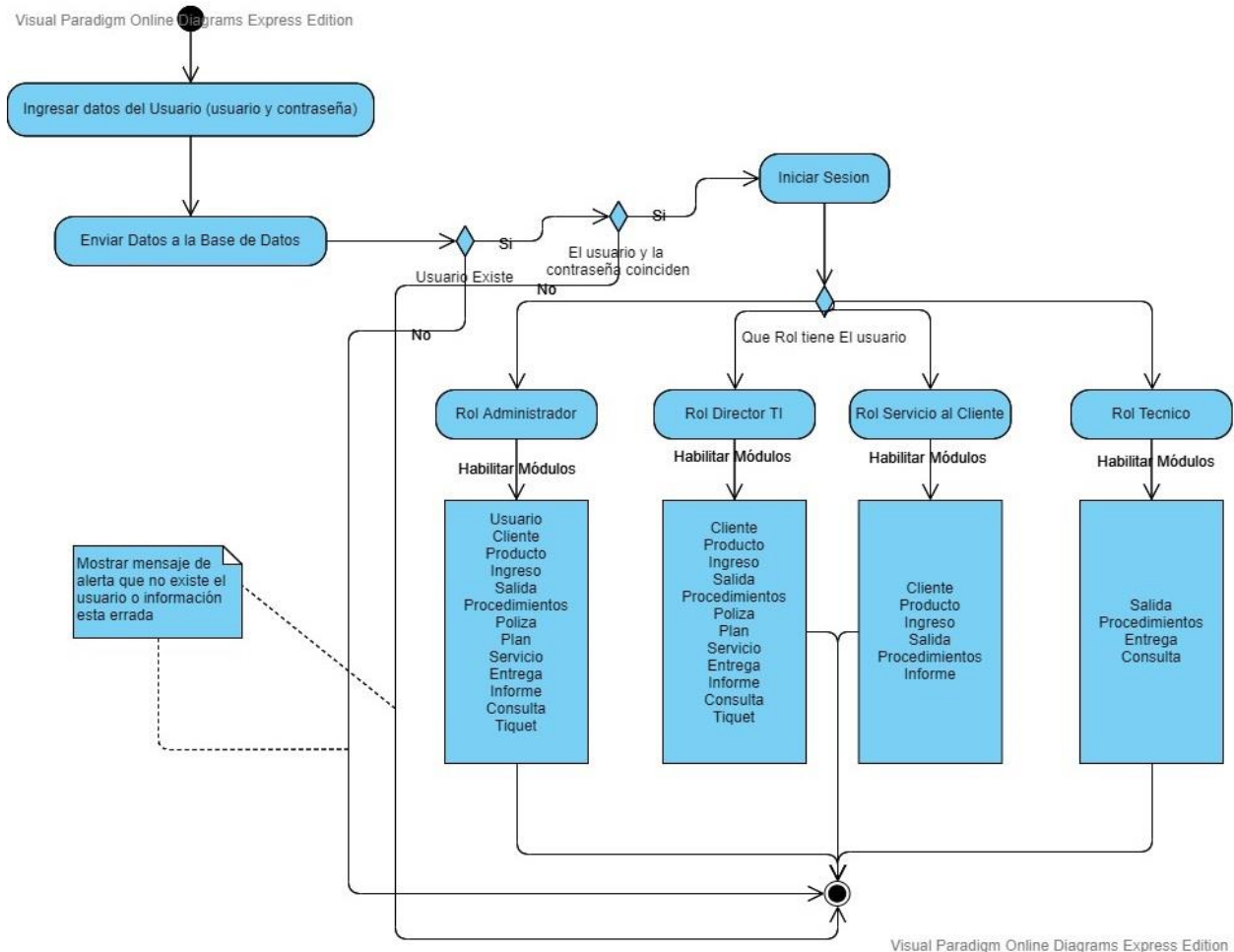
El diagrama de actividades muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Las personas, los componentes de software o los equipos pueden realizar estas acciones.

Los atributos de calidad presentes son:

Confidencialidad asegura la capacidad del software de asegurar que cualquiera de sus características y/o contenidos estén enmascarados u ocultos de entidades no autorizadas utilizando el patrón RBAC (rol based access control): Basado en el principio de acceso con el menor privilegio dicho patrón se utilizará en todo proceso que requiera algún tipo de acceso a los datos de la aplicación, adicionalmente se aplica para la autenticación JWT y así garantizar la seguridad del sistema.

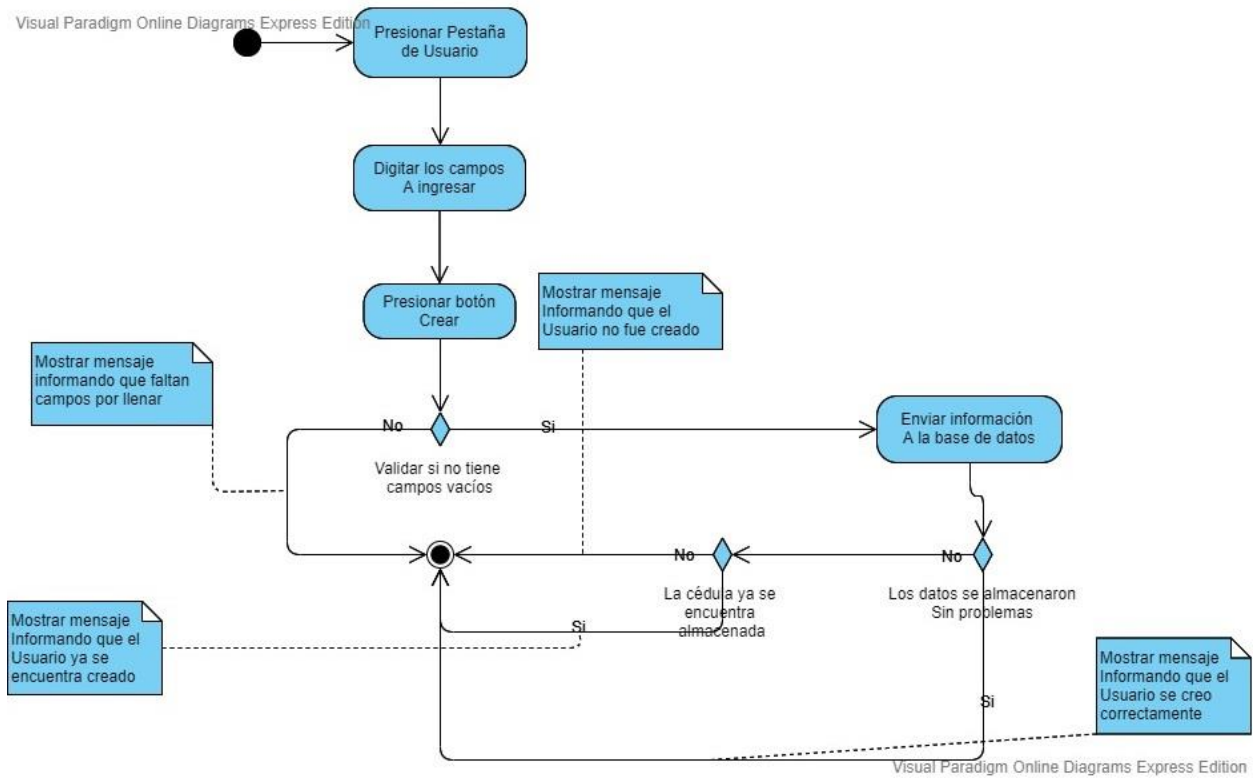
El diagrama de actividades (Autenticación) describe cómo el usuario inicia sesión y el sistema valida los privilegios que tiene el usuario para así habilitar los módulos correctos.

Diagrama de Actividades: Autenticacion



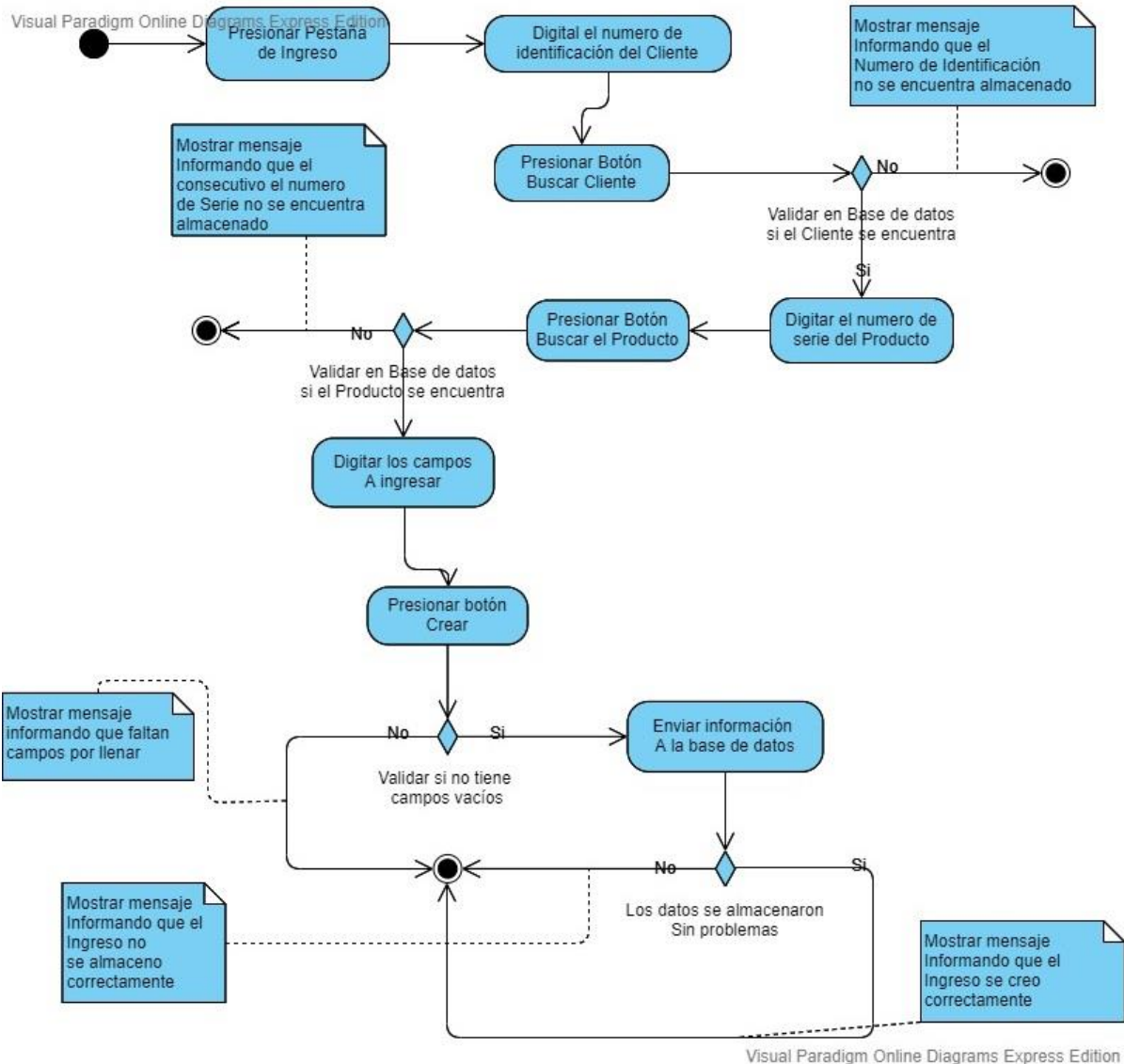
El diagrama de actividades (Crear Usuario) describe cómo el usuario realiza el proceso de creación de un Usuario, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Usuario



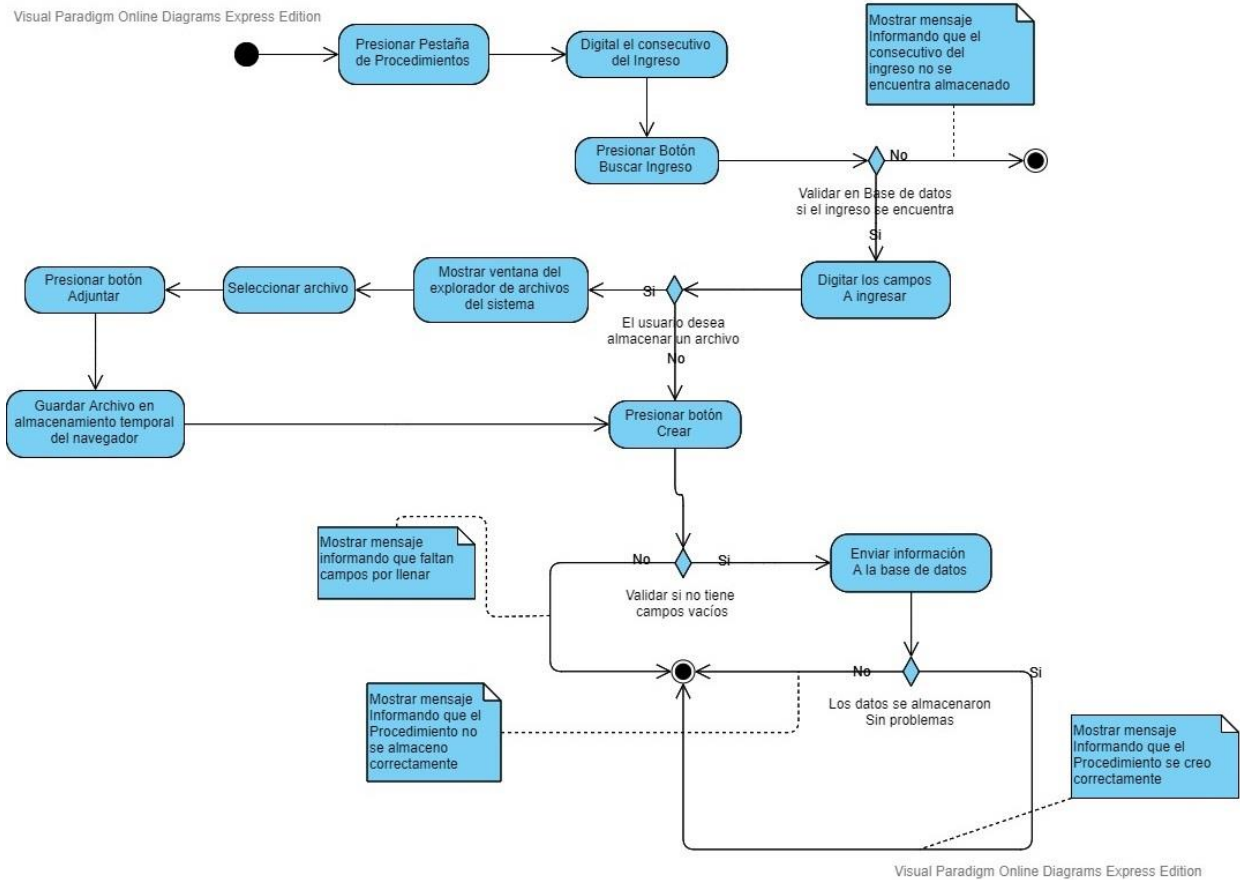
El diagrama de actividades (Crear Ingreso) describe cómo el usuario realiza el proceso de creación de un ingreso, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Ingreso



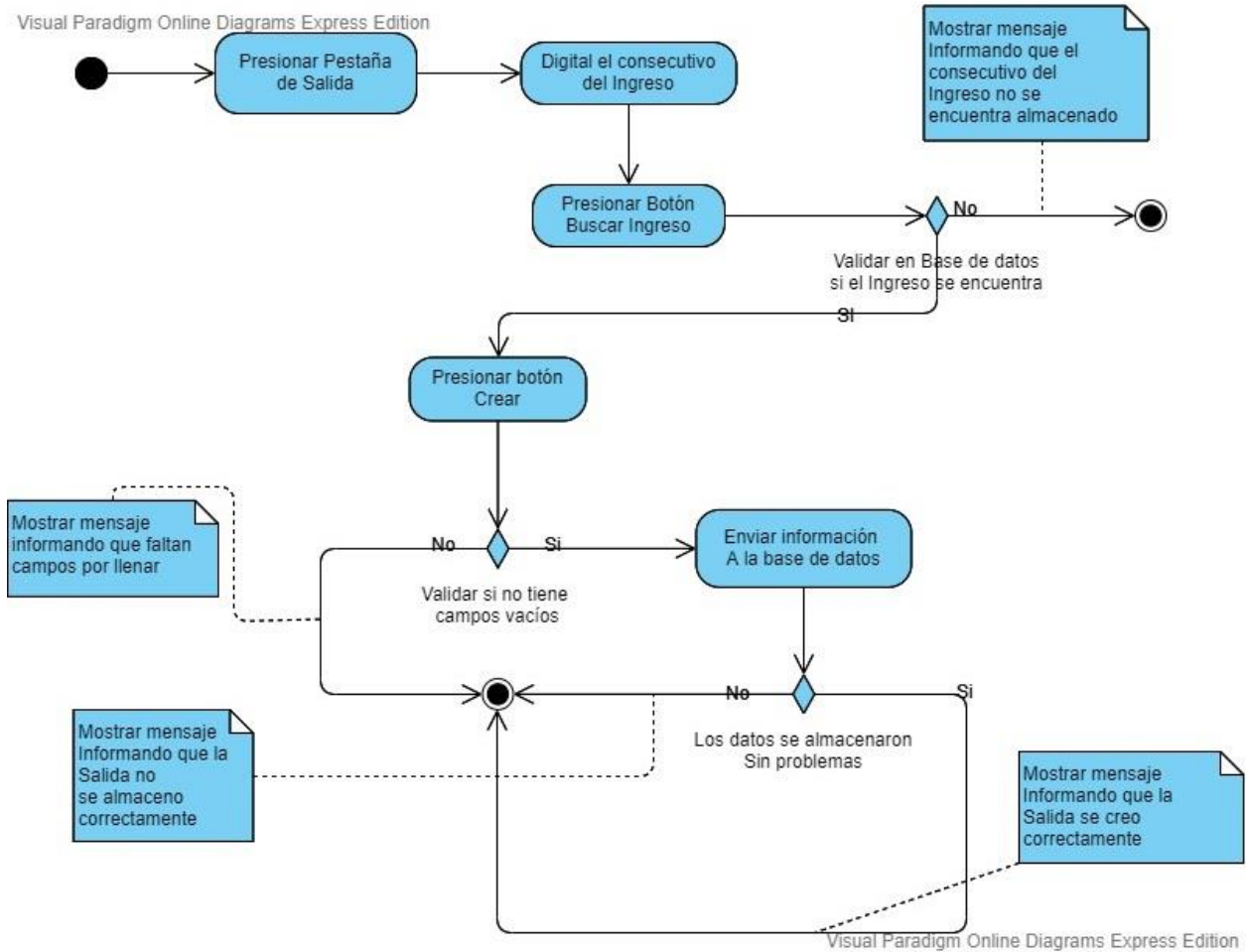
El diagrama de actividades (Crear Procedimiento de un ingreso) describe cómo el usuario realiza el proceso de creación de un procedimiento a un ingreso el cual tuvo que ser creado previamente, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Procedimiento - Ingreso



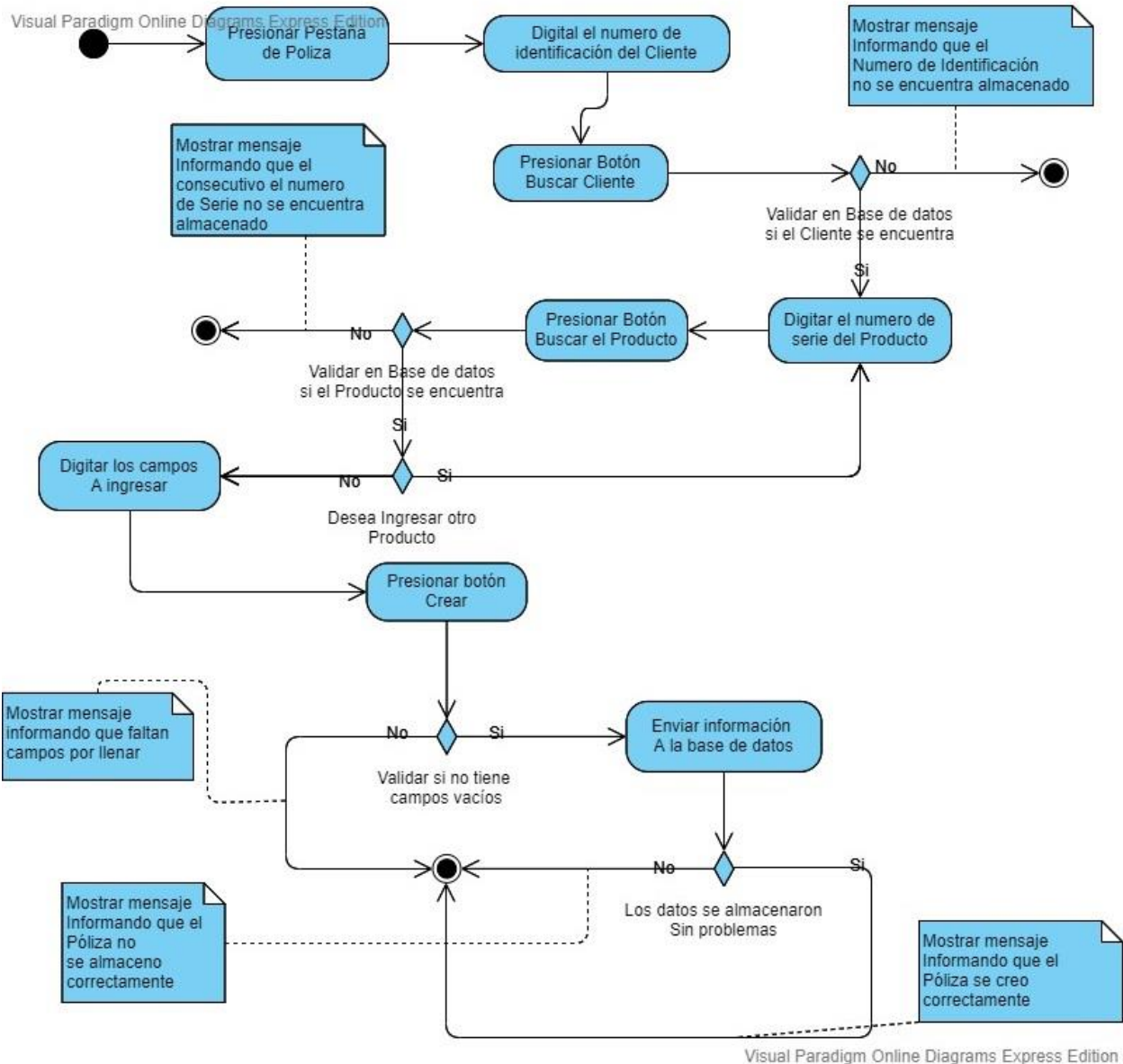
El diagrama de actividades (Crear Salida) describe cómo el usuario realiza el proceso de creación de una Salida de un ingreso el cual tuvo que ser creado previamente para realizar el cambio de estado del ingreso, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Salida



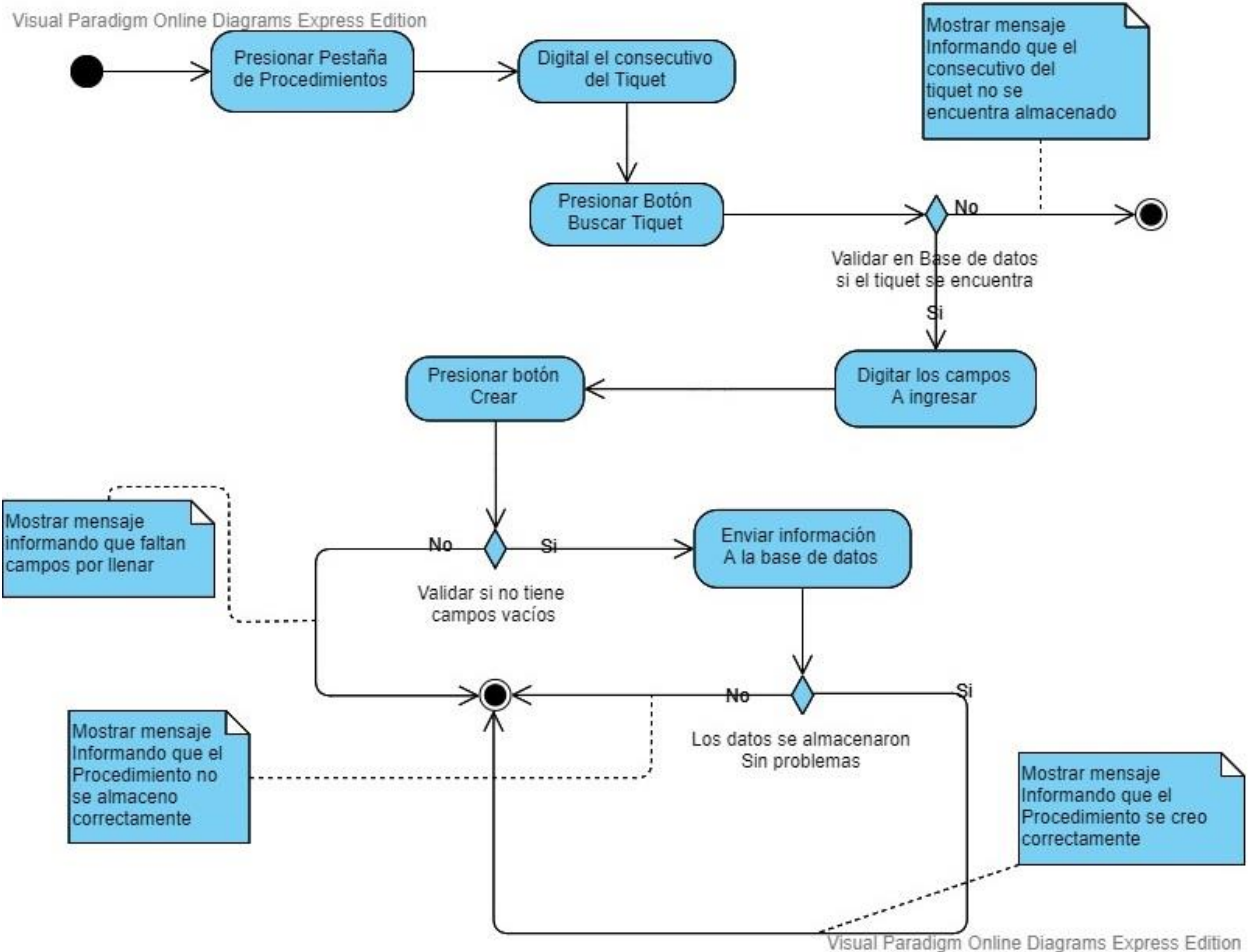
El diagrama de actividades (Crear Póliza) describe cómo el usuario realiza el proceso de creación de una póliza de servicio donde previamente se debió crear un cliente y como mínimo 1 producto para ser agregados a la póliza de servicio, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Poliza



El diagrama de actividades (Crear Procedimiento de un Tiquet) describe cómo el usuario realiza el proceso de creación de un procedimiento a un tiquet el cual tuvo que ser creado previamente, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Procedimiento Tiquet



El diagrama de actividades (Crear Entrega) describe cómo el usuario realiza el proceso de creación de una Entrega de un tiquet el cual tuvo que ser creado previamente para realizar el cambio de estado del tiquet, donde previamente a iniciado sesión y el sistema validado si el usuario tiene el privilegio correcto.

Diagrama de Actividades: Crear Entrega

Visual Paradigm Online Diagrams Express Edition

