**Casos de USO, UML**

**Jorge Armando Toloza Carvajal**

**Análisis y desarrollo de software**

**Ficha # 2627106**

**Instructor: CARLOS ERNESTO LIZARAZO SIERRA**

**SENA**

**Regional Santa fe de Antioquia**

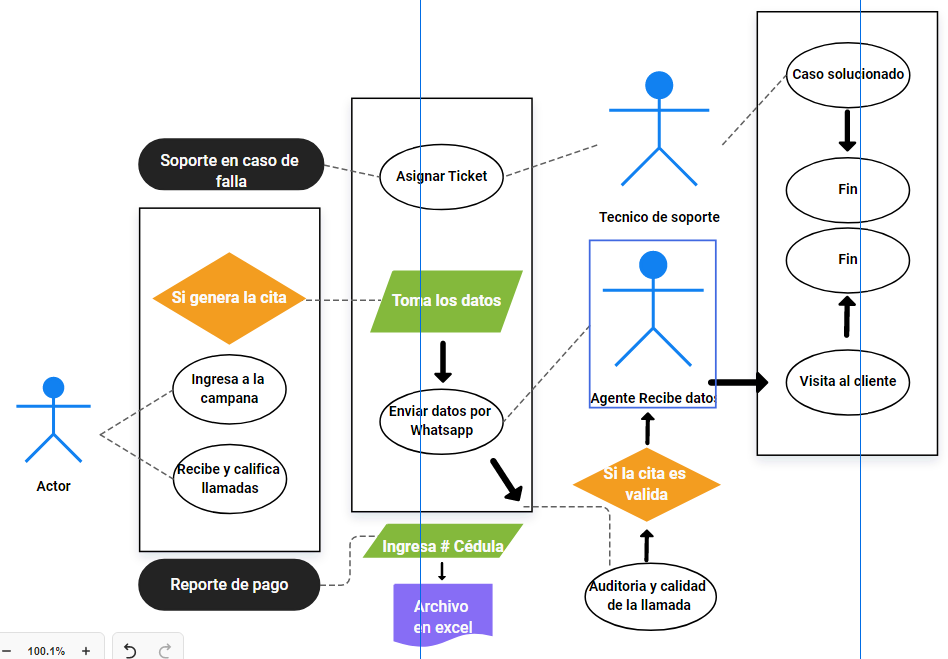
**Marzo 2023**

**OBJETIVO**: Vamos a diagramar el funcionamiento del software para entenderlo e identificar dónde se realizarán los ajustes que se requieren, para cumplir dicho objetivo utilizaremos diferentes diagramas **y estándares para el desarrollo de software**

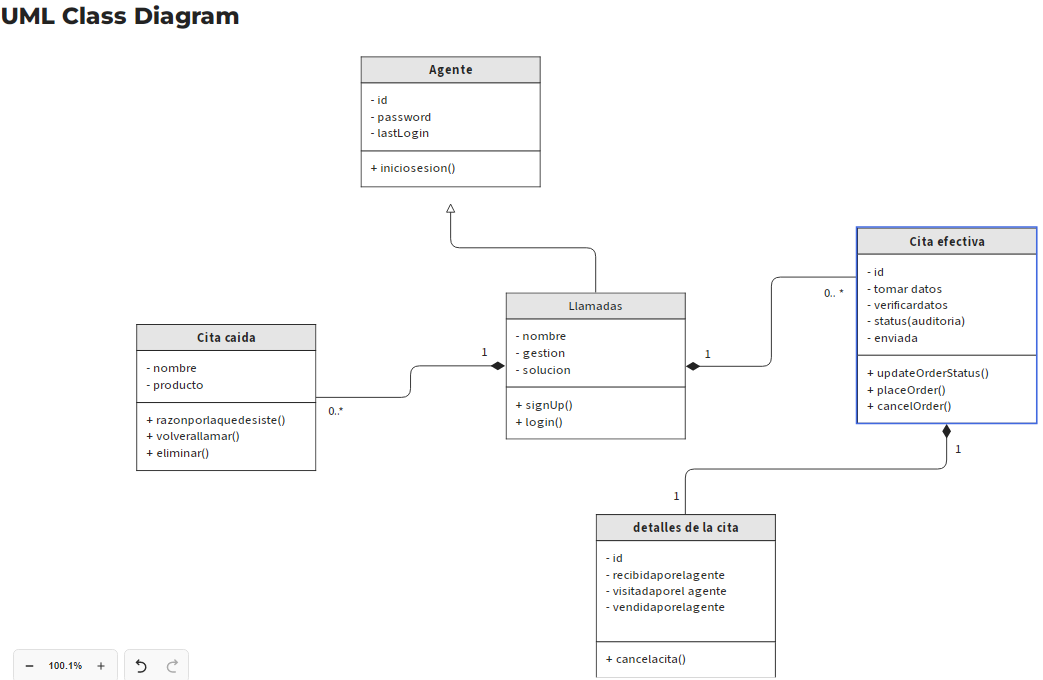
|  |  |  |
| --- | --- | --- |
| 1. **Casos de uso** | **CALLHOUS – CENTRO DE LLAMADAS** | |
| **Descripción** | En esta descripción vamos registrar todo el proceso desde que el asesor se conecta hasta que genera una cita o finaliza sesión | |
| **Pre condición** | Cada asesor tiene un usuario y contraseña que están agregados como favoritos en el navegador para que pueda conectarse al sistema | |
|  | **Paso** |  |
| **1** | El asesor ingresa al navegador y pincha sobre la pestaña guardada bajo el nombre AGT y su número de usuario |
| **2** | Al darle clic mostrara el usuario y contraseña que están previamente guardados y podrá dar clic en la pestaña submit para conectarse |
| **3** | Elegir la campaña dando clic sobre ADD ALL + SUBMIT |
| **4** | Quitar el pause para estar activo y disponible para recibir llamadas. |
| 5 | Recibir llamadas y calificarlas según corresponda |
| **Post condición** | **A continuación, presentaremos el proceso solo para el caso en que el asesor logre captar un cliente** | |
|  | **paso** |  |
|  | **1** | Una vez el asesor logre captar al cliente y darle a conocer los beneficios del producto, ya no deberá grabar manual, porque ahora el sistema grabara automáticamente todas las llamadas después de 60 segundos |
|  | **2** | Si en la llamada el asesor logra convencer al cliente dispondrá de la pestaña ingresar cita y al pinchar se desplegará el formulario para el ingreso de los datos |
|  | **4** | Una ves tome los datos del cliente se activará una pestaña para enviar la cita, con grabación, oprimiendo sobre el botón WhatsApp |
|  | **5** | Cuando el asesor desee ver la nomina podrá hacerlo pinchando en el botón pagos |

**2. Diagrama de casos de uso**

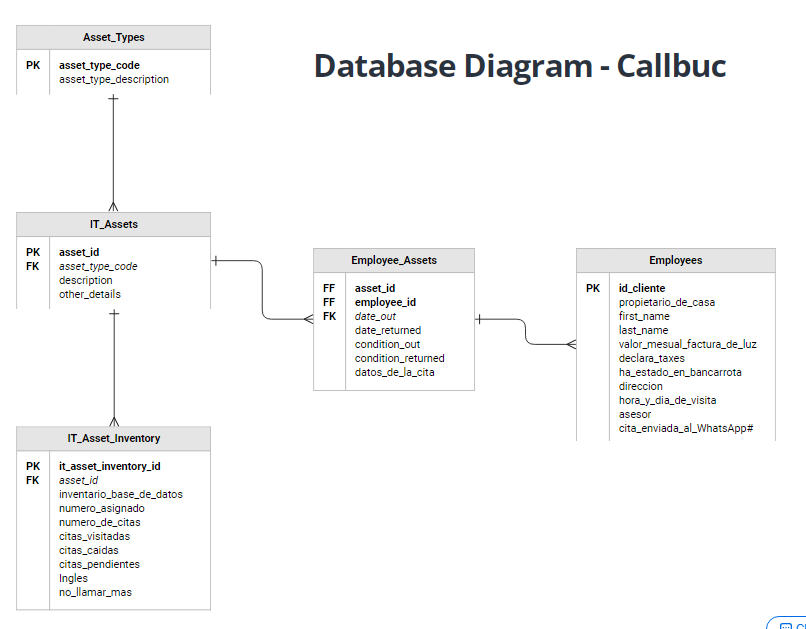
A continuación, presentaremos un diagrama que nos permitirá entender de forma sencilla el funcionamiento del programa una vez este completo



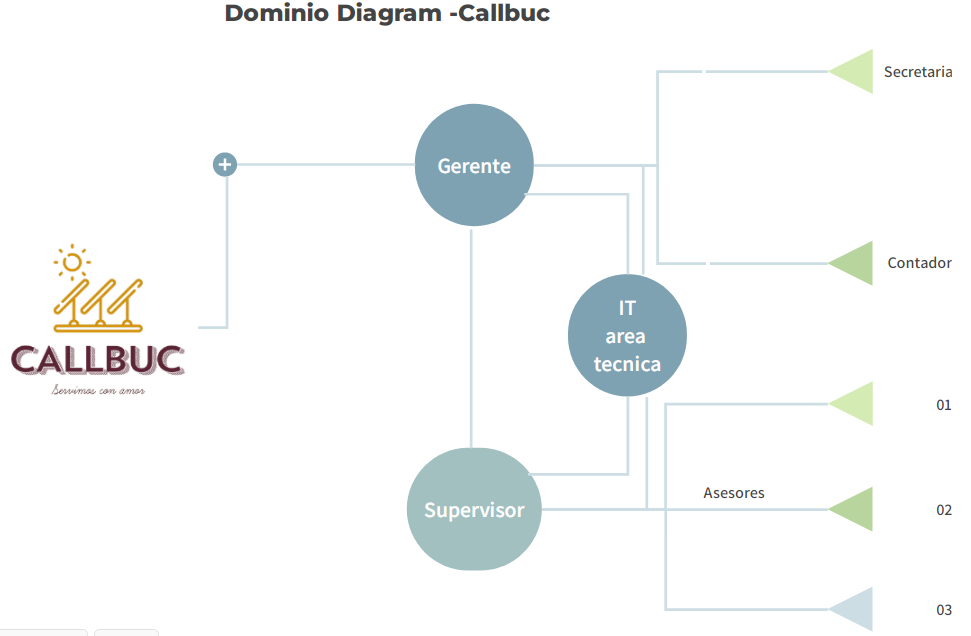
3. Diagrama de clases UML



1. Modelo de bases de datos



1. **Modelo de dominio**



3. ¿Cuál es la importancia de UML?

UML (Unified Modeling Language) es un lenguaje de modelado de software estándar utilizado para visualizar, especificar, construir y documentar los artefactos de software. UML es una herramienta de modelado esencial para los desarrolladores de software, ya que les permite crear modelos detallados de los sistemas de software que están diseñando. A continuación, se presentan algunas de las razones por las que UML es importante en el desarrollo de software:

**Comunicación efectiva:** UML proporciona un lenguaje común para que los desarrolladores, los clientes y otros interesados en el proyecto puedan comunicarse de manera efectiva y comprender los requerimientos y el diseño del sistema de software.

**Estandarización**: UML es un estándar aceptado internacionalmente, lo que significa que los modelos UML creados por diferentes equipos y organizaciones son compatibles entre sí.

**Modelado visua**l: UML permite a los desarrolladores crear modelos visuales de los sistemas de software, lo que facilita la comprensión de los sistemas complejos y su diseño.

**Mejora de la calidad del software**: UML puede ayudar a los desarrolladores a identificar problemas de diseño y otros errores temprano en el proceso de desarrollo, lo que puede mejorar la calidad del software y reducir los costos de corrección de errores.

**Reutilización de código:** UML puede ayudar a los desarrolladores a identificar y diseñar componentes de software reutilizables, lo que puede ahorrar tiempo y esfuerzo en proyectos futuros.

En resumen, UML es una herramienta de modelado de software esencial que ayuda a los desarrolladores a diseñar sistemas de software de manera efectiva y eficiente, mejorar la calidad del software y reducir los costos de desarrollo.

1. ¿Cuáles son los diagramas que pueden representarse con UML?

**Diagrama de casos de uso**: Este diagrama se utiliza para representar los escenarios o casos de uso que un sistema debe soportar.

**Diagrama de clases**: Este diagrama se utiliza para representar las clases del sistema, sus atributos, métodos, relaciones y herencias.

**Diagrama de secuencia:** Este diagrama se utiliza para representar la secuencia de interacciones entre objetos del sistema durante un escenario de caso de uso.

**Diagrama de colaboración:** Este diagrama se utiliza para representar la estructura de los objetos y las interacciones entre ellos durante un escenario de caso de uso.

**Diagrama de actividades**: Este diagrama se utiliza para representar el flujo de trabajo o procesamiento en un sistema.

**Diagrama de componentes**: Este diagrama se utiliza para representar los componentes del sistema, sus dependencias y relaciones.

**Diagrama de despliegue:** Este diagrama se utiliza para representar la arquitectura física del sistema y cómo se despliegan los componentes en los nodos de hardware.

**Diagrama de estados**: Este diagrama se utiliza para representar los estados y transiciones de un objeto del sistema.

**Diagrama de casos de uso**: Este diagrama se utiliza para representar los escenarios o casos de uso que un sistema debe soportar.

**Diagrama de clases**: Este diagrama se utiliza para representar las clases del sistema, sus atributos, métodos, relaciones y herencias.

**Diagrama de secuencia:** Este diagrama se utiliza para representar la secuencia de interacciones entre objetos del sistema durante un escenario de caso de uso.

**Diagrama de colaboración:** Este diagrama se utiliza para representar la estructura de los objetos y las interacciones entre ellos durante un escenario de caso de uso.

**Diagrama de actividades**: Este diagrama se utiliza para representar el flujo de trabajo o procesamiento en un sistema.

Diagrama de componentes: Este diagrama se utiliza para representar los componentes del sistema, sus dependencias y relaciones.

**Diagrama de despliegue:** Este diagrama se utiliza para representar la arquitectura física del sistema y cómo se despliegan los componentes en los nodos de hardware.

**Diagrama de estados**: Este diagrama se utiliza para representar los estados y transiciones de un objeto del sistema.

1. ¿Qué herramientas de software existen en el mercado para UML?

**Visual Paradigm**: es una herramienta de modelado de software líder en el mercado que soporta todos los tipos de diagramas UML. Tiene una interfaz de usuario intuitiva y permite la colaboración en equipo, así como la integración con otros sistemas.

**Enterprise Architect**: es una herramienta de modelado de software potente y completa que soporta todos los tipos de diagramas UML. Ofrece una amplia gama de características y funcionalidades para la modelización de sistemas complejos.

**Lucidchart:** es una herramienta de modelado de software en línea que ofrece una interfaz de usuario fácil de usar y soporta una amplia variedad de tipos de diagramas UML, incluyendo diagramas de flujo de datos y diagramas de red.

**Draw.io**: es una herramienta de modelado de software en línea gratuita que soporta todos los tipos de diagramas UML y ofrece una amplia gama de opciones de personalización.

**Gliffy:** es una herramienta de modelado de software en línea que ofrece una interfaz de usuario fácil de usar y soporta una amplia variedad de tipos de diagramas UML.

**Astah**: es una herramienta de modelado de software que soporta todos los tipos de diagramas UML y ofrece una amplia gama de características y funcionalidades para la modelización de sistemas complejos.