**UNIVERSIDAD AUTÓNOMA**

**GABRIEL RENE MORENO**

**FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



**Asistente Virtual Avanzado para reducir el tiempo de espera de los clientes de la empresa VIVA**

**Estudiantes:** Bustos Belzu Pablo Ramiro - 212072811

Vaca García Pierluigi Abdías - 210025190

Santa Cruz – Bolivia

Contenido

[CAPÍTULO 1 5](#_Toc153897444)

[1.- Planteamiento del problema 5](#_Toc153897445)

[1.1.- Introducción 5](#_Toc153897446)

[1.1.1.- Situación problemática 6](#_Toc153897447)

[1.1.2.- Situación deseada 6](#_Toc153897448)

[1.2.- Objetivos 6](#_Toc153897449)

[1.2.1.- Objetivo general 6](#_Toc153897450)

[1.2.2.- Objetivos específicos 6](#_Toc153897451)

[*1.3.-* Metodología 7](#_Toc153897452)

[1.3.1.- Fase Inicio 7](#_Toc153897453)

[1.3.2.- Fase de Elaboración 7](#_Toc153897454)

[1.3.3.- Fase de Construcción 8](#_Toc153897455)

[1.3.4.- Fase de Pruebas 8](#_Toc153897456)

[1.4.- Alcance 8](#_Toc153897457)

[CAPÍTULO 2 9](#_Toc153897458)

[2.- Procesamiento del lenguaje natural. 9](#_Toc153897459)

[2.1.- Analítica de Textos en la Detección de Sentimientos 15](#_Toc153897460)

[2.2.- Redes neuronales recurrentes en el aprendizaje automático 17](#_Toc153897461)

[2.3.- Proceso de una orden de trabajo. 17](#_Toc153897462)

[CAPÍTULO 3 20](#_Toc153897463)

[3.- Telefónica Nuevatel Viva 20](#_Toc153897464)

[3.1.1.- Antecedentes 20](#_Toc153897465)

[3.1.2.- Misión 20](#_Toc153897466)

[3.1.3.- Visión 20](#_Toc153897467)

[3.1.4.- Estructura Organizacional 21](#_Toc153897468)

[3.1.5.- Infraestructura 23](#_Toc153897469)

[3.1.6.- Procedimiento. 24](#_Toc153897470)

[CAPÍTULO 4 25](#_Toc153897471)

[4.- Especificación de requerimientos según el estándar IEEE 830 25](#_Toc153897472)

[4.1.- Introducción 25](#_Toc153897473)

[4.1.1.- Propósito 25](#_Toc153897474)

[4.1.2.- Ámbito del sistema 25](#_Toc153897475)

[4.1.3.- Definiciones, acrónimos y abreviaturas 26](#_Toc153897476)

[4.1.4.- Referencia 26](#_Toc153897477)

[4.2.- Descripción general 26](#_Toc153897478)

[4.2.1.- Perspectiva del proyecto 26](#_Toc153897479)

[4.2.2.- Funciones del proyecto 26](#_Toc153897480)

[4.2.3.- Características de los usuarios 27](#_Toc153897481)

[4.3.- Restricciones 28](#_Toc153897482)

[4.3.1.- Políticas reguladoras 28](#_Toc153897483)

[4.3.2.- Funciones de control 28](#_Toc153897484)

[4.3.3.- Requisitos del lenguaje 28](#_Toc153897485)

[4.3.4.- Requisitos de fiabilidad 28](#_Toc153897486)

[4.3.5.- Credibilidad de la aplicación 28](#_Toc153897487)

[4.3.6.- Consideraciones de seguridad 29](#_Toc153897488)

[4.3.7.- Suposiciones y dependencias 29](#_Toc153897489)

[4.4.- Requisitos específicos 29](#_Toc153897490)

[4.4.1.- Requisitos funcionales 29](#_Toc153897491)

[4.4.2.- Requisitos no funcionales 30](#_Toc153897492)

[CAPÍTULO 5 32](#_Toc153897493)

[5.- Captura de requisitos 32](#_Toc153897494)

[5.1.- Priorización de los casos de uso 32](#_Toc153897495)

[5.2.- Detalle de casos de uso 33](#_Toc153897496)

[CAPÍTULO 6 47](#_Toc153897497)

[6.- Análisis 47](#_Toc153897498)

[6.1.- Modelo de dominio inicial 47](#_Toc153897499)

[6.2.- Análisis de arquitectura 48](#_Toc153897500)

[6.2.1.- Identificación de paquetes 48](#_Toc153897501)

[6.2.2.- Análisis de paquete 49](#_Toc153897502)

[6.3.- Análisis de casos de uso 50](#_Toc153897503)

[6.4.- Análisis de clases 54](#_Toc153897504)

[CAPÍTULO 7 60](#_Toc153897505)

[7.- DISEÑO 60](#_Toc153897506)

[7.1.- Diseño de la arquitectura 60](#_Toc153897507)

[7.2.- Diseño de paquetes 61](#_Toc153897508)

[7.3.- Diseño de la arquitectura física 62](#_Toc153897509)

[7.4.- DISEÑO DE CASOS DE USO 63](#_Toc153897510)

[7.5.- Diagrama de la base de datos 66](#_Toc153897511)

[7.5.1.- Diseño conceptual 66](#_Toc153897512)

[7.5.2.- Diseño lógico 67](#_Toc153897513)

[7.5.3.- Diseño físico 68](#_Toc153897514)

[CAPÍTULO 8 86](#_Toc153897515)

[8.- IMPLEMENTACIÓN DE ARQUITECTURA 86](#_Toc153897516)

[8.1.- Diagrama de componente 86](#_Toc153897517)

[8.2.- IMPLEMENTACIÓN DE ARQUITECTURA DE SUBSISTEMA 87](#_Toc153897518)

[8.3.- HERRAMIENTAS DE DESARROLLO 91](#_Toc153897519)

[8.3.1.- SISTEMA OPERATIVO 91](#_Toc153897520)

[8.3.2.- LENGUAJES DE PROGRAMACIÓN 92](#_Toc153897521)

[8.3.3.- BASE DE DATOS 95](#_Toc153897522)

[8.3.4.- HERRAMIENTAS CASE 96](#_Toc153897523)

[CAPÍTULO 9 98](#_Toc153897524)

[9.- Flujo de trabajo: Prueba 98](#_Toc153897525)

[9.1.- Plan de prueba 98](#_Toc153897526)

[9.1.1.- Caja Negra 98](#_Toc153897527)

[9.1.2.- Caja Blanca 102](#_Toc153897528)

[9.1.3.- TDD (Desarrollo dirigido por pruebas) 106](#_Toc153897529)

[CAPITULO 10 108](#_Toc153897530)

[10.- COSTOS 108](#_Toc153897531)

[10.1.- POSIBLES COSTOS 108](#_Toc153897532)

[10.1.1.- Hardware 108](#_Toc153897533)

[10.1.2.- Software 109](#_Toc153897534)

[10.1.3.- Recursos humanos 109](#_Toc153897535)

[Bibliografía 110](#_Toc153897536)

# CAPÍTULO 1

1. Planteamiento del problema

## Introducción

Actualmente, se observan que muchas empresas tienen un soporte de atención al cliente, ya sean las empresas telefónicas, los bancos, universidades, toda empresa que preste un servicio. Con lo cual se debe contar las 24 horas del día para atender a los clientes, este es un servicio que se viene brindando desde que apareció el teléfono.

En ocasiones este servicio está tan saturado que no podemos ser atendidos por un operador y se debe esperar hasta que se nos pueda atender, sin embargo, del mismo modo hay momentos dados en los que nadie está haciendo consultas y se tiene a los operadores sin estar desempeñando sus funciones.

Esto con el fin de poder subsanar estas dos problemáticas en esta época han aparecido herramientas que nos permiten crear un asistente que pueda contestar todas las consultas de nuestros clientes sin la necesidad de esperar, o de que el asistente no esté cumpliendo sus funciones.

De esta manera es que se desarrollara un software capaz de brindar solución a esta problemática, mejorando así la calidad de servicio que ofrece.

### Situación problemática

La empresa Viva enfrenta un desafío en su servicio de atención al cliente y soporte técnico, disponible las 24 horas. Esto implica costos significativos de personal y, en periodos de alta demanda, el equipo actual no puede atender completamente las necesidades de los clientes, lo que repercute negativamente en la percepción de la empresa. Además, aunque Viva recopila datos diariamente sobre las experiencias de los clientes con el servicio técnico, aún no se aprovechan de manera óptima para la mejora de los servicios.

### Situación deseada

La empresa de telefonía Viva tiene como objetivo principal la implementación de un servicio de atención al cliente disponible las 24 horas, con el propósito de asegurar una atención constante y efectiva a nuestros usuarios en cualquier momento del día. Además, buscamos aprovechar soluciones informáticas avanzadas para llevar a cabo un análisis exhaustivo de las solicitudes recibidas por parte de nuestros usuarios. Este análisis nos permitirá obtener una comprensión más profunda de sus experiencias y necesidades, lo que a su vez nos ayudará a mejorar nuestros servicios y ofrecer respuestas más eficientes y personalizadas a sus requerimientos.

## Objetivos

### Objetivo general

Desarrollar Asistente Virtual Avanzado para reducir el tiempo de espera de los clientes de la empresa VIVA.

### Objetivos específicos

* Realizar reuniones con los encargados de brindar el servicio de soporte técnico.
* Definir los requisitos funcionales y no funcionales a través de los requerimientos solicitados.
* Analizar los requisitos para entender la situación problemática.
* Identificar los casos de uso
* Diseñar la arquitectura del software
* Identificar los patrones de arquitectura
* Diseñar la base de datos
* Implementar los casos de uso
* Realizar pruebas con el objetivo de encontrar errores

## Metodología

La metodología que adoptamos para el proyecto es el Proceso Unificado de Desarrollo de Software (PUDS), Rumbaugh, Jacobson, y Booch, (2001). Además, se utilizará al Lenguaje Unificado de Modelado (UML) como lenguaje de representación de los modelos resultantes en cada actividad del PUDS, Jacobson et al. (2000).

### Fase Inicio

Captura de Requisitos

* Requisitos funcionales y no funcionales
* Lista de actores y casos de uso
* Detalle de casos de uso
* Prototipo de la interfaz de los casos de uso

Resultado

* Modelo de los casos de uso

### Fase de Elaboración

Actividades

* Diseño de arquitectura
* Diseño de casos de uso
* Diseño de datos
* Resultado
* Descripción del diseño de software

### Fase de Construcción

Implementación

Traducción de los modelos de diseño, a lenguaje de programación.

Además, utilizaremos el UML (Lenguaje Unificado de Modelado) como lenguaje de representación visual.

### Fase de Pruebas

En esta fase final, el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema.

## Alcance

* Módulo de soporte técnico. - En este módulo se gestionará las solitudes de asistencia de soporte técnico que requieran los usuarios.
* Modulo registro de usuarios. - En este módulo se registrará a los diferentes usuarios, técnicos, gerentes que tendrán acceso al sistema.
* Módulo de servicios. - En este módulo se registrarán los servicios de telefonía e internet que ofrece la empresa telefónica.
* Módulo de análisis de la información. - En este módulo se analizará las conversaciones que tengan con el asistente para determinar el grado de satisfacción del usuario.
* Módulo de asistente virtual. - En este módulo se entrenará un modelo que atienda las solicitudes de los usuarios.
* Módulo de contratos. - En este módulo se registrarán los contratos que adquieran los nuevos usuarios por nuestros servicios.

# CAPÍTULO 2

1. Procesamiento del lenguaje natural.

(Bagnato,J.I) En NLP no es suficiente con comprender solo las palabras sin que se debe comprender el conjunto de las palabras que llevan a una oración de la misma forma a las líneas que conforman un párrafo.

Usos del Procesamiento del Lenguaje Natural:

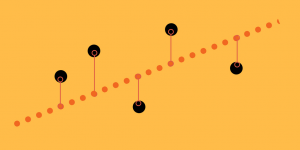
* Resumen de textos: Consiste en realizar un resumen de un texto/párrafo del cual obtendrá la idea principal e ignorará el texto no relevante
* ChatBots: Tener una conversación fluida con el usuario y responder a preguntas.
* Generación automática de keywords y generación de texto: Lo que nos quiere decir que a partir de palabras clave se generara un texto en el estilo que se define de antemano.
* Reconocimiento de entidades: Obtendrá de un texto/párrafo a los involucrados en este caso las Personas, Marcas, entidades comerciales o gubernamentales, Países, Ciudades...
* Análisis de sentimientos: Comprender si un texto/párrafo es neutro, positivo o negativo.
* Traducción automática de idiomas
* Clasificación automática de texto: Consiste en clasificar los textos dependiendo de los temas recurrentes y crear una categoría.

Técnicas usadas en el Procesamiento del Lenguaje Natural

* Tokenizar: Separar las palabras del texto en entidades llamadas tokens.
* Tagging Part of Speech (PoS): Clasificar la oraciones en verbo, sustantivo, adjetivo, preposición, etc..
* Shallow Parsing / Chunks: Para entender la gramática de las oraciones, se hace un parseo de sus tokens y a partir de un PoS se arma un árbol de su estructura.
* Significado de las palabras: lexical semantics y word sense disambiguation.
* Pragmatic Analysis: Detectar como se dicen las cosas: Ironia, sarcasmo, intencionalidad, etc.
* Bag of words: es una manera de representar el vocabulario que utilizaremos en nuestro modelo y consiste en crear una matriz en la que cada columna es un token y se contabilizará la cantidad de veces que aparece ese token en cada oración (representadas en cada fila)
* Word2vec: Es una técnica que aprende de leer enormes cantidades de textos y memorizar qué palabras parecen ser similares en diversos contextos. Luego de entrenar suficientes datos, se generan vectores de 300 dimensiones para cada palabra conformando un nuevo vocabulario en donde las palabras “similares” se ubican cercanas unas de otras. Utilizando vectores pre-entrenados, logramos tener muchísima riqueza de información para comprender el significado semántico de los textos.

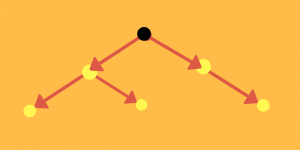
(Bagnato, J.I) A lo largo del Aprendizaje Automático nos encontraremos con diversos algoritmos para la resolución de problemas y es nuestro objetivo saber que algoritmo se debe emplear para resolver un problema en particular.

Algoritmos de regresión: Se modela la relación entre distintas variables (features) y nos devuelve un valor numérico o boolean. Por ejemplo: Si deseamos predecir cuándo va llover nuestras variables serian: temperatura, viento, humedad de la cual tendremos como resultado una afirmación o negación. Otro ejemplo: Consiste en predecir el precio de un inmueble como variables tendríamos: metros cuadrados, cuartos, barrio, etc., del cual se obtendrá como resultado un valor numérico.



Nota: Algoritmos de Regresión, Logístico o Lineal. Nos ayudan a clasificar o predecir valores. Se intentará compensar la mejor respuesta a partir del menor error.

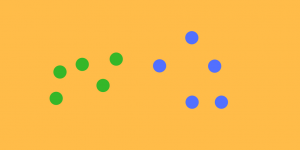
Algoritmos de Árbol de Decisión: Modelan la toma de Decisión basado en los valores actuales (reales) de los atributos que tienen nuestros datos. Se utilizan sobre todo para clasificación de información, bifurcando y modelando los posibles caminos tomados y su probabilidad de ocurrencia para mejorar su precisión. Una vez armados, los árboles de decisión ejecutan muy rápido para obtener resultados.

Nota: Árbol de Decisión. Buscará el mejor árbol, balanceando la posibilidad de ocurrencia y su importancia en cada rama y hojas para clasificar un resultado

Algoritmos Bayesianos: Son algoritmos que utilizan explícitamente el Teorema de Bayes de probabilidad para problemas de Clasificación y Regresión.

Nota: Algoritmos Bayesianos aprovechan los teoremas de probabilidad en Aprendizaje Automático

Algoritmos de Clustering: Se utilizan para agrupar datos existentes de los que desconocemos sus características en común o queremos descubrirlas.  
Estos métodos intentan crear “puntos centrales” y jerarquías para diferenciar grupos y descubrir características comunes por cercanía.



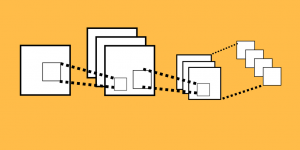
Nota: Algoritmos de Agrupación son de Aprendizaje no supervisado y encontrarán relaciones entre los datos que seguramente no contemplamos a simple vista.

Algoritmos de Redes Neuronales: Son algoritmos y estructuras inspirados en las funciones biológicas de las redes neuronales.  
Se suelen utilizar para problemas de Clasificación y Regresión, pero realmente tienen un gran potencial para resolver multitud de problemáticas. Son muy buenas para detectar patrones. Las Redes Neuronales Artificiales requieren mucha capacidad de procesamiento y memoria y estuvieron muy limitadas por la tecnología del pasado hasta estos últimos años en los que resurgieron con mucha fuerza dando lugar al Aprendizaje Profundo.



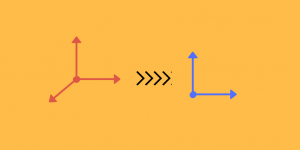
Nota: Las Redes Neuronales imitan el comportamiento de activación biológico y la interconexión entre neuronas para buscar soluciones no lineales a problemas complejos

Algoritmos de Aprendizaje Profundo: Son la [evolución de las Redes Neuronales Artificiales](https://www.aprendemachinelearning.com/breve-historia-de-las-redes-neuronales-artificiales/) que aprovechan el abaratamiento de la tecnología y la mayor capacidad de ejecución, memoria y disco para explotar gran cantidad de datos en enormes redes neuronales interconectarlas en diversas capas que pueden ejecutar en paralelo para realizar cálculos.



Nota: Las redes convolucionales hacen que una red neuronal de aprendizaje profundo tenga la capacidad de reconocer animales, humanos y objetos dentro de imágenes

Algoritmos de Reducción de Dimensión: Buscan explotar la estructura existente de manera no supervisada para simplificar los datos y comprimirlos.  
Son útiles para visualizar datos o para simplificar el conjunto de variables que luego pueda usar un algoritmo supervisado.

Nota: Reducción de dimensión nos permite graficar o simplificar modelos muy complejos que en su condición inicial contenían demasiadas características.

(Camps, G) La compresión del Lenguaje Natural resulta muy útil para categorizar y archivar documentos, la generación de texto y puede aportar un valor añadido a la construcción de modelos de inteligencia artificial.

**Análisis de sentimientos**

El análisis de sentimientos se encarga de identificar, extraer y cuantificar información subjetiva de un texto. Es comúnmente utilizado para determinar la polaridad y la subjetividad de opiniones de clientes o usuarios de las redes sociales.

**Segmentación de texto**

La segmentación de texto o tokenización consiste en separar un texto en segmentos más cortos como palabras u oraciones. Si bien esta tarea es relativamente fácil para idiomas como el castellano o el inglés, algunas lenguas como el chino no necesariamente tienen una separación clara entre palabras.

**Lematización y Stemming**

La lematización consiste en hallar el lema de una palabra, es decir, encontrar la forma de la palabra que encontraríamos en un diccionario. Por ello, un nombre en plural pasará a ser singular, un adjetivo será masculino y las conjugaciones verbales se representarán con el infinitivo. Este proceso implica saber la categoría gramatical (en inglés, part-of-speech, POS) de la palabra en el contexto de la oración. Por otro lado, stemming es un proceso que consiste en reducir una palabra a su raíz.

**Reconocimiento de entidades nombradas**

El Reconocimiento de Entidades Nombradas (NER) se encarga de encontrar entidades que se identifican con un nombre propio y a clasificarlas según su tipo. En muchas ocasiones, estas entidades tienen la primera letra de su nombre en mayúscula porque nos referimos a un objeto concreto de la vida real como una persona, una organización o un país.

**Bolsa de palabras y TF-IDF**

Una bolsa de palabras (bag-of-words, BOW) consiste en la representación de un documento mediante el conjunto de las palabras que contiene, sin importar el orden. Dicho de otra manera, cada palabra diferente del documento sería un elemento de la bolsa, donde también se almacena el número de veces que esta palabra se repite a lo largo del documento.

Por otro lado, la frecuencia de término – frecuencia inversa de documento (TF-IDF) es una medida que refleja la importancia de una palabra dentro de un documento. Consiste en multiplicar dos términos: TF e IDF. Para el primero, se puede usar el recuento que se obtiene de la bolsa de palabras. El término IDF es una medida de cuánta información aporta una palabra, teniendo en cuenta el número de documentos analizados y el número de documentos que contienen dicha palabra:



## Analítica de Textos en la Detección de Sentimientos

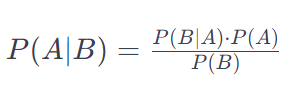
Según (Rosenbrock, 2021), El análisis del sentimiento o la minería de opinión es el estudio computacional de opiniones, sentimientos y emociones expresadas a través de un texto.  En general, las opiniones pueden centrarse en un producto, un servicio, un individuo, una organización, un evento o un tema. Utilizamos el término objeto para denotar la entidad de destino que se ha comentado. Un objeto puede además tener un conjunto de componentes (o partes) y un conjunto de atributos o propiedades).

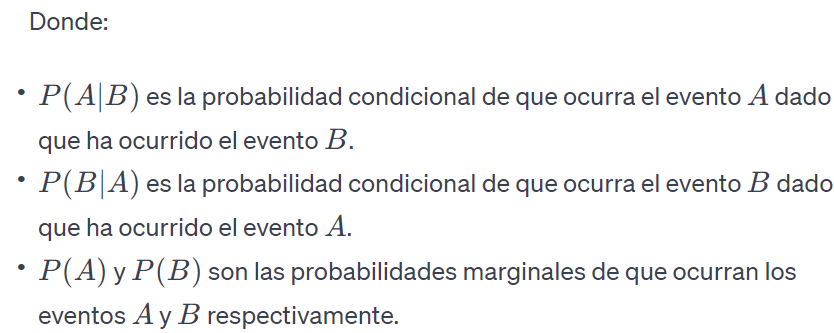
Según (Gonzales, 2021), la analítica de sentimientos es una de las vertientes del procesamiento de lenguaje natural (PLN), la cual permite la obtención de información subjetiva, es decir que esta derivada directamente de los sentimientos de las personas; como bien sabemos, la aplicación del PLN está basada principalmente en el análisis de texto no estructurado, por lo que esto va de la mano de la analítica de texto, con la que es factible contar, agrupar y categorizar palabras para generar una estructura y así dar significado a grandes volúmenes de información.

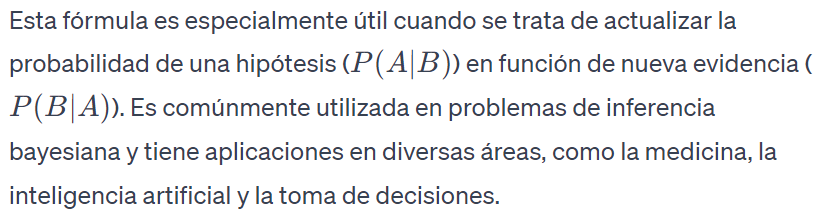
La obtención de esta información puede provenir de distintos tipos de fuente, desde comentarios escritos en alguna red social, hasta de conversaciones telefónicas del servicio al cliente, que es nuestro caso, en específico de llamadas a una empresa del giro automotriz; su aplicación nos permitirá la identificación de frases y entidades clave, como organizaciones, lugares o personas, para comprender el contexto general de la información que se está analizando.

**Teorema de Bayes**

El algoritmo de Bayes predice una probabilidad condicional de que una instancia pertenezca a una clase, a partir de las instancias en el conjunto de entrenamiento que señalen que tiene una alta probabilidad de pertenecer a cierta clase. (Gonzales, 2021)





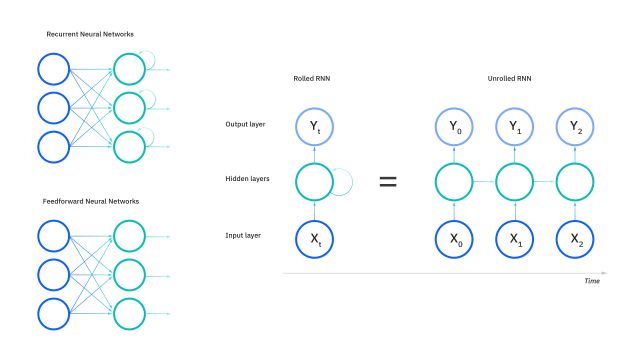


## Redes neuronales recurrentes en el aprendizaje automático

Según (IBM) Una red neuronal recurrente (RNN) **es un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series de tiempo**. Estos algoritmos de aprendizaje profundo se utilizan comúnmente para problemas ordinales o temporales, como la traducción de idiomas, el reconocimiento de voz y subtítulos de imágenes. Cabe mencionar que están incorporados en aplicaciones populares como **Siri, búsqueda por voz y Google Translate**.

Si bien las redes neuronales profundas tradicionales asumen que los datos de entrada y los resultados son independientes entre sí, **los resultados de las redes neuronales recurrentes dependen de los elementos anteriores dentro de la secuencia**. Aunque los eventos futuros también serían útiles para determinar los resultados de una secuencia dada, las redes neuronales recurrentes unidireccionales no pueden tener en cuenta estos eventos en sus predicciones.

(eldeber, 2023)



Nota: El grafico pertenece a los tipos de redes neuronales recurrentes. (IBM)

## Proceso de una orden de trabajo.

1. Recepción de solicitud de trabajo (secretaria/o). **(**Figura 2)

2. Registro en el sistema la orden de trabajo (secretaria/o).

3. Notifica las órdenes de trabajo registradas (secretaria/o) (administrador).

4. Verifica el tipo de fallo (administrador).

5. Asigna y distribuye al personal (técnico) para los diferentes tipos de fallos (administrador).

6. Envía las órdenes de trabajo con el (técnico) asignado para cada soporte a realizar (administrador) (secretaria/o).

7. Notifica la asignación de tareas a realizarse (secretaria/o) (técnico).

8. Recepta las órdenes de trabajo a realizar (técnico).

9. Ejecuta la orden de trabajo a realizarse (técnico).

10. Realiza el registro manual y describe el tipo de fallo que existió, en la cual se detallan: datos generales, actividades realizadas y materiales utilizados (técnico).

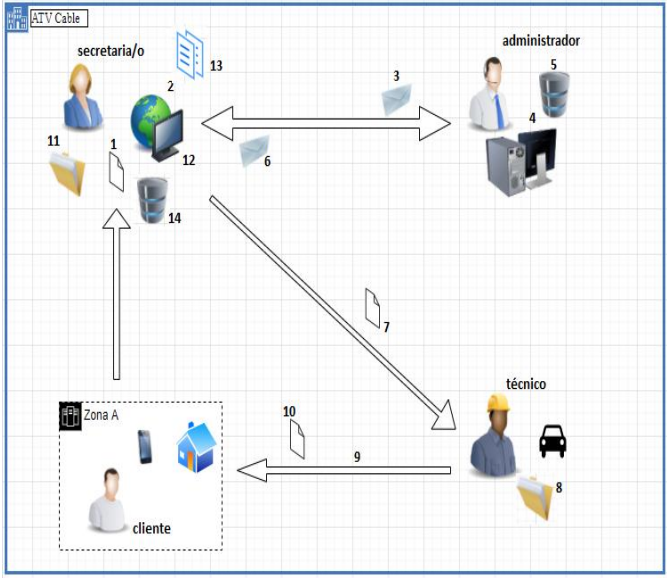
11. Recepta el cumplimiento de las órdenes de trabajo realizadas (secretaria/o).

12. Verifica el cumplimiento de las órdenes de trabajo (secretaria/o).

13. Realiza el registro de la orden de trabajo y a su vez un reporte de cada una de ellas (secretaria/o).

14. Archiva las órdenes de trabajo (secretaria/o).

**Figura 2**



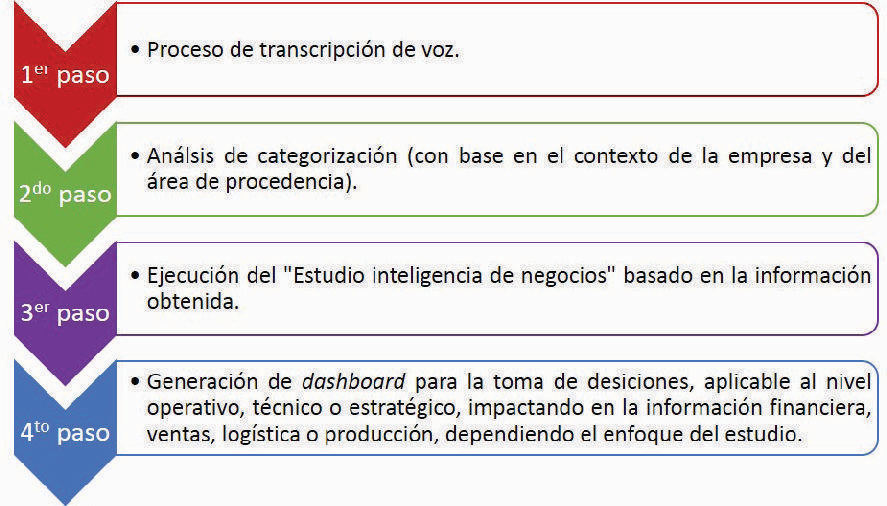
Nota: El grafico representa Proceso para gestionar una orden de trabajo.

Fuente (AtvCable)

Esta aplicación del PLN, sumada a otros avances y posibilidades tecnológicas, nos permite la suplantación de procesos tradicionales en los cuales era necesaria una gran inversión en recursos humanos para llevarlos a cabo.

En la Figura 3 podemos observar la estructura de un proceso mecánico ejecutado directamente por recursos humanos:

**Figura 3**



Nota: Proceso de análisis tradicional de información de conversaciones telefónicas en una empresa mexicana dedicada al análisis de datos. (Gonzales, 2021)

# CAPÍTULO 3

1. Telefónica Nuevatel Viva

### Antecedentes

Nuevatel PCS fue fundada en 1999 a partir de las inversiones de [Western Wireless International](https://en.wikipedia.org/wiki/Western_Wireless) con un 72% y [COMTECO](https://es.wikipedia.org/wiki/COMTECO) con un 28% y comenzó operaciones un año después. El 9 de enero de 2005, Western Wireless celebró un acuerdo de fusión con el proveedor de telecomunicaciones [Alltel](https://es.wikipedia.org/wiki/Alltel" \t "Alltel) con sede en [Little Rock, Arkansas](https://es.wikipedia.org/wiki/Little_Rock,_Arkansas), en el que Alltel acordó pagar US$6000 millones en [acciones](https://es.wikipedia.org/wiki/Acciones) y en efectivo a los accionistas de Western Wireless. La Empresa de Telecomunicaciones Nuevatel PCS de Bolivia S.A., más conocida como VIVA, es una operadora de telecomunicaciones de [Bolivia](https://es.wikipedia.org/wiki/Bolivia). Fue fundada en [1999](https://es.wikipedia.org/wiki/1999). En la actualidad se encuentra entre las mayores empresas del país. Su especialidad se encuentra en la cobertura de telefonía celular en Bolivia.

### Misión

En VIVA, nuestra misión es brindar experiencias excepcionales y soluciones innovadoras a nuestros clientes en Bolivia. Nos comprometemos a involucrar a nuestra comunidad para moldear nuestros servicios y satisfacer mejor sus necesidades. Creemos que la autenticidad y la interacción humana son clave para llevar la experiencia del cliente y nuestros productos a nuevos niveles.

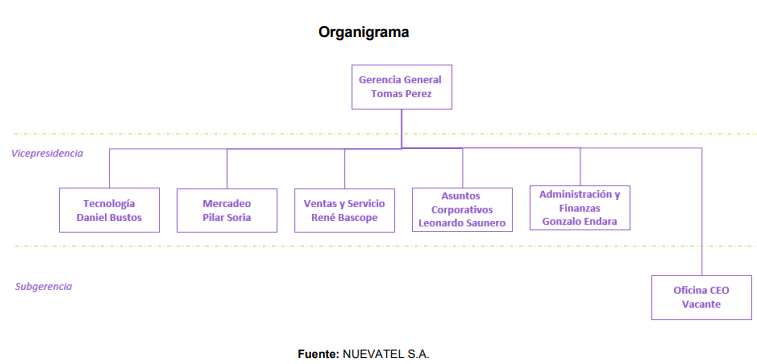
### Visión

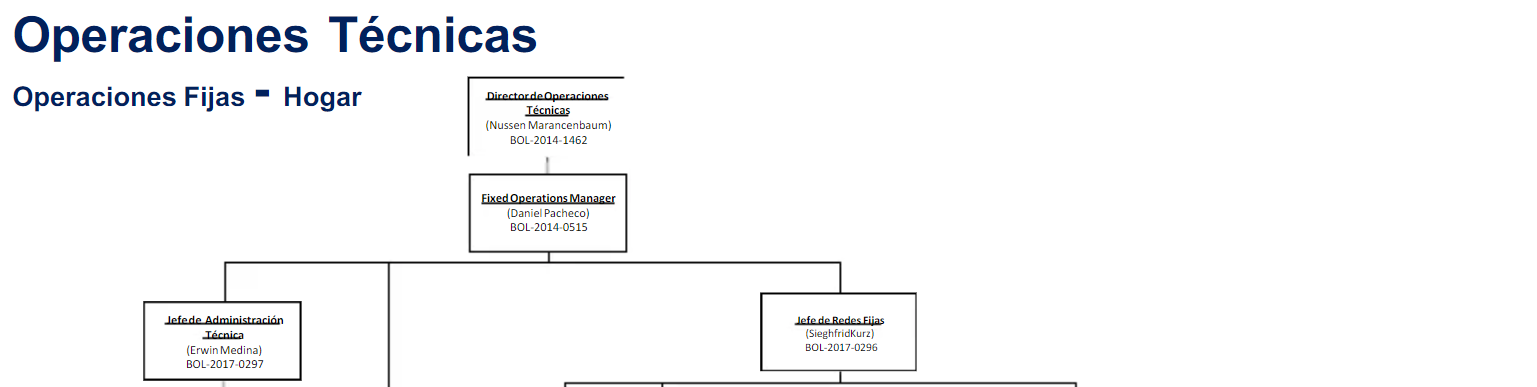
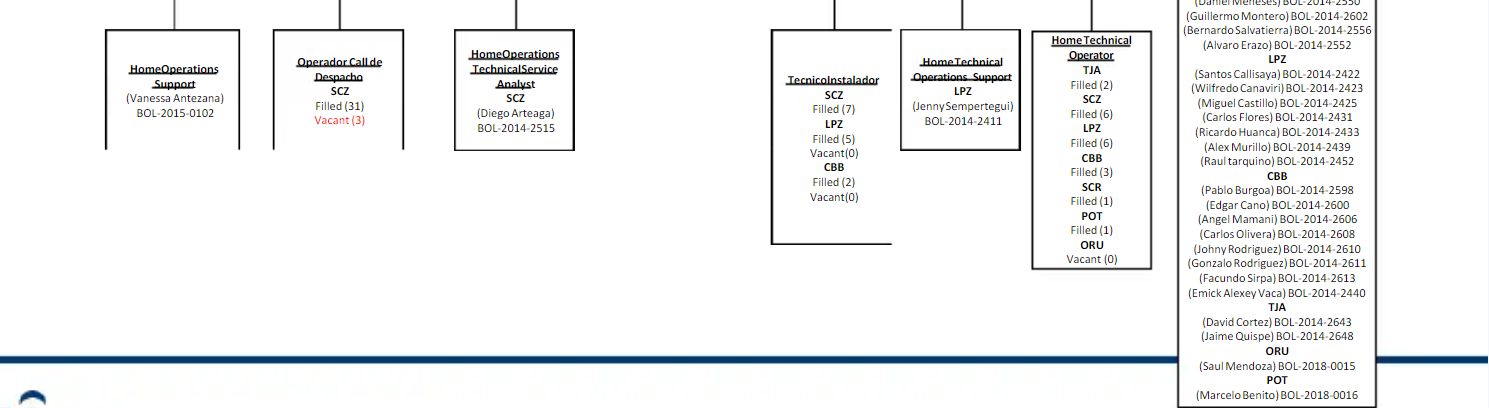
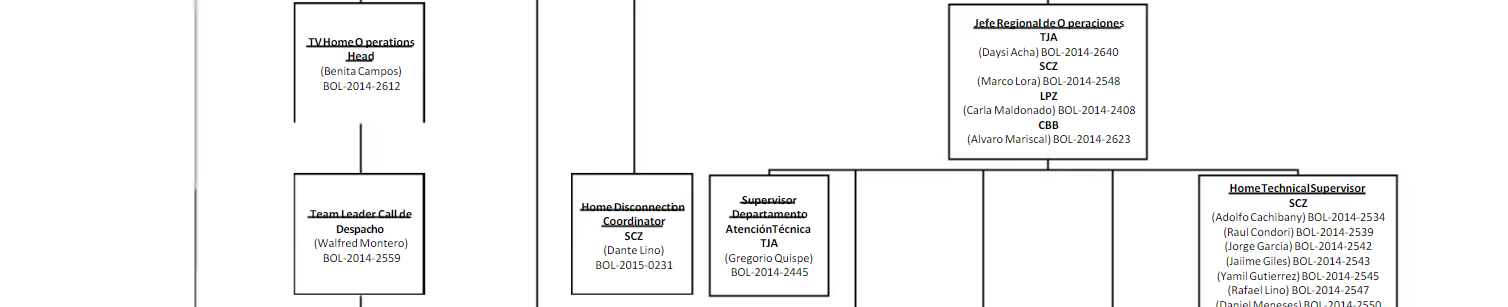
En VIVA, nuestra visión es convertirnos en el principal proveedor de soluciones de telecomunicaciones en Bolivia, ofreciendo servicios innovadores que conecten a las personas y transformen la forma en que viven, trabajan y disfrutan la vida.

### Estructura Organizacional

**Figura 3**

*Estructura organizacional de empresa Viva, área de operaciones técnicas*



Fuente (Organigrama Viva, 2019)

### Infraestructura

La empresa Viva cuenta con diferentes sucursales:

* TIENDA CENTRAL - SANTA CRUZ

Av. San Martin, entre calle Los Claveles y calle Begonias s/n, Zona Equipetrol

* Tienda VIVA al paso

JAVIER GUTIERREZ HILAQUITA

Zona 1rer Anillo, Av. Cañoto entre Junín y casi esquina Ayacucho # 383.

* Tienda VIVA al paso

MARGOTH MONTERO BALDERAS

Calle Bolívar, entre 24 de septiembre y Beni. Shopping Bolívar Local 4, Zona Central media cuadra antes de llegar a la plaza principal.

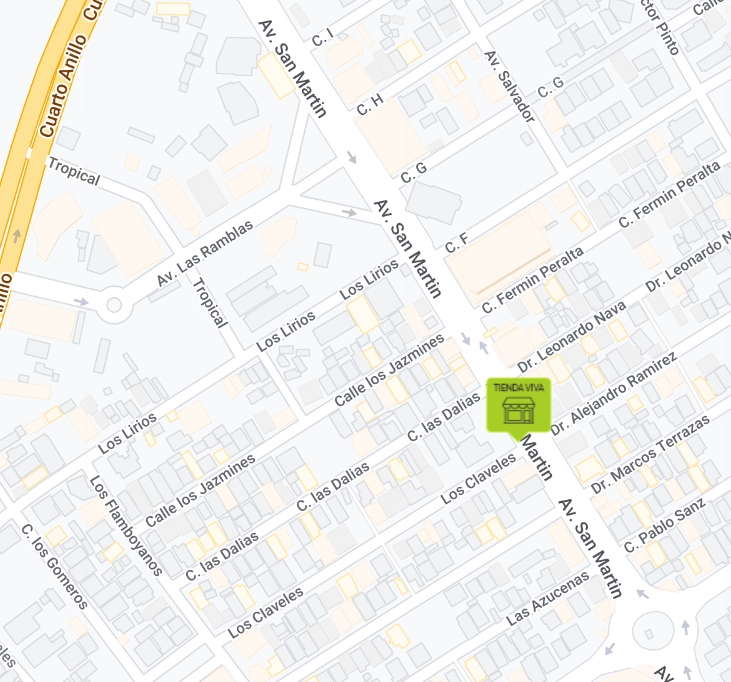
* Tienda VIVA al paso

CARMEN ALEJANDRA ARCE ARCE

Av. Grigota, Comercial Grigota, Zona la Ramada 1rer anillo Pasillo Principal, Local #26

**Figura 4**

***Ubicación Geográfica tienda central en Santa Cruz, Barrio Equipetrol.***

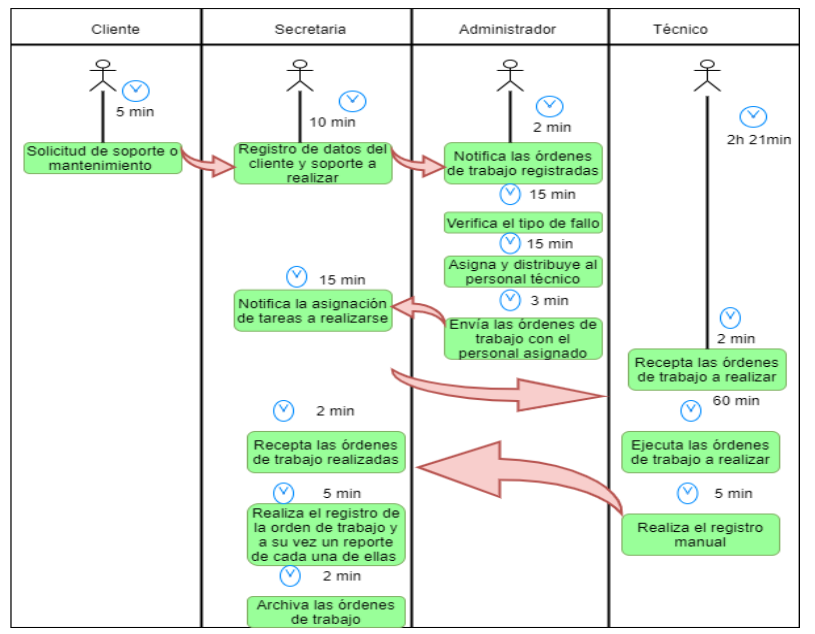


Fuente (Google Maps)

### Procedimiento.

**Figura 5**

Solicitud de modelo y realización de una orden de trabajo



Nota: Proceso para registro de solicitudes de soporte y asignación del trabajo para la visita del técnico.

Fuente (Elaboración propia)

# CAPÍTULO 4

1. Especificación de requerimientos según el estándar IEEE 830

## Introducción

En la presente sección del documento se procederá a detallar los requisitos del software a través de distintos mecanismos, como ser entrevistas, encuestas en otro. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Practica Recomendada para Especificaciones de Requisitos de Software ANSI/IEEE 830,1998.

### Propósito

El presente documento tiene como objeto definir todas las funcionalidades y restricciones para el desarrollo de un software de Call Center para la gestión de órdenes de servicio técnico para una telefónica. Este documento servirá como canal de comunicación entre los miembros de la empresa para una mejor comprensión del software. El documento va dirigido a los usuarios finales de la empresa.

### Ámbito del sistema

El presente proyecto de asistente Call Center para gestionar las ordenes de servicio técnico, brindara una mejor experiencia al usuario reduciendo los tiempos de espera y mejorando el proceso de asignación de trabajos a los técnicos.

### Definiciones, acrónimos y abreviaturas

**Tabla 1: Acrónimos y abreviaturas**

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Cliente | Persona que usará el sistema para gestionar procesos |
| NLP | Procesamiento de lenguaje natural |
| ERS | Especificación de Requisitos Software |
| STT | Procesamiento de voz a texto |
| **TTS** | **Procesamiento de texto a voz** |

### Referencia

Méndez, G (22 de octubre 2008) Especificación de Requisitos según el Estándar de la IEEE 830.

## Descripción general

El software se ejecutará internamente por parte del personal de la telefónica administrando el flujo de información recibida a través de la web y de la app.

### Perspectiva del proyecto

El software de control de solicitudes dependerá del modelo que se entrenó con los datos recolectados. Al que se accederá desde una API para realizar sus funciones principales como la realización de solicitudes de soporte técnico.

### Funciones del proyecto

Las funciones de la aplicación son primero permitir a los agentes el ingresar por medio de una cuenta, después acceder a los datos de sus clientes y las solicitudes de soporte.

**F1. Gestionar usuario:** Se registrará a toda persona que ingrese al sistema.

**F2. Gestionar interacción:** Tener centralizada la información de las solicitudes por si un cliente está interesado en pedir soporte técnico o solicitar información.

**F3. Gestionar reporte:** El cliente podrá visualizar un reporte de su solicitud.

**F4. Gestionar ordenes de trabajo:** Son las visitas técnicas asignadas a cada técnico.

**F5. Gestionar tipo de servicio técnico:** Son todos los diferentes servicios que brinda la empresa al cliente para mejorar el servicio.

**F6. Publicación de la información:** Se requiere un dashboard para visualizar la información en tiempo real.

**F7. Analizar llamada:** Se desea analizar el nivel de satisfacción del cliente.

**F8. Reducir el tiempo de respuesta:** Se requiere una tecnología para reducir el tiempo de esperar al cliente.

**F9. Administrar las solicitudes, consultas y peticiones:** Automatizar todas las consultas que requiera el cliente.

**F10. Gestionar técnicos:** Se registrará a los empleados que realizar el trabajo de brindar soporte técnico a los clientes de la telefonía.

**F11. Gestionar planes de servicio:** Se registrará los planes de internet, tv cable y planes post pagos que se ofrece a los clientes.

**F12. Gestionar contratos:** Se administrará todos los servicios contratados por los clientes.

**F13. Gestionar Facturación:** Se gestionará los pagos que realizan los clientes.

### Características de los usuarios

|  |  |  |
| --- | --- | --- |
| USUARIO | CARACTERÍSTICA | DESCRIPCIÓN |
| Técnico | Conocimientos en equipos de telecomunicación | Es el encargado de las instalaciones domiciliarias |
| Jefe de Administración Técnica | Conocimientos en administración de personal. | Es el encargado de registrar a los técnicos. |
| Gerente de ventas y servicios | Conocimientos básicos de computación. | Es el encargado de ver los reportes, tomar decisiones y registrar los planes que ofrecen. |
| Cliente | No necesitan conocimientos técnicos | Tendrá acceso a su contrato realizado y podrá informar cualquier inconveniente a la inmobiliaria |

## Restricciones

Las restricciones son las siguientes:

### Políticas reguladoras

La aplicación se desarrollará mediante el framewok Laravel, MySQL para la gestión de la base de datos y Speech to text para la captura de la información de voz a texto que tienen licencia gratuita, por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

### Funciones de control

El sistema debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder la información que le corresponde de acuerdo a su rol. Debe tener controles adecuados para la validación de datos, de igual manera la programación de las actividades específicas para cada seminario.

### Requisitos del lenguaje

Todo el material que se realice para el usuario y la aplicación debe de estar en lenguaje español.

### Requisitos de fiabilidad

La información correspondiente a la programación de los seminarios como la fecha, la hora, el sitio, cantidad de inscritos, ponente y los recursos deben estar ajustados a la realidad para evitar desorganización.

### Credibilidad de la aplicación

Para garantizar una buena credibilidad el sistema deberá ser sometido a una serie de pruebas para establecer que se encuentra acorde a los requerimientos que se plasman en el documento en tanto a la consistencia de datos como al rendimiento de la aplicación, tales como tiempos de respuesta.

### Consideraciones de seguridad

Cada usuario deberá autenticarse y su acceso verificado por una sola Terminal para su respectiva labor de acuerdo a lo que su rol especifique. Todas las claves de seguridad deberán estar seguras y en su defecto encriptadas en la base de datos para dar una buena seguridad al sistema y su información.

### Suposiciones y dependencias

Se han definido las siguientes suposiciones:

* + - * + Estará implementada en un servidor web en la nube.
        + Se implementará un modelo independiente, pero se realizará las pruebas sobre un gestor MySQL.
        + Para la comunicación entre el sistema deberá tener acceso a internet.

## Requisitos específicos

### Requisitos funcionales

#### Cliente

RF1. Gestionar usuario: El usuario podrá registrarse al software para poder acceder y contactarse con el asistente virtual.

RF2. Administrar Interacción: El usuario podrá solicitar información, contratar los servicios y/o solicitar soporte técnico.

RF3.Gestionar cancelación de orden de trabajo: Se registrará posibles cancelaciones o reprogramaciones de visitas técnicas agendadas con un técnico.

#### Jefe de Administración Técnica

RF4. Gestionar Técnico: El administrador podrá registrar a los trabajadores y sus diferentes especialidades.

RF5. Gestionar tipo de servicio técnico: El administrador registra los diferentes servicios técnicos que ofrece al cliente para mejorar la calidad del servicio.

RF6. Administrar orden de trabajo: El asistente virtual podrá asignar ordenes de trabajo para los técnicos.

#### Gerente de ventas y servicios

RF7. Administrar planes: El administrador registrara los planes de llamadas, tv cable e internet que ofrece la empresa, también los diferentes combos promocionales que la empresa ofrece.

RF8. Gestionar roles y privilegios: El administrador podrá asignar permisos para controlar el acceso al sistema.

RF9. Administrar Chat. - Se realizará el análisis de la conversación que mantuvo el usuario con el asistente virtual

RF10. Gestionar contrato: Se registrará los clientes que decidieron contratar alguno de nuestros servicios.

RF11. Gestionar Servicio Contratado: Se tendrá un registro sobre qué servicio tiene contratado cada cliente.

### Requisitos no funcionales

* **Rendimiento**

El tiempo de respuesta de la aplicación web no deberá ser mayor a 10 segundos.

* **Seguridad**

Encriptado de contraseñas en la base de datos

Garantizar la confiabilidad y el desempeño del sistema realizando validaciones tanto en el cliente como en el servidor.

* **Fiabilidad**

El software deberá tener una interfaz clara y sencilla

* **Disponibilidad**

La disponibilidad del software deberá estar disponible las 24 horas del día, garantizando la información que requiera el usuario en su momento

* **Mantenibilidad**

Deberá disponer de una documentación sobre las operaciones que realiza el software para que cualquier encargado pueda realizar el mantenimiento con el mínimo esfuerzo.

El software deberá estar basado en un estándar de programación.

* **Portabilidad**

La aplicación web se adaptará en cualquier sistema operativo.

# CAPÍTULO 5

1. Captura de requisitos

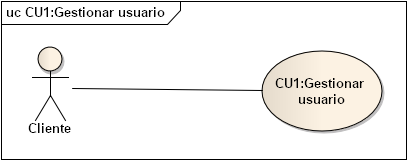
## Priorización de los casos de uso

|  |  |
| --- | --- |
| Estado | **Aprobado**: Es usado cuando el caso de uso esta entre los principales en construirse.  **Incorporado**: Es usado cuando el caso de uso esta entre los próximos a realizarse o tomarse en cuenta.  **Propuesto**: No es considerado inicialmente. |
| Prioridad | **Normal**: El caso de uso se realiza inicialmente.  **Critico**: Cuando el caso de uso requiere más información que le será proporcionada de otro caso de uso.  **Significativo**: Conforme a la necesidad de ver información con las vistas posibles. |
| Riesgo | **Normal**: Se usa en los casos de uso básicos de registros de datos.  **Critico**: Cuando el caso de uso necesita de mucha información para realizarse.  **Accesorio**: Cuando el caso de uso es base para otros. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caso de uso** | **Actor** | **Estado** | **Prioridad** | **Riesgo** |
| CU1: Gestionar usuario | Cliente | Aprobado | Normal | Normal |
| CU2: Administrar Interacción | Cliente | Aprobado | Critico | Critico |
| CU3: Gestionar Técnico | Jefe de administración técnica | Aprobado | Normal | Normal |
| CU4: Administrar planes | Gerente de ventas y servicios | Aprobado | Normal | Normal |
| CU5: Gestionar tipo de servicio técnico | Jefe de administración técnica | Aprobado | Normal | Normal |
| CU6: Gestionar roles y privilegios | Gerente de ventas y servicios | Aprobado | Normal | Normal |
| CU7: Administrar orden de trabajo | Jefe de administración técnica | Aprobado | Critico | Critico |
| CU8: Gestionar contratos solicitados | Gerente de ventas y servicios | Aprobado | Critico | Critico |
| CU9: Gestionar cancelación de orden de trabajo | Cliente | Aprobado | Normal | Normal |
| CU10: Gestionar Servicio Contratado | Gerente de ventas y servicios | Aprobado | Normal | Normal |
| CU11: Gestionar Chat | Gerente de ventas y servicios | Aprobado | Normal | Normal |

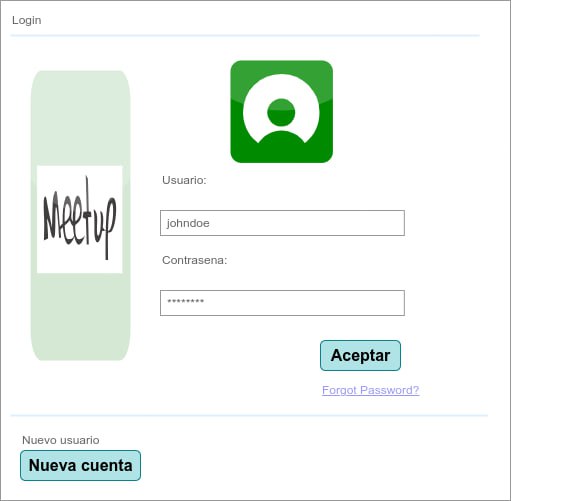
## Detalle de casos de uso

###### CU1: Gestionar Usuario.

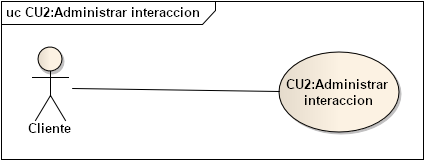


|  |  |
| --- | --- |
| **Caso de uso** | **CU1** Gestionar Usuario |
| **Propósito** | Gestionar los datos que registra el usuario. |
| **Resumen** | Se registrará los datos de los usuarios, para usuarios nuevos y con esos datos poder iniciar sesión. |
| **Actor iniciador** | Cliente |
| **Precondición** | Ninguna |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * El usuario puede registrarse como usuario nuevo. * El usuario podrá actualizar y cambiar sus datos. * El usuario podrá iniciar sesión con su diferente tipo de rol, como ser: cliente, administrador. |
| **Excepción** | Ninguna. |

**Prototipo**

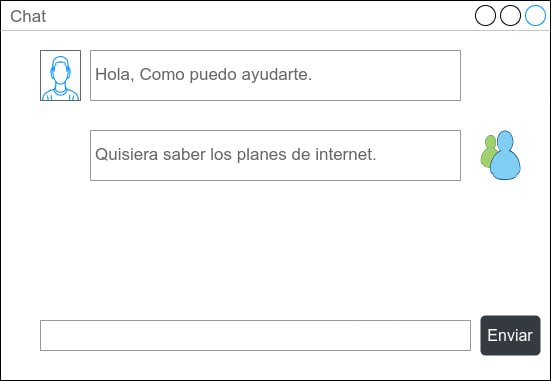


###### CU2: Administrar Interacción

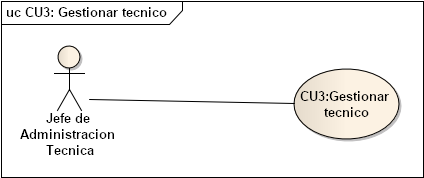


|  |  |
| --- | --- |
| **Caso de uso** | **CU2** Administrar Interacción |
| **Propósito** | Registrar las solicitudes que el usuario pide al asistente virtual. |
| **Resumen** | Se tendrá un registro de las interacciones que solicita el usuario, cuando interactúa con el sistema y es atendido de manera automática. |
| **Actor iniciador** | Cliente |
| **Precondición** | Gestionar Usuario |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * Se registrará cada interacción que solicita el usuario * Brindará el servicio de automatización al contratar un plan ya sea de internet, tv cable o de telefonía. * También se registrará las solicitudes de soporte técnico que requiera el cliente. |
| **Excepción** | Ninguna. |

**Prototipo**



###### CU3: Gestionar Técnico

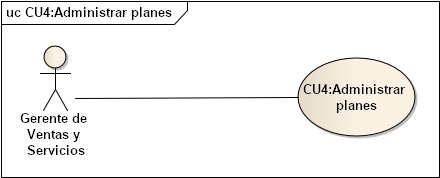


|  |  |
| --- | --- |
| **Caso de uso** | **CU2 Gestionar Técnico** |
| **Propósito** | Registrar a todos los técnicos de la empresa |
| **Resumen** | Se tendrá un registro de los empleados que tiene la empresa y brindan soporte técnico a los clientes. |
| **Actor iniciador** | Jefe de Administración Técnica |
| **Precondición** | Gestionar Usuario |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * Se registrarán técnicos según sus especialidades * Antes deben estar registrados como usuarios * Se podrá agregar, eliminar y actualizar datos de los técnicos. |
| **Excepción** | Ninguna. |

**Prototipo**

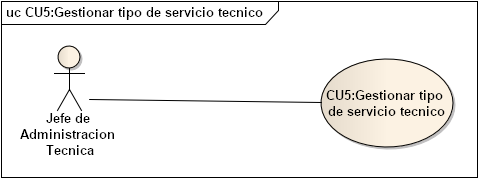


###### CU4: Administrar planes



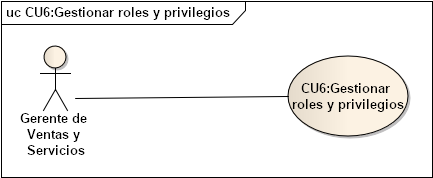
|  |  |
| --- | --- |
| **Caso de uso** | **CU4 Administrar planes** |
| **Propósito** | Registrar los planes de servicio |
| **Resumen** | Se registrará todos los planes de servicios de internet, tv cable y telefonía que ofrece la empresa a sus clientes. |
| **Actor iniciador** | Gerente de Ventas y Servicios |
| **Precondición** | Ninguna |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * El administrador podrá registrar los planes de servicio que ofrece la empresa de telefonía, además puede crear, editar y actualizar información de los planes, combos y/o promociones. |
| **Excepción** | Ninguna. |

###### CU5.- Gestionar Tipo de servicio técnico



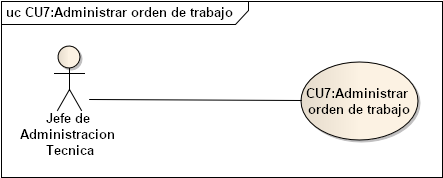
|  |  |
| --- | --- |
| **Caso de uso** | **CU5 Gestionar tipo de servicio técnico** |
| **Propósito** | Registrar todos los tipos de servicios técnicos |
| **Resumen** | Se registrará todos los tipos de servicio técnicos que ofrece la empresa para mejorar el servicio que recibe el cliente, además poder tener un registro de que funciones cumplen los técnicos. |
| **Actor iniciador** | Jefe de Administración Técnica |
| **Precondición** | Ninguna |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * El administrador registrara todos los tipos de servicios técnicos que se ofrecen a los clientes. * Se tendrá un registro de todas las solicitudes que requieran un servicio técnico. * Se podrán agregar, editar y borrar los tipos de servicios técnicos. |
| **Excepción** | Ninguna. |

###### CU6.- Gestionar roles y privilegios



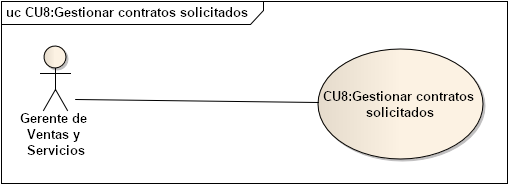
|  |  |
| --- | --- |
| **Caso de uso** | **CU6:** Gestionar roles y privilegios |
| **Propósito** | Gestionar los datos del paciente |
| **Resumen** | Se podrá ver todos los usuarios con sus respectivos privilegios que tiene un usuario, esto nos permite tener un control de que es lo que un usuario puede hacer en el sistema. |
| **Actor iniciador** | Gerente de Ventas y Servicios |
| **Precondición** | Ninguna |
| **Post Condición** | Ninguna |
| **Flujo Principal** | 1. Lo primero que debe hacer es entrar al sistema como un administrador 2. Debe ir a la pestaña que dice privilegios 3. Presionar el botón de Guardar 4. Ahora debe iniciar sesión con el reconocimiento facial |
| **Excepción** | Ninguna |

###### CU7.- Administrar orden de trabajo



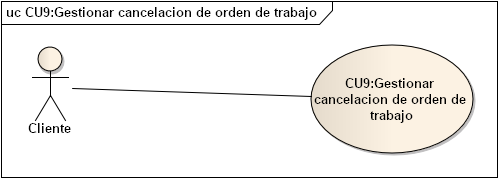
|  |  |
| --- | --- |
| **Caso de uso** | **CU7:** Administrar orden de trabajo |
| **Propósito** | Administra a la asignación de una orden de trabajo |
| **Resumen** | Se podrá ver todos los registros de las ordenes de trabajo que se tienen por atender. |
| **Actor iniciador** | Jefe de Administración Técnica |
| **Precondición** | Ninguna |
| **Post Condición** | Gestionar cliente |
| **Flujo Principal** | 1. Lo primero que debe hacer es entrar a la pagina. 2. Debe asignar a un técnico a la solicitud. 3. Presionar el botón de Guardar |
| **Excepción** | Ninguna |

###### CU8.- Gestionar contratos solicitados



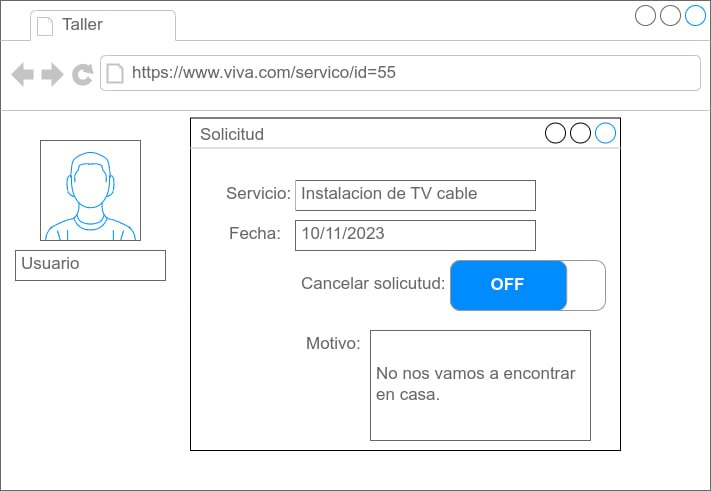
|  |  |
| --- | --- |
| **Caso de uso** | **CU8:** Gestionar contratos solicitados |
| **Propósito** | Gestionar los contratos que se llevaron a cabo |
| **Resumen** | Se podrá ver todos los registros de los servicios contratados por un cliente, que se llevaron a cabo en la interacción. |
| **Actor iniciador** | Gerente de Ventas y Servicios |
| **Precondición** | Administrar interacción |
| **Post Condición** | Gestionar ficha de atención |
| **Flujo Principal** | 1. Lo primero que debe hacer es entrar a la pagina 2. Debe validar el contrato 3. Presionar el botón de Guardar |
| **Excepción** | Ninguna |

###### CU9.- Gestionar cancelación de orden de trabajo

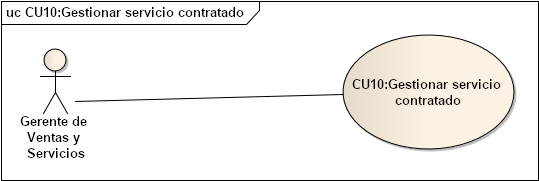


|  |  |
| --- | --- |
| **Caso de uso** | **CU9:** Gestionar cancelación de orden de trabajo |
| **Propósito** | Cancelar una solicitud de orden de trabajo |
| **Resumen** | El cliente podrá cancelar una solicitud de orden de trabajo |
| **Actor iniciador** | Paciente |
| **Precondición** | Administrar orden de trabajo |
| **Post Condición** | Ninguna |
| **Flujo Principal** | 1. Lo primero que debe hacer es entrar a la pagina 2. Debe dar seleccionar el checkbox para así cancelar la orden de trabajo 3. Presionar el botón de Guardar |
| **Excepción** | Ninguna |

**Prototipo**

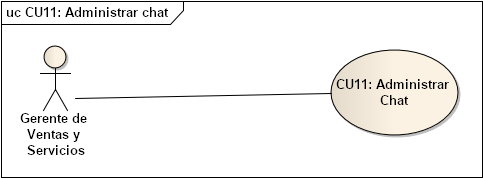


###### CU10.- Gestionar Servicio Contratado

******

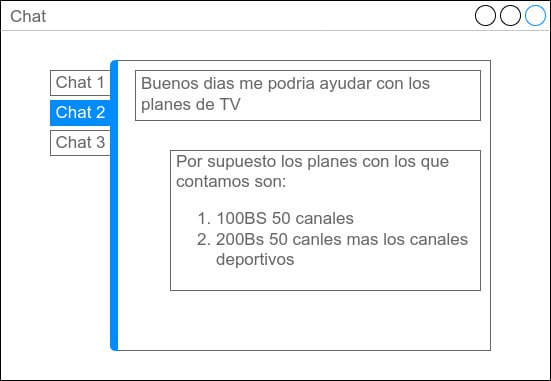
|  |  |
| --- | --- |
| **Caso de uso** | **CU10:** Gestionar servicio contratado |
| **Propósito** | Registrar los servicios que se va a poner a disposición de los clientes |
| **Resumen** | Se registrarán los servicios de los que los usuarios podrán disponer para que puedan tener un contrato en donde se indica que servicios fueron contratados |
| **Actor iniciador** | Gerente de Ventas y Servicios |
| **Precondición** | Ninguna |
| **Post Condición** | Administrar interacción |
| **Flujo Principal** | 1. Lo primero que debe hacer es entrar a la página 2. Debe llenar el formulario con los datos que son requeridos 3. Presionar el botón de Guardar |
| **Excepción** | Ninguna |

###### CU11.- Gestionar Chat

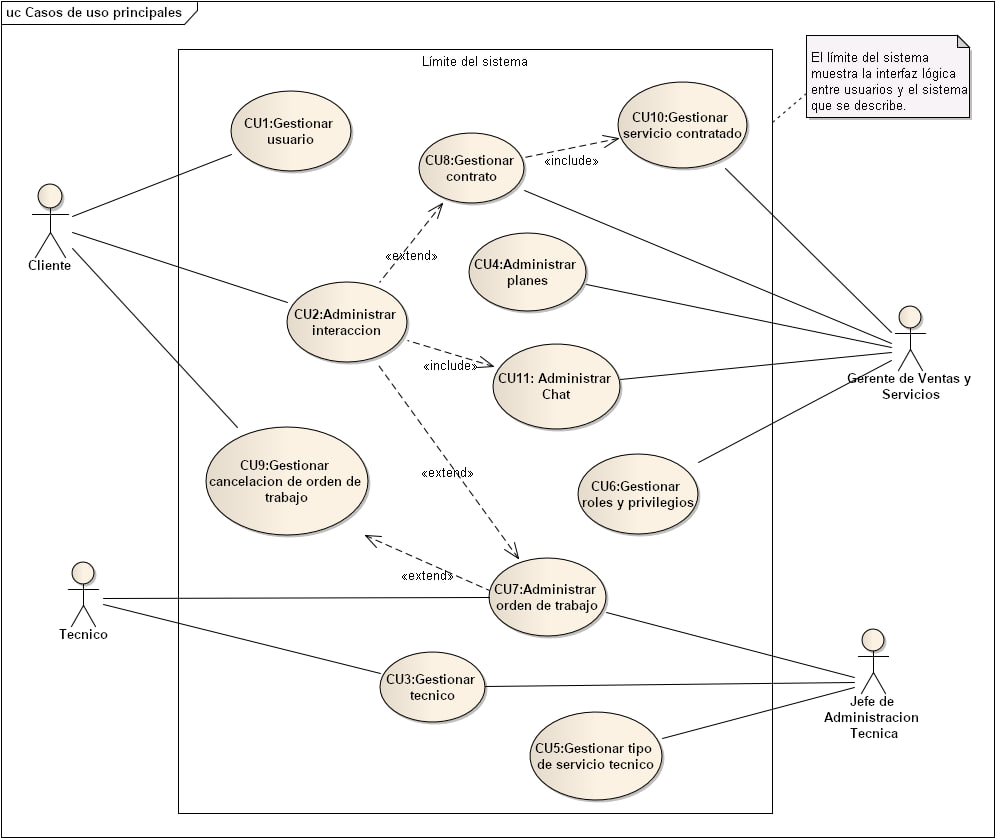


|  |  |
| --- | --- |
| **Caso de uso** | **CU11:** Gestionar Chat |
| **Propósito** | Administrar toda la información brindada por el usuario |
| **Resumen** | Se tendrá registrado toda la conversación que tendrán el usuario y el asistente virtual para su posterior análisis de la conversación y validar el desempeño de nuestros servicios. |
| **Actor iniciador** | Cliente |
| **Precondición** | Gestionar cliente |
| **Post Condición** | Ninguna |
| **Flujo Principal** | * El usuario ingresa al software e interactúa con el asistente virtual. * Finalizada la conversación se realiza una evaluación del lenguaje natural de cada frase que dijo el usuario. * Al final se saca un porcentaje promedio de la satisfacción del usuario con el servicio. |
| **Excepción** | Ninguna |

**Prototipo**



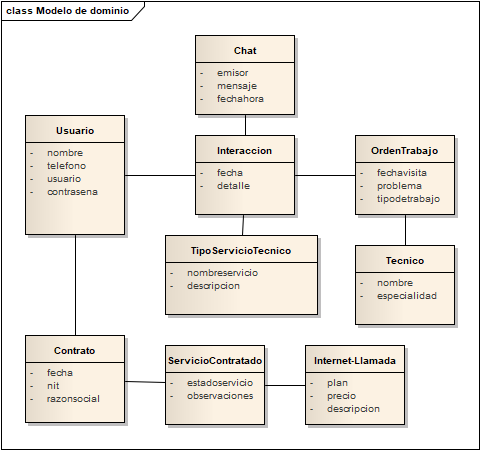
###### Diagrama general de casos de uso



# CAPÍTULO 6

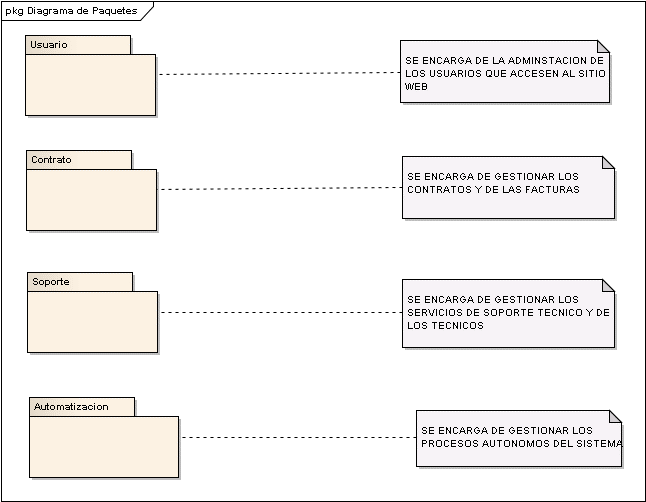
1. Análisis

## Modelo de dominio inicial

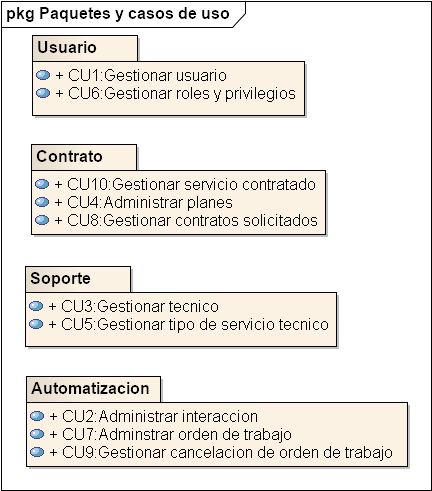


## Análisis de arquitectura

### Identificación de paquetes

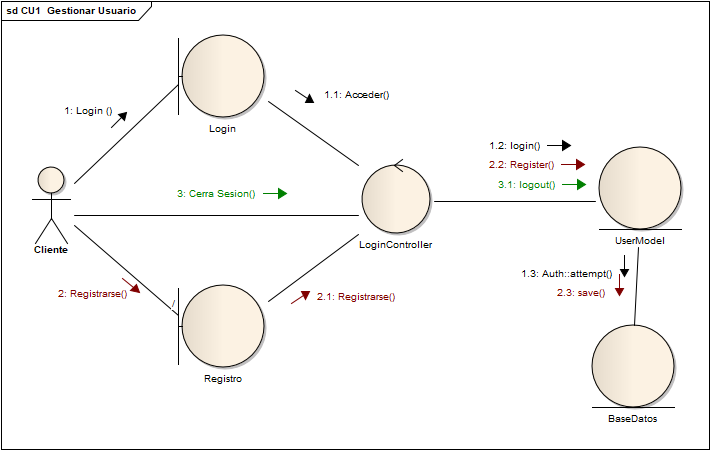


### Análisis de paquete

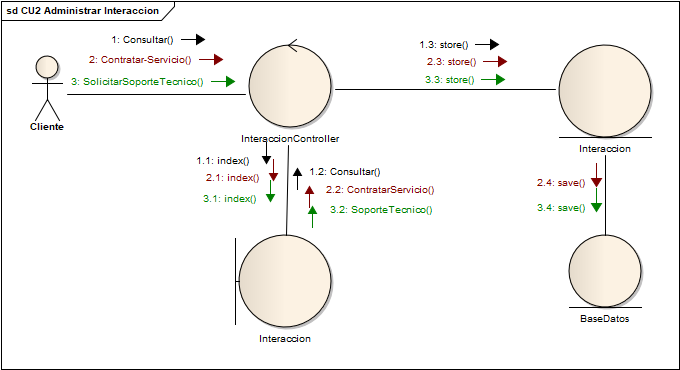


## Análisis de casos de uso

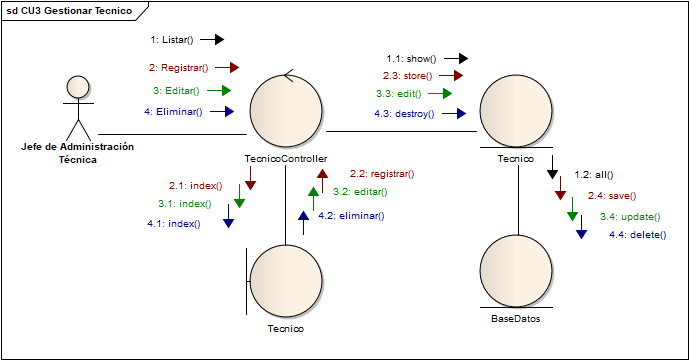
###### CU1: Gestionar usuario



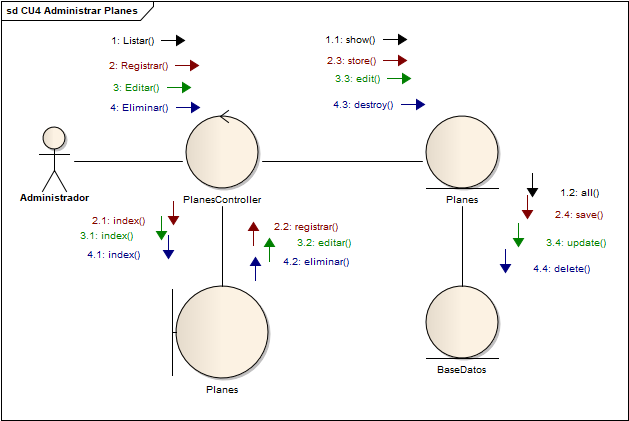
###### CU2: Administrar interacción



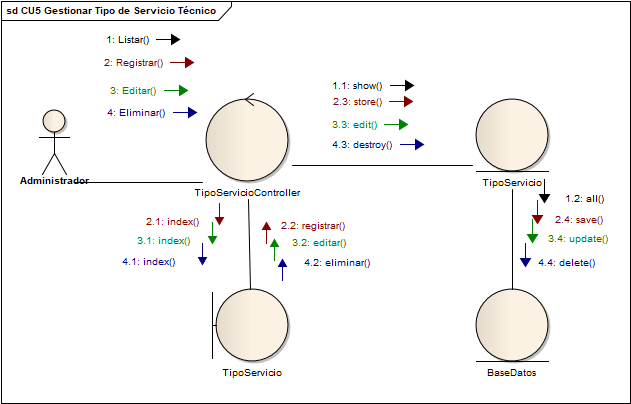
###### CU3: Gestionar técnico



###### CU4: Gestionar planes

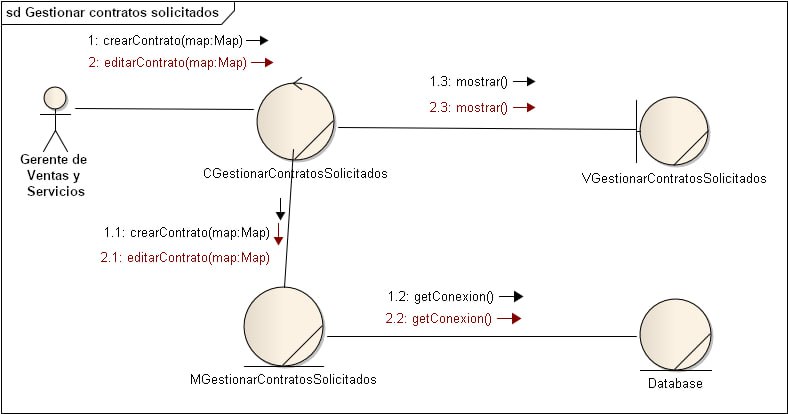


###### CU5: Gestionar tipo de servicio técnico

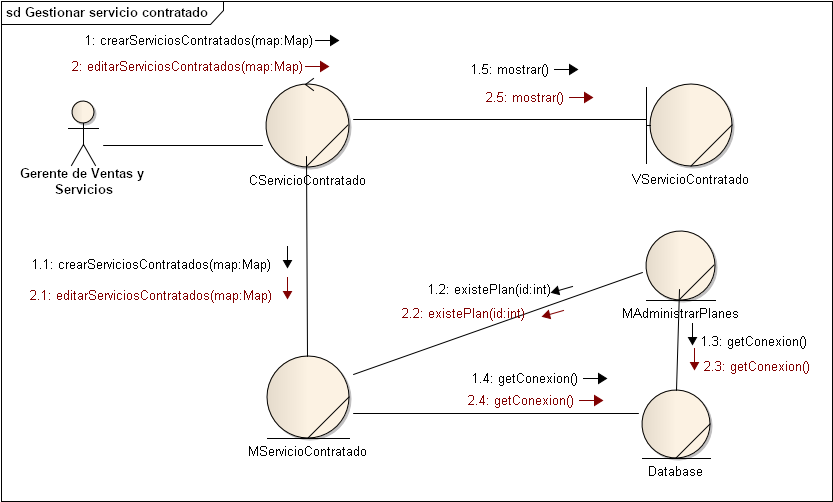


###### CU7: Administrar orden de trabajo

###### CU8: Gestionar contratos solicitados

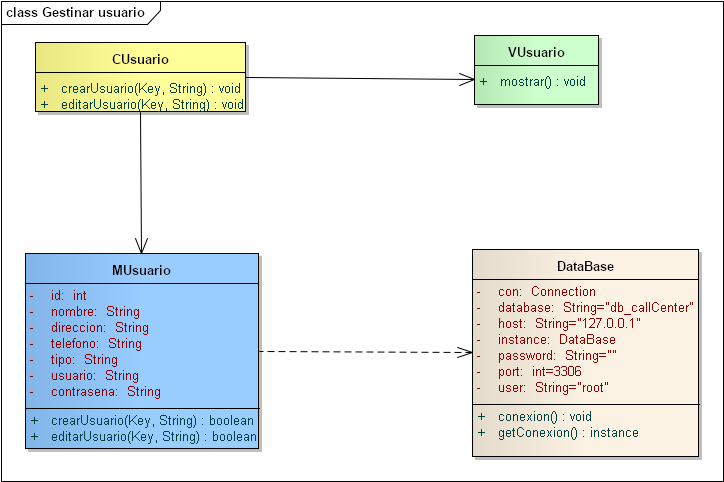


###### CU10: Gestionar servicio contratado

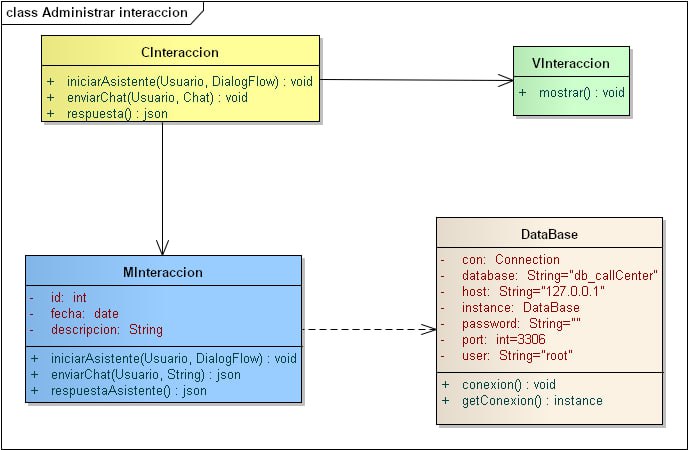


## Análisis de clases

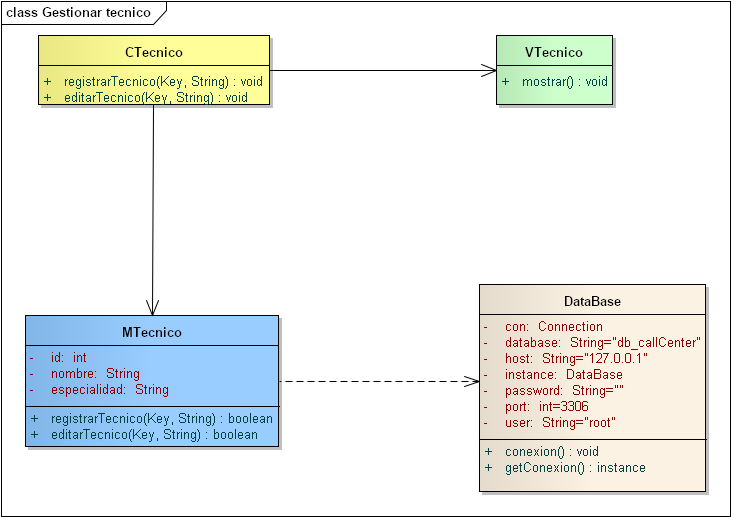
###### CU1: Gestionar usuario



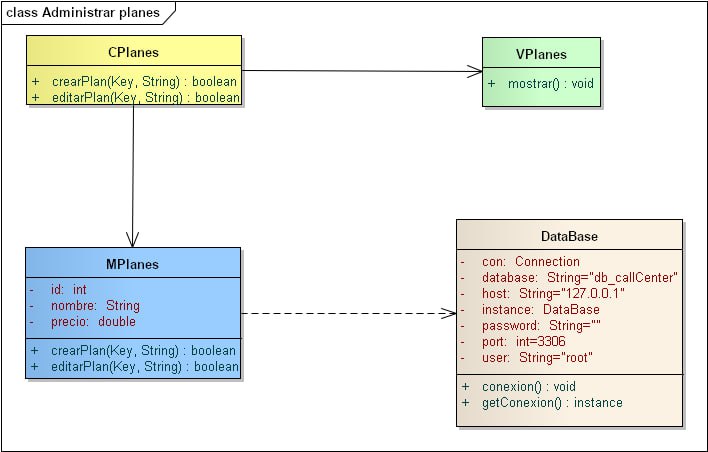
###### CU2: Administrar interacción



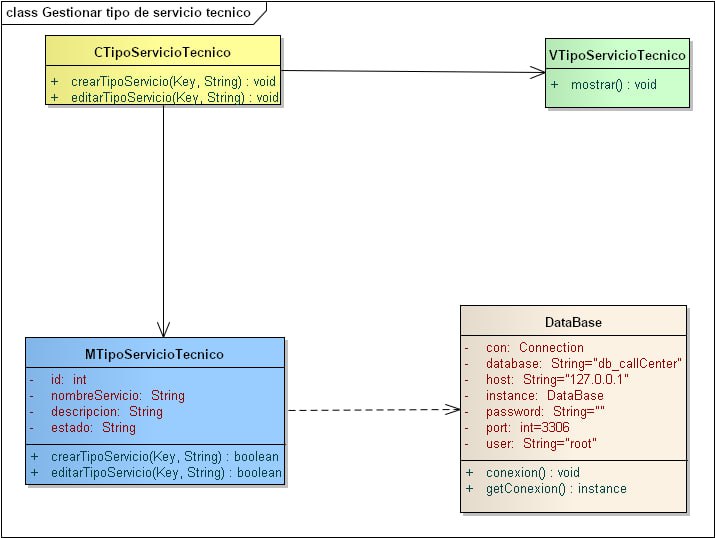
###### CU3: Gestionar técnico



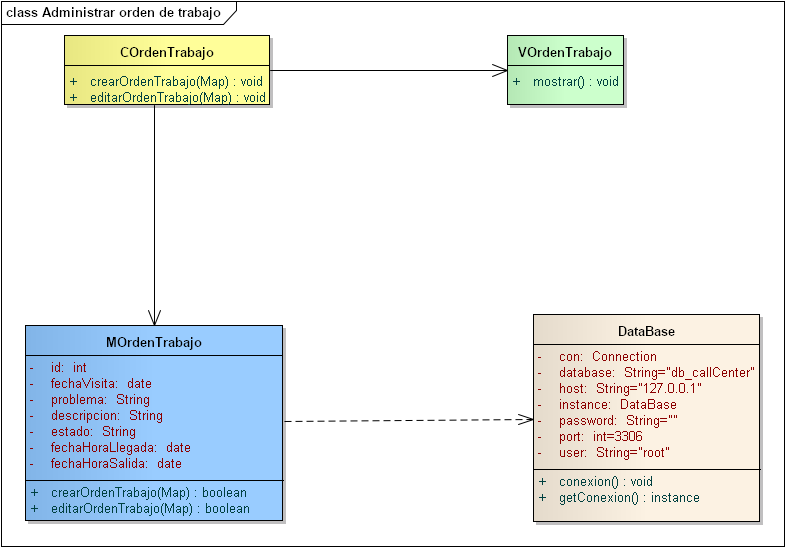
###### CU4: Gestionar planes



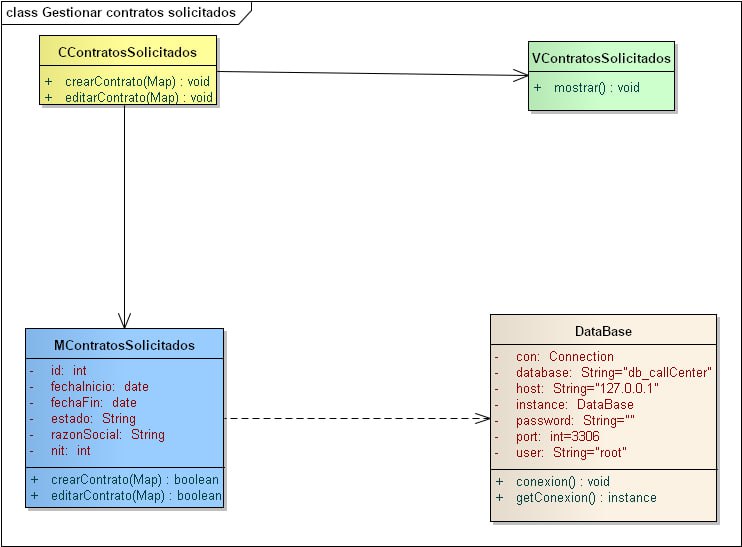
###### CU5: Gestionar tipo de servicio técnico



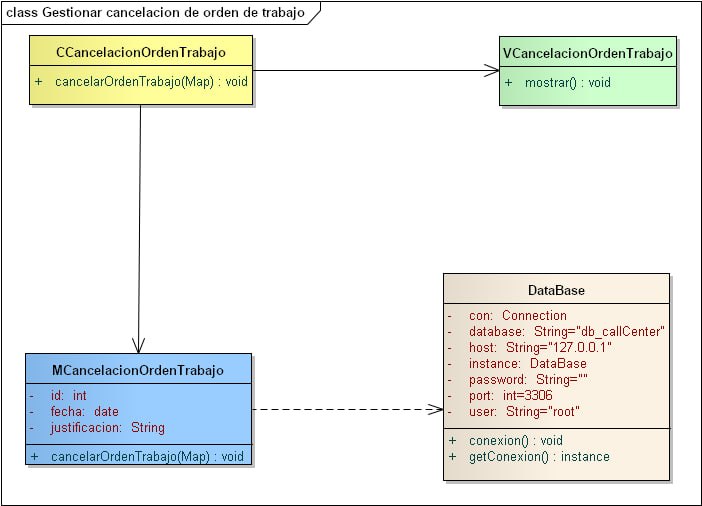
###### CU7: Administrar orden de trabajo



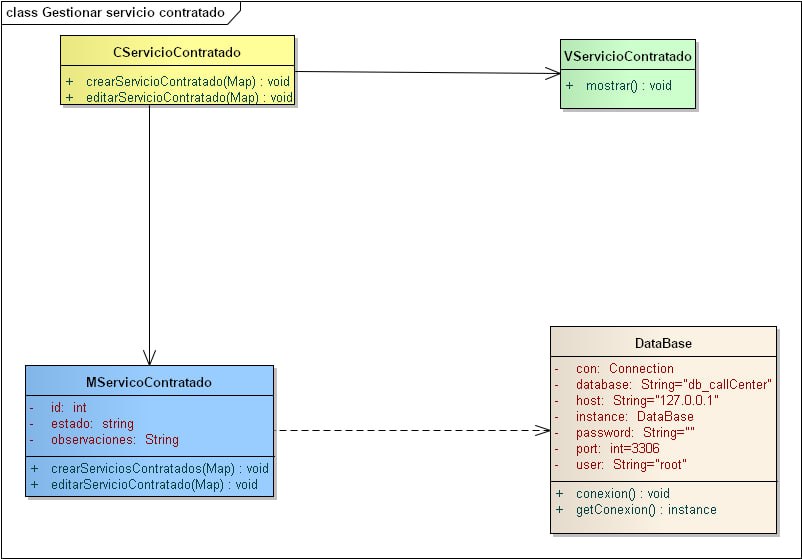
###### CU8: Gestionar contratos solicitados



###### CU9: Gestionar cancelación de orden de trabajo



###### CU10: Servicio contratado



###### CU11: Administrar chat

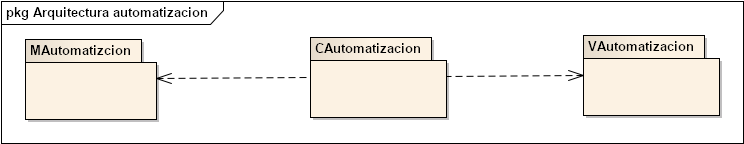


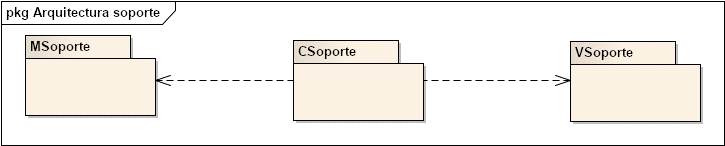
# CAPÍTULO 7

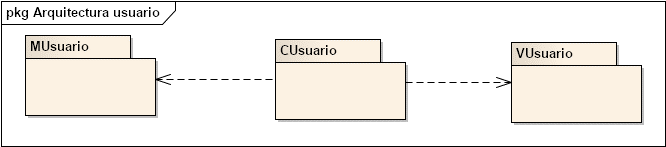
1. DISEÑO

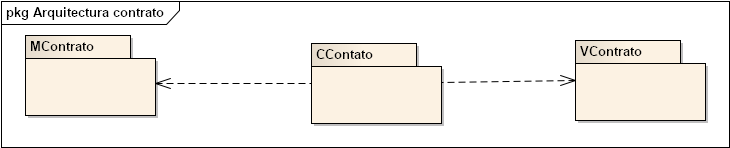
## Diseño de la arquitectura

## Diseño de paquetes

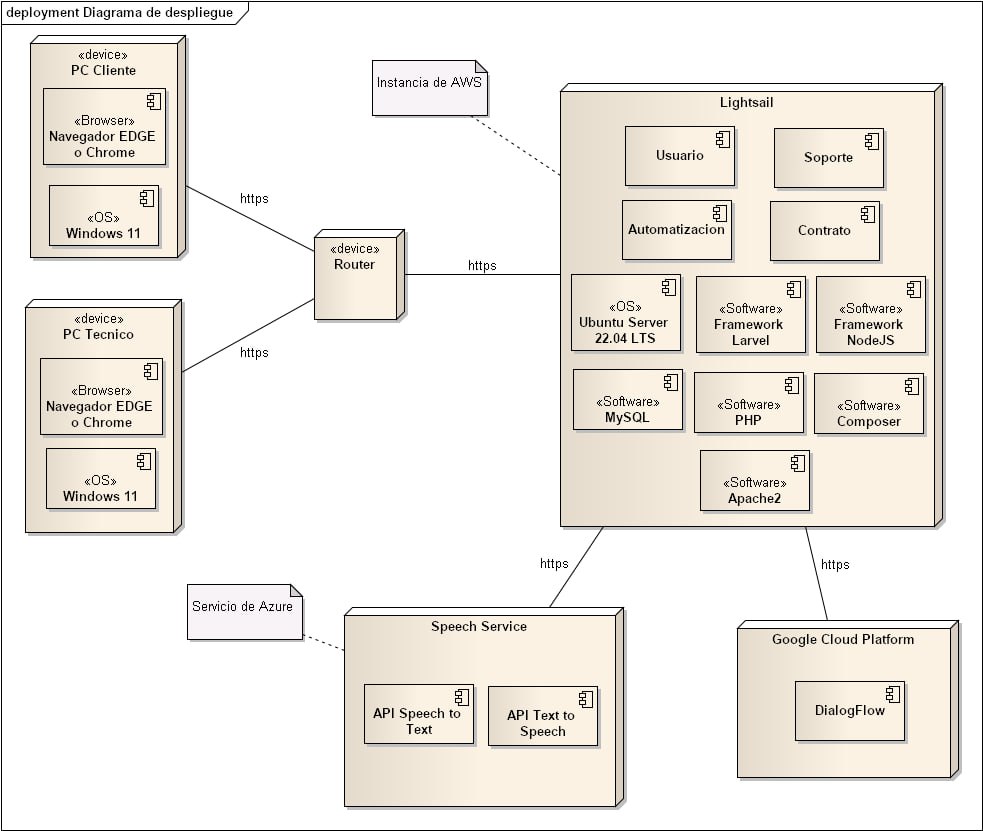




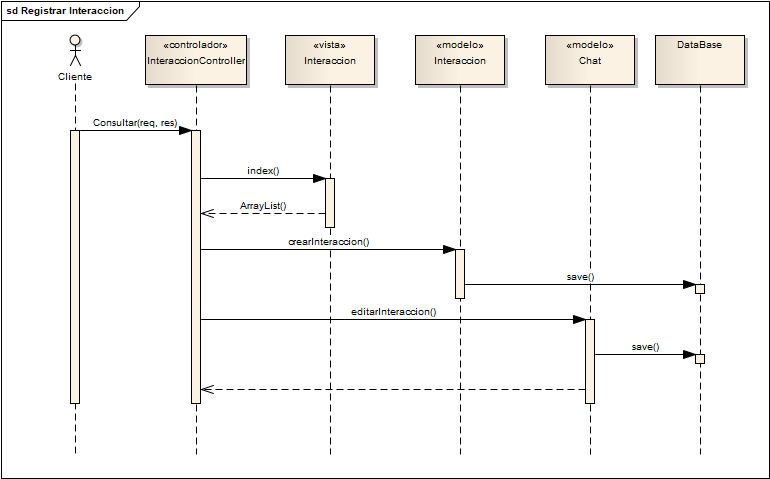


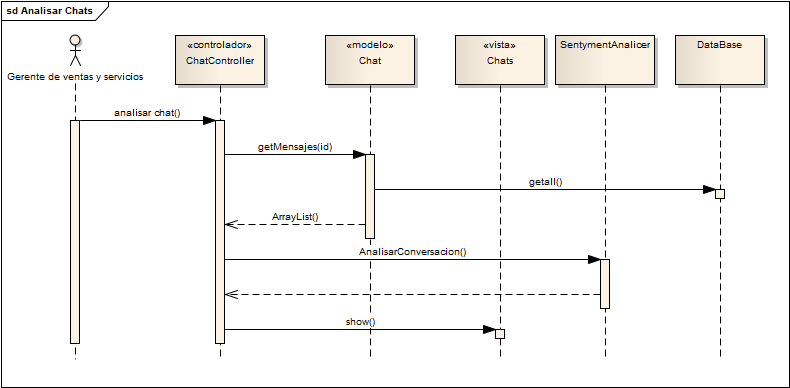


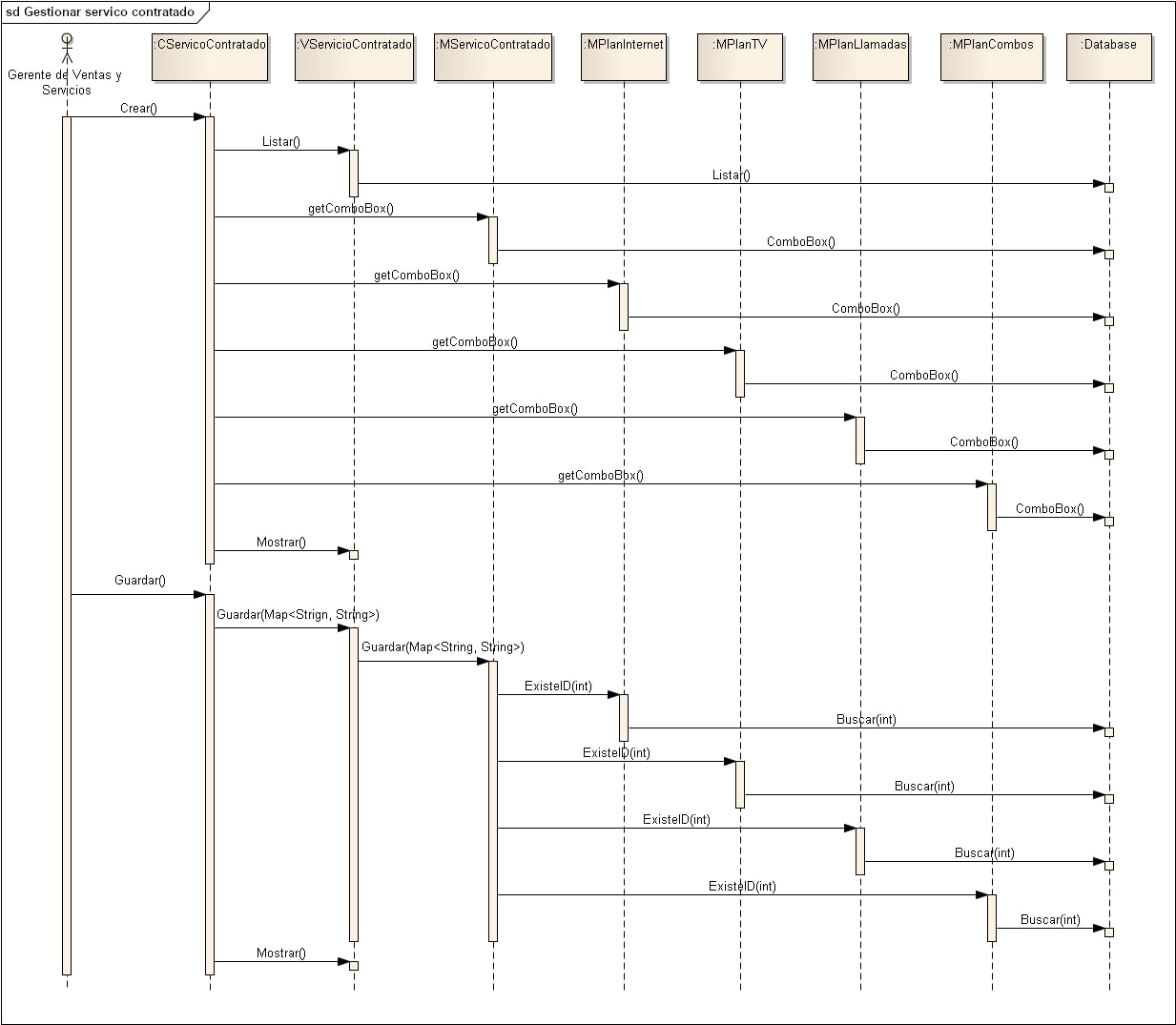
## Diseño de la arquitectura física

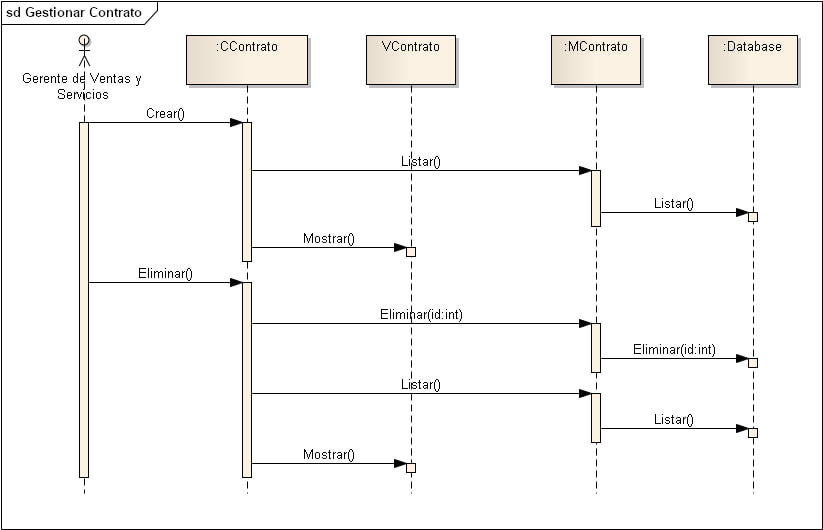


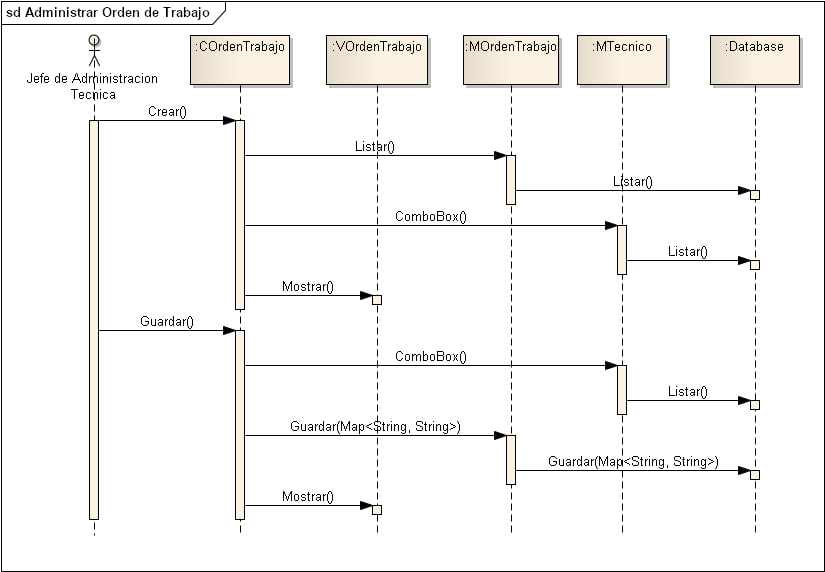
## DISEÑO DE CASOS DE USO







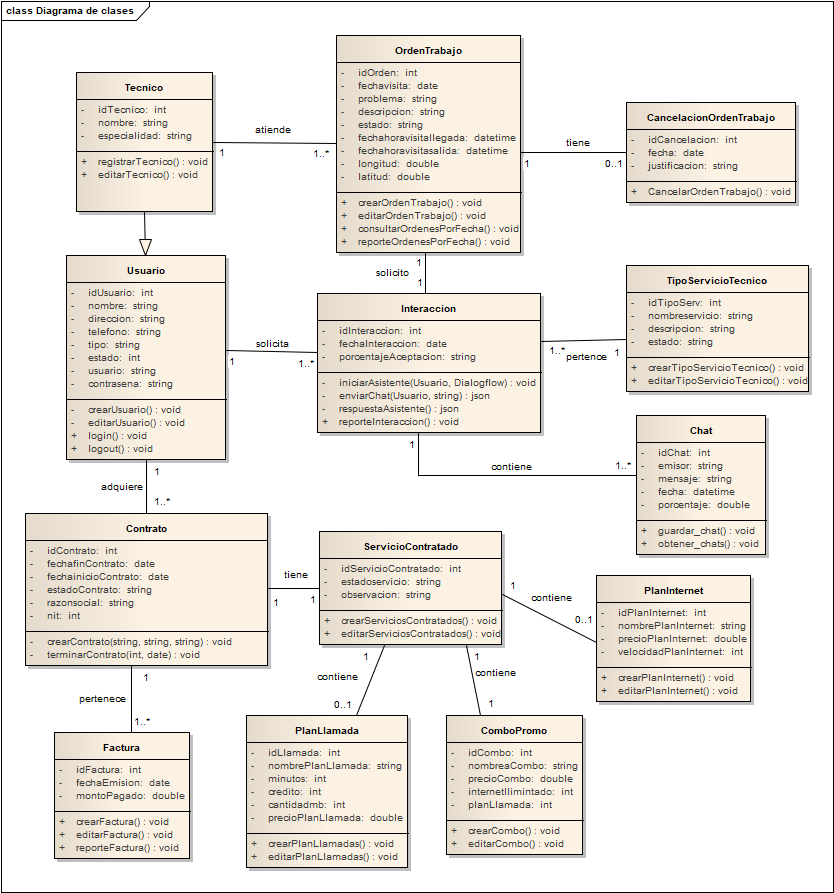




## Diagrama de la base de datos

### Diseño conceptual

###### DIAGRAMA DE CLASES



### Diseño lógico

Usuario

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PK** |  |  |  |  |  |  |
| id | nombre | dirección | teléfono | tipo | usuario | contraseña |

Interacción

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PK** |  |  | **FK** | **FK** |
| id | fecha | detalle | id\_usuario | id\_TipoServicio |

TipoServicioTecnico

|  |  |  |  |
| --- | --- | --- | --- |
| **PK** |  |  |  |
| id | nombre | descripción | estado |

Técnico

|  |  |  |  |
| --- | --- | --- | --- |
| **PK** |  |  | FK |
| id | nombre | especialidad | Id\_usuario |

OrdenTrabajo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PK** |  |  |  |  |  |  | FK | FK |
| id | fechaVisita | problema | descripción | estado | llegada | salida | id\_tecnico | id\_interaccion |

CancelacionOrdenTrabajo

|  |  |  |  |
| --- | --- | --- | --- |
| **PK** |  |  | **FK** |
| id | fecha | justificación | id\_OrdenTrabajo |

ServicioContratado

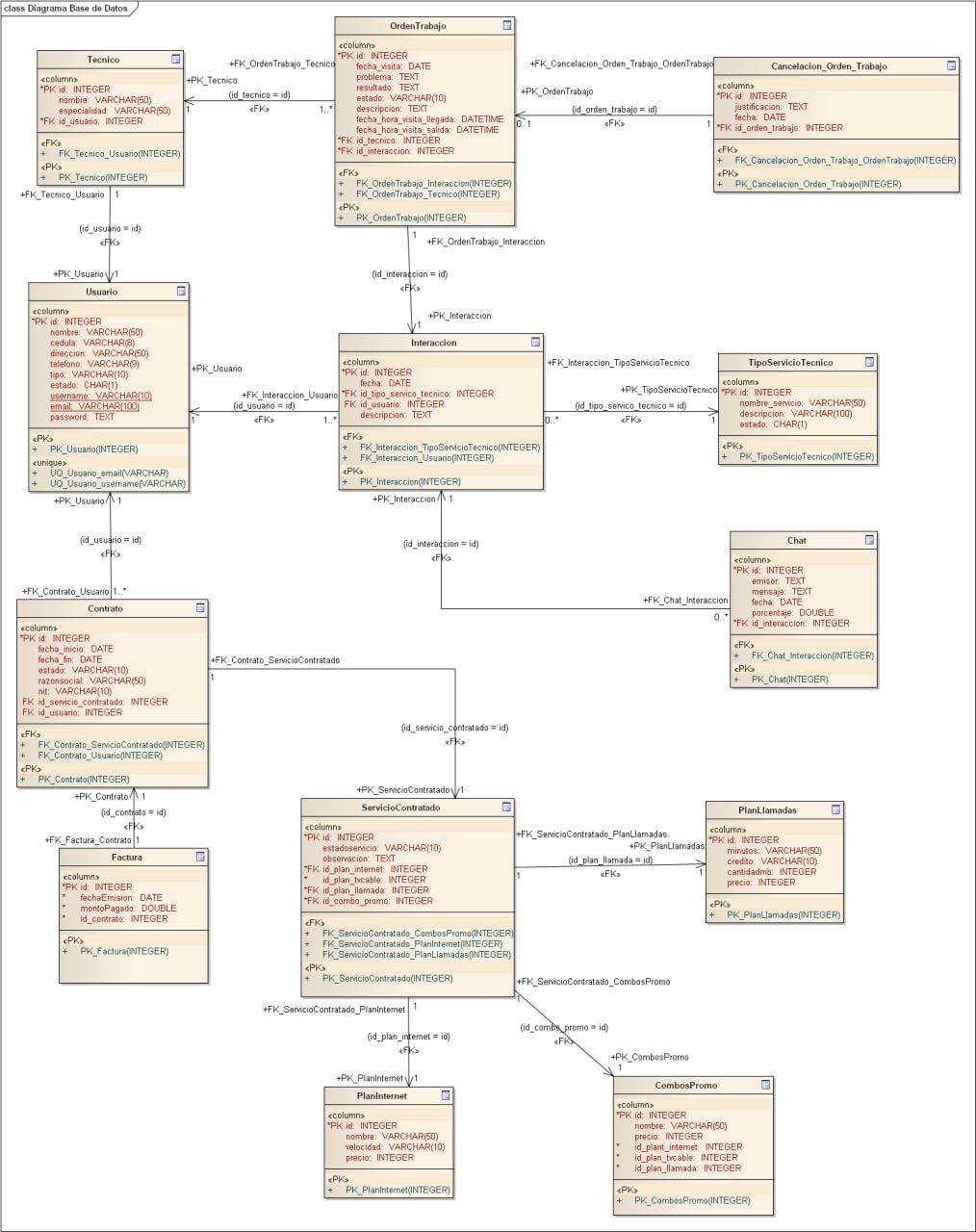
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PK** |  |  | **FK** | **FK** | **FK** | **FK** |
| id | estado | observaciones | id\_PlanLlamada | id\_PlanTvCable | id\_PlanInternet | id\_ComboPromo |

Contrato

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PK** |  |  |  |  |  | **FK** |
| id | fechaInicio | fechaFin | estado | razonSocial | nit | id\_ServicioContratado |

### Diseño físico

#### Diagrama de la base de datos



#### Script de la base de datos

-- phpMyAdmin SQL Dump

-- version 5.1.1

-- https://www.phpmyadmin.net/

-- Servidor: 127.0.0.1:3306

-- Tiempo de generación: 12-07-2023 a las 22:52:22

-- Versión del servidor: 10.5.19-MariaDB-cll-lve

-- Versión de PHP: 7.2.34

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Base de datos: `u300302440\_db\_callcenter`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `cancelacion\_ordentrabajo`

--

CREATE TABLE `cancelacion\_ordentrabajo` (

`id` bigint(20) UNSIGNED NOT NULL,

`justification` text NOT NULL,

`fecha` date NOT NULL,

`id\_orden\_trabajo` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `combos\_promo`

--

CREATE TABLE `combos\_promo` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre` varchar(191) NOT NULL,

`precio` int(11) NOT NULL,

`id\_plan\_internet` bigint(20) UNSIGNED NOT NULL,

`id\_plan\_tvcable` bigint(20) UNSIGNED NOT NULL,

`id\_plan\_llamada` bigint(20) UNSIGNED NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- -------------------------------------------------------

-- Estructura de tabla para la tabla `contrato`

--

CREATE TABLE `contrato` (

`id` bigint(20) UNSIGNED NOT NULL,

`fecha\_inicio` date NOT NULL,

`fecha\_fin` date NOT NULL,

`estado` varchar(191) NOT NULL,

`nombre\_facturacion` varchar(191) NOT NULL,

`nit` varchar(191) NOT NULL,

`id\_servicio\_contratado` bigint(20) UNSIGNED DEFAULT NULL,

`id\_usuario` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `factura`

--

CREATE TABLE `factura` (

`id` bigint(20) UNSIGNED NOT NULL,

`fecha` date NOT NULL,

`monto` int(11) NOT NULL,

`id\_servicio\_contratado` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `failed\_jobs`

--

CREATE TABLE `failed\_jobs` (

`id` bigint(20) UNSIGNED NOT NULL,

`uuid` varchar(191) NOT NULL,

`connection` text NOT NULL,

`queue` text NOT NULL,

`payload` longtext NOT NULL,

`exception` longtext NOT NULL,

`failed\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `interaccion`

--

CREATE TABLE `interaccion` (

`id` bigint(20) UNSIGNED NOT NULL,

`fecha` date NOT NULL,

`descripcion` text NOT NULL,

`id\_tipo\_servicio\_tecnico` bigint(20) UNSIGNED DEFAULT NULL,

`id\_usuario` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `migrations`

--

CREATE TABLE `migrations` (

`id` int(10) UNSIGNED NOT NULL,

`migration` varchar(191) NOT NULL,

`batch` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `migrations`

--

INSERT INTO `migrations` (`id`, `migration`, `batch`) VALUES

(1, '2014\_10\_12\_000000\_create\_users\_table', 1),

(2, '2014\_10\_12\_100000\_create\_password\_resets\_table', 1),

(3, '2019\_08\_19\_000000\_create\_failed\_jobs\_table', 1),

(4, '2019\_12\_14\_000001\_create\_personal\_access\_tokens\_table', 1),

(5, '2023\_07\_05\_154248\_planinternets', 1),

(6, '2023\_07\_05\_190852\_plan\_tv\_cables', 1),

(7, '2023\_07\_05\_193725\_combo\_tv\_internet\_telef', 2),

(8, '2023\_07\_05\_193750\_plan\_llamadas', 2),

(9, '2023\_07\_05\_194436\_tecnicos', 3),

(10, '2023\_07\_05\_194831\_tipo\_servicios\_tecnicos', 4),

(11, '2023\_07\_12\_174641\_combos\_promo', 5),

(12, '2023\_07\_12\_180315\_servicio\_contratado', 6),

(13, '2023\_07\_12\_181024\_factura', 7),

(14, '2023\_07\_12\_181525\_contrato', 8),

(15, '2023\_07\_12\_182135\_interaccion', 9),

(16, '2023\_07\_12\_182654\_ordentrabajo', 10),

(17, '2023\_07\_12\_183147\_cancelacion\_ordentrabajo', 11);

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `ordentrabajo`

--

CREATE TABLE `ordentrabajo` (

`id` bigint(20) UNSIGNED NOT NULL,

`fecha\_visita` date NOT NULL,

`problema` text NOT NULL,

`resultado` text NOT NULL,

`estado` varchar(191) NOT NULL,

`descripcion` text NOT NULL,

`fecha\_hora\_visita\_llegada` datetime NOT NULL,

`fecha\_hora\_visita\_salida` datetime NOT NULL,

`id\_tecnico` bigint(20) UNSIGNED DEFAULT NULL,

`id\_interaccion` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `password\_resets`

--

CREATE TABLE `password\_resets` (

`email` varchar(191) NOT NULL,

`token` varchar(191) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `personal\_access\_tokens`

--

CREATE TABLE `personal\_access\_tokens` (

`id` bigint(20) UNSIGNED NOT NULL,

`tokenable\_type` varchar(191) NOT NULL,

`tokenable\_id` bigint(20) UNSIGNED NOT NULL,

`name` varchar(191) NOT NULL,

`token` varchar(64) NOT NULL,

`abilities` text DEFAULT NULL,

`last\_used\_at` timestamp NULL DEFAULT NULL,

`expires\_at` timestamp NULL DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `planinternets`

--

CREATE TABLE `planinternets` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre` varchar(50) NOT NULL,

`velocidad` varchar(10) NOT NULL,

`precio` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `planinternets`

--

INSERT INTO `planinternets` (`id`, `nombre`, `velocidad`, `precio`, `created\_at`, `updated\_at`) VALUES

(1, 'Plan plata', '15 mb', 150, '2023-07-10 17:43:30', '2023-07-10 17:43:30');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `plan\_llamadas`

--

CREATE TABLE `plan\_llamadas` (

`id` bigint(20) UNSIGNED NOT NULL,

`minutos` varchar(50) NOT NULL,

`credito` varchar(10) NOT NULL,

`cantidadmb` int(11) NOT NULL,

`precio` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `plan\_llamadas`

--

INSERT INTO `plan\_llamadas` (`id`, `minutos`, `credito`, `cantidadmb`, `precio`, `created\_at`, `updated\_at`) VALUES

(1, '200', '100', 4000, 160, '2023-07-10 17:46:51', '2023-07-10 17:46:51');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `plan\_tv\_cables`

--

CREATE TABLE `plan\_tv\_cables` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre` varchar(50) NOT NULL,

`precio` int(11) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `plan\_tv\_cables`

--

INSERT INTO `plan\_tv\_cables` (`id`, `nombre`, `precio`, `created\_at`, `updated\_at`) VALUES

(1, 'Plan essencial', 80, '2023-07-10 17:44:00', '2023-07-10 17:44:00');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `servicio\_contratado`

--

CREATE TABLE `servicio\_contratado` (

`id` bigint(20) UNSIGNED NOT NULL,

`estadoservicio` varchar(191) NOT NULL,

`observacion` text NOT NULL,

`id\_plan\_internet` bigint(20) UNSIGNED DEFAULT NULL,

`id\_plan\_tvcable` bigint(20) UNSIGNED DEFAULT NULL,

`id\_plan\_llamada` bigint(20) UNSIGNED DEFAULT NULL,

`id\_combo\_promo` bigint(20) UNSIGNED DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `tecnicos`

--

CREATE TABLE `tecnicos` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre` varchar(50) NOT NULL,

`especialidad` varchar(50) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `tecnicos`

--

INSERT INTO `tecnicos` (`id`, `nombre`, `especialidad`, `created\_at`, `updated\_at`) VALUES

(1, 'Carlos Lopez', 'Instalación domiciliaria', NULL, '2023-07-10 17:49:40');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `tipo\_servicios\_tecnicos`

--

CREATE TABLE `tipo\_servicios\_tecnicos` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre\_servicio` varchar(50) NOT NULL,

`descripcion` text NOT NULL,

`estado` char(1) NOT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `tipo\_servicios\_tecnicos`

--

INSERT INTO `tipo\_servicios\_tecnicos` (`id`, `nombre\_servicio`, `descripcion`, `estado`, `created\_at`, `updated\_at`) VALUES

(1, 'Instalacion', 'Servicio de instalacion domiciliaria', '1', '2023-07-05 19:53:24', '2023-07-05 19:53:24'),

(2, 'Revision Equipos', 'Se enviara un trabajador para revisar las instalaciones domiciliarias y evaluar la calidad del servicios', '1', '2023-07-05 20:02:17', '2023-07-05 20:02:17');

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `users`

--

CREATE TABLE `users` (

`id` bigint(20) UNSIGNED NOT NULL,

`nombre` varchar(50) NOT NULL,

`cedula` varchar(8) NOT NULL,

`direccion` varchar(50) NOT NULL,

`telefono` varchar(9) NOT NULL,

`tipo` varchar(10) NOT NULL,

`estado` char(1) NOT NULL,

`username` varchar(10) NOT NULL,

`email` varchar(191) NOT NULL,

`email\_verified\_at` timestamp NULL DEFAULT NULL,

`password` varchar(191) NOT NULL,

`remember\_token` varchar(100) DEFAULT NULL,

`created\_at` timestamp NULL DEFAULT NULL,

`updated\_at` timestamp NULL DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;

--

-- Volcado de datos para la tabla `users`

--

INSERT INTO `users` (`id`, `nombre`, `cedula`, `direccion`, `telefono`, `tipo`, `estado`, `username`, `email`, `email\_verified\_at`, `password`, `remember\_token`, `created\_at`, `updated\_at`) VALUES

(1, 'Pablo Bustos', '53632111', 'Barrio Equipetrol', '60989453', 'usuario', 'a', 'pp', 'Barrio Equipetrol', NULL, '$2y$10$ZsXSGQbWeDL1pj.meBO96OPsGLh4tPk9yoX8utAWrYF0jjYLzAZ1K', NULL, '2023-07-11 11:18:09', '2023-07-11 11:18:09'),

(2, 'Ramiro Belzu', '7777777', 'Barrio Guapuru', '63636363', 'admin', 'a', 'admin', 'Barrio Guapuru', NULL, '$2y$10$wZSzHrdUoIwmtzbRE6kH/.iW584hzM9Je31sKkjatkeTDb9JehW9.', NULL, '2023-07-11 11:32:40', '2023-07-11 11:32:40');

--

-- Índices para tablas volcadas

--

--

-- Indices de la tabla `cancelacion\_ordentrabajo`

--

ALTER TABLE `cancelacion\_ordentrabajo`

ADD PRIMARY KEY (`id`),

ADD KEY `cancelacion\_ordentrabajo\_id\_orden\_trabajo\_foreign` (`id\_orden\_trabajo`);

--

-- Indices de la tabla `combos\_promo`

--

ALTER TABLE `combos\_promo`

ADD PRIMARY KEY (`id`),

ADD KEY `combos\_promo\_id\_plan\_internet\_foreign` (`id\_plan\_internet`),

ADD KEY `combos\_promo\_id\_plan\_tvcable\_foreign` (`id\_plan\_tvcable`),

ADD KEY `combos\_promo\_id\_plan\_llamada\_foreign` (`id\_plan\_llamada`);

--

-- Indices de la tabla `contrato`

--

ALTER TABLE `contrato`

ADD PRIMARY KEY (`id`),

ADD KEY `contrato\_id\_servicio\_contratado\_foreign` (`id\_servicio\_contratado`),

ADD KEY `contrato\_id\_usuario\_foreign` (`id\_usuario`);

--

-- Indices de la tabla `factura`

--

ALTER TABLE `factura`

ADD PRIMARY KEY (`id`),

ADD KEY `factura\_id\_servicio\_contratado\_foreign` (`id\_servicio\_contratado`);

--

-- Indices de la tabla `failed\_jobs`

--

ALTER TABLE `failed\_jobs`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `failed\_jobs\_uuid\_unique` (`uuid`);

--

-- Indices de la tabla `interaccion`

--

ALTER TABLE `interaccion`

ADD PRIMARY KEY (`id`),

ADD KEY `interaccion\_id\_tipo\_servicio\_tecnico\_foreign` (`id\_tipo\_servicio\_tecnico`),

ADD KEY `interaccion\_id\_usuario\_foreign` (`id\_usuario`);

--

-- Indices de la tabla `migrations`

--

ALTER TABLE `migrations`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `ordentrabajo`

--

ALTER TABLE `ordentrabajo`

ADD PRIMARY KEY (`id`),

ADD KEY `ordentrabajo\_id\_tecnico\_foreign` (`id\_tecnico`),

ADD KEY `ordentrabajo\_id\_interaccion\_foreign` (`id\_interaccion`);

--

-- Indices de la tabla `password\_resets`

--

ALTER TABLE `password\_resets`

ADD KEY `password\_resets\_email\_index` (`email`);

--

-- Indices de la tabla `personal\_access\_tokens`

--

ALTER TABLE `personal\_access\_tokens`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `personal\_access\_tokens\_token\_unique` (`token`),

ADD KEY `personal\_access\_tokens\_tokenable\_type\_tokenable\_id\_index` (`tokenable\_type`,`tokenable\_id`);

--

-- Indices de la tabla `planinternets`

--

ALTER TABLE `planinternets`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `plan\_llamadas`

--

ALTER TABLE `plan\_llamadas`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `plan\_tv\_cables`

--

ALTER TABLE `plan\_tv\_cables`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `servicio\_contratado`

--

ALTER TABLE `servicio\_contratado`

ADD PRIMARY KEY (`id`),

ADD KEY `servicio\_contratado\_id\_plan\_internet\_foreign` (`id\_plan\_internet`),

ADD KEY `servicio\_contratado\_id\_plan\_tvcable\_foreign` (`id\_plan\_tvcable`),

ADD KEY `servicio\_contratado\_id\_plan\_llamada\_foreign` (`id\_plan\_llamada`),

ADD KEY `servicio\_contratado\_id\_combo\_promo\_foreign` (`id\_combo\_promo`);

--

-- Indices de la tabla `tecnicos`

--

ALTER TABLE `tecnicos`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `tipo\_servicios\_tecnicos`

--

ALTER TABLE `tipo\_servicios\_tecnicos`

ADD PRIMARY KEY (`id`);

--

-- Indices de la tabla `users`

--

ALTER TABLE `users`

ADD PRIMARY KEY (`id`),

ADD UNIQUE KEY `users\_cedula\_unique` (`cedula`),

ADD UNIQUE KEY `users\_username\_unique` (`username`),

ADD UNIQUE KEY `users\_email\_unique` (`email`);

--

-- AUTO\_INCREMENT de las tablas volcadas

--

--

-- AUTO\_INCREMENT de la tabla `cancelacion\_ordentrabajo`

--

ALTER TABLE `cancelacion\_ordentrabajo`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `combos\_promo`

--

ALTER TABLE `combos\_promo`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `contrato`

--

ALTER TABLE `contrato`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `factura`

--

ALTER TABLE `factura`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `failed\_jobs`

--

ALTER TABLE `failed\_jobs`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `interaccion`

--

ALTER TABLE `interaccion`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `migrations`

--

ALTER TABLE `migrations`

MODIFY `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=18;

--

-- AUTO\_INCREMENT de la tabla `ordentrabajo`

--

ALTER TABLE `ordentrabajo`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `personal\_access\_tokens`

--

ALTER TABLE `personal\_access\_tokens`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `planinternets`

--

ALTER TABLE `planinternets`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT de la tabla `plan\_llamadas`

--

ALTER TABLE `plan\_llamadas`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT de la tabla `plan\_tv\_cables`

--

ALTER TABLE `plan\_tv\_cables`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT de la tabla `servicio\_contratado`

--

ALTER TABLE `servicio\_contratado`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT de la tabla `tecnicos`

--

ALTER TABLE `tecnicos`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

--

-- AUTO\_INCREMENT de la tabla `tipo\_servicios\_tecnicos`

--

ALTER TABLE `tipo\_servicios\_tecnicos`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- AUTO\_INCREMENT de la tabla `users`

--

ALTER TABLE `users`

MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=3;

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `cancelacion\_ordentrabajo`

--

ALTER TABLE `cancelacion\_ordentrabajo`

ADD CONSTRAINT `cancelacion\_ordentrabajo\_id\_orden\_trabajo\_foreign` FOREIGN KEY (`id\_orden\_trabajo`) REFERENCES `ordentrabajo` (`id`);

--

-- Filtros para la tabla `combos\_promo`

--

ALTER TABLE `combos\_promo`

ADD CONSTRAINT `combos\_promo\_id\_plan\_internet\_foreign` FOREIGN KEY (`id\_plan\_internet`) REFERENCES `planinternets` (`id`),

ADD CONSTRAINT `combos\_promo\_id\_plan\_llamada\_foreign` FOREIGN KEY (`id\_plan\_llamada`) REFERENCES `plan\_llamadas` (`id`),

ADD CONSTRAINT `combos\_promo\_id\_plan\_tvcable\_foreign` FOREIGN KEY (`id\_plan\_tvcable`) REFERENCES `plan\_tv\_cables` (`id`);

--

-- Filtros para la tabla `contrato`

--

ALTER TABLE `contrato`

ADD CONSTRAINT `contrato\_id\_servicio\_contratado\_foreign` FOREIGN KEY (`id\_servicio\_contratado`) REFERENCES `servicio\_contratado` (`id`),

ADD CONSTRAINT `contrato\_id\_usuario\_foreign` FOREIGN KEY (`id\_usuario`) REFERENCES `users` (`id`);

--

-- Filtros para la tabla `factura`

--

ALTER TABLE `factura`

ADD CONSTRAINT `factura\_id\_servicio\_contratado\_foreign` FOREIGN KEY (`id\_servicio\_contratado`) REFERENCES `servicio\_contratado` (`id`);

--

-- Filtros para la tabla `interaccion`

--

ALTER TABLE `interaccion`

ADD CONSTRAINT `interaccion\_id\_tipo\_servicio\_tecnico\_foreign` FOREIGN KEY (`id\_tipo\_servicio\_tecnico`) REFERENCES `tipo\_servicios\_tecnicos` (`id`),

ADD CONSTRAINT `interaccion\_id\_usuario\_foreign` FOREIGN KEY (`id\_usuario`) REFERENCES `users` (`id`);

--

-- Filtros para la tabla `ordentrabajo`

--

ALTER TABLE `ordentrabajo`

ADD CONSTRAINT `ordentrabajo\_id\_interaccion\_foreign` FOREIGN KEY (`id\_interaccion`) REFERENCES `interaccion` (`id`),

ADD CONSTRAINT `ordentrabajo\_id\_tecnico\_foreign` FOREIGN KEY (`id\_tecnico`) REFERENCES `tecnicos` (`id`);

--

-- Filtros para la tabla `servicio\_contratado`

--

ALTER TABLE `servicio\_contratado`

ADD CONSTRAINT `servicio\_contratado\_id\_combo\_promo\_foreign` FOREIGN KEY (`id\_combo\_promo`) REFERENCES `combos\_promo` (`id`),

ADD CONSTRAINT `servicio\_contratado\_id\_plan\_internet\_foreign` FOREIGN KEY (`id\_plan\_internet`) REFERENCES `planinternets` (`id`),

ADD CONSTRAINT `servicio\_contratado\_id\_plan\_llamada\_foreign` FOREIGN KEY (`id\_plan\_llamada`) REFERENCES `plan\_llamadas` (`id`),

ADD CONSTRAINT `servicio\_contratado\_id\_plan\_tvcable\_foreign` FOREIGN KEY (`id\_plan\_tvcable`) REFERENCES `plan\_tv\_cables` (`id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

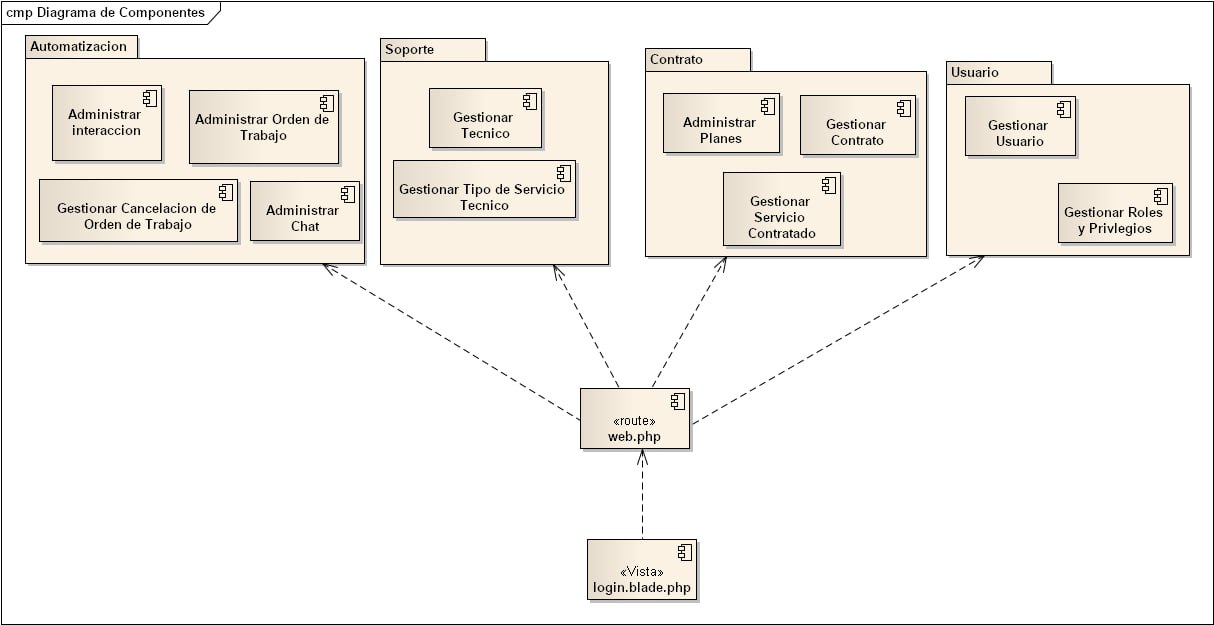
/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

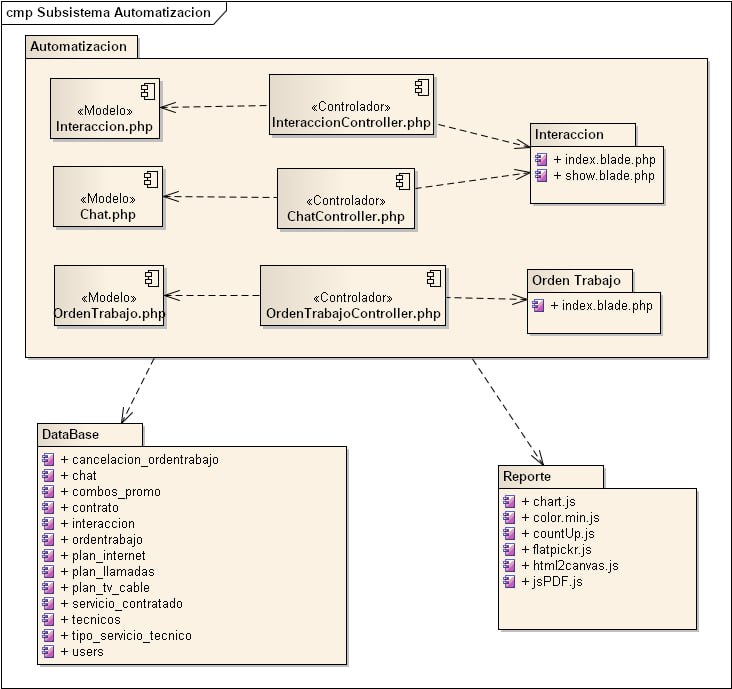
# CAPÍTULO 8

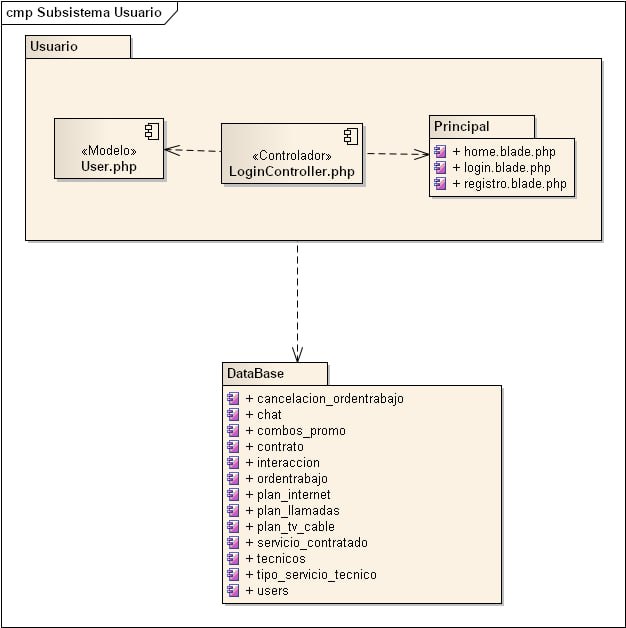
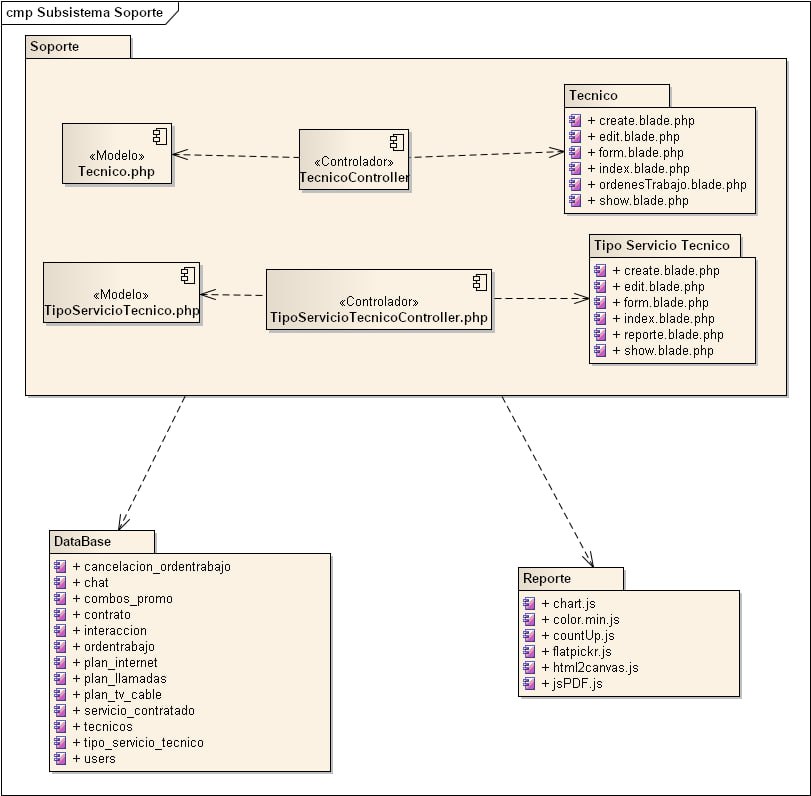
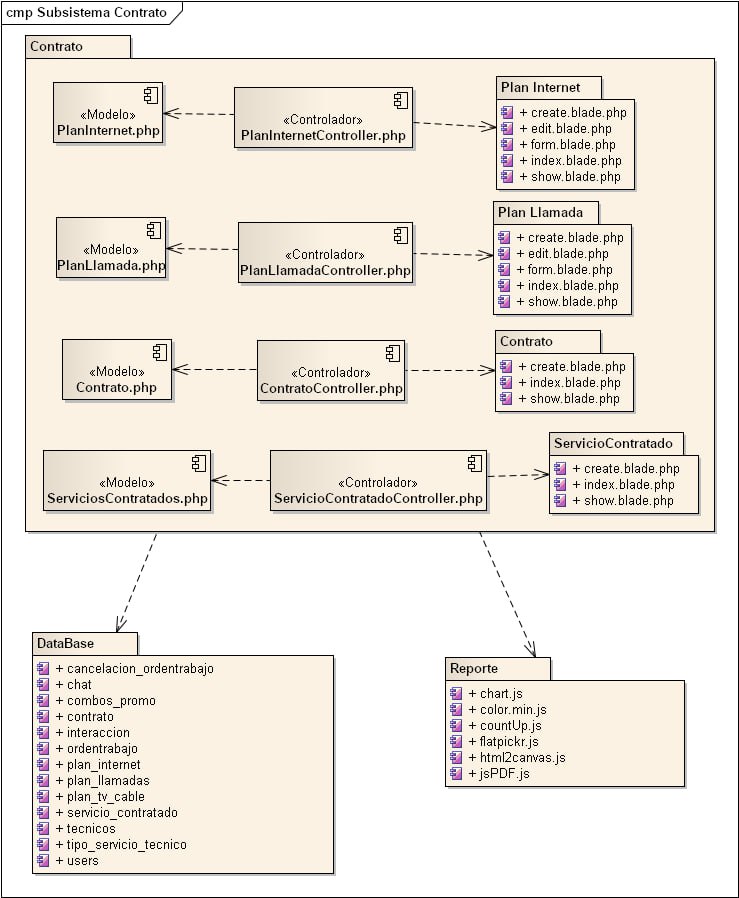
1. IMPLEMENTACIÓN DE ARQUITECTURA

## Diagrama de componente



## IMPLEMENTACIÓN DE ARQUITECTURA DE SUBSISTEMA





## HERRAMIENTAS DE DESARROLLO

### SISTEMA OPERATIVO

#### Pop 22.04 jammy

Pop-OS es un sistema operativo para profesionales de STEM y creativos que utilizar su computadora como una herramienta para descubrir y crear. Libera tu Potencial en un software seguro y confiable de código abierto. Basado en tu curiosidad excepcional.

**Racionaliza tu flujo de trabajo**

Pop-OS está diseñado para una navegación rápida, fácil organización del espacio de trabajo, y fluido, cómodo flujo de trabajo. Su sistema operativo debe fomentar descubrimiento, no obstruirlo.

**Herramientas de desarrollo funcionan sin problemas**

Desarrollar software fuera de la caja con una amplia gama de bibliotecas y herramientas en tu eliminación. Pop-OS utiliza la gestión de paquetes APT y Flatpak, lo que significa que fácil instalar, quitar y actualizar todo el software en el sistema operativo.

**Aprendizaje profundo**

Código de la vida en la maquinaria del futuro. Perfible su modelo para predicen un camino de huracanes, y use Tensorman para mantenerse organizado a lo largo del camino.

**Ingeniería**

Datos seguros dentro de una fortaleza impenetrable. Cierra tierra, luego vete. en un viaje para viajes interestelares. Tus posibilidades y Potencial son infinitas.

#### Windows 10

Windows 10 es la última versión del sistema operativo desarrollado por Microsoft. Lanzado en julio de 2015, Windows 10 ofrece una experiencia de computación moderna y versátil para usuarios de PC y dispositivos móviles. Presenta una interfaz intuitiva y familiar, con el menú de inicio tradicional combinado con la pantalla de inicio de Windows 8, lo que permite una navegación fácil y rápida.

Windows 10 también se destaca por su compatibilidad con una amplia gama de hardware y software, lo que lo convierte en una opción popular tanto para usuarios domésticos como para entornos empresariales. Ofrece características innovadoras, como el asistente virtual Cortana, la integración con servicios en la nube a través de OneDrive y una tienda de aplicaciones con una amplia selección de software disponible. Además, Windows 10 continúa recibiendo actualizaciones regulares de funciones y seguridad, lo que garantiza un sistema operativo actualizado y protegido. En resumen, Windows 10 es un sistema operativo moderno y versátil que brinda una experiencia de usuario mejorada y una mayor productividad en una variedad de dispositivos.

### LENGUAJES DE PROGRAMACIÓN

#### Node JS

Node.js es un entorno de tiempo de ejecución de código abierto basado en el motor de JavaScript V8 de Google Chrome. Permite ejecutar código JavaScript en el lado del servidor, lo que lo convierte en una opción popular para el desarrollo de aplicaciones web y servidores. Node.js utiliza un modelo de E/S sin bloqueo y orientado a eventos, lo que lo hace altamente eficiente y escalable en entornos con alta concurrencia.

Una de las características más destacadas de Node.js es su capacidad para trabajar con conexiones de red de manera eficiente, lo que lo convierte en una opción ideal para aplicaciones en tiempo real, como chats, juegos en línea y aplicaciones de transmisión en vivo. Además, Node.js cuenta con un ecosistema de paquetes y módulos muy amplio, gracias a npm (Node Package Manager), que permite a los desarrolladores acceder a una gran cantidad de bibliotecas y herramientas para agilizar el desarrollo de aplicaciones.

Otra ventaja de Node.js es su capacidad para compartir código entre el lado del cliente y el servidor, lo que simplifica el proceso de desarrollo y mejora la coherencia en la lógica de la aplicación. Además, al estar basado en JavaScript, un lenguaje de programación popular y ampliamente adoptado, los desarrolladores que ya están familiarizados con JavaScript pueden aprovechar sus habilidades existentes para construir aplicaciones en Node.js. En resumen, Node.js es un entorno de tiempo de ejecución flexible y de alto rendimiento que permite a los desarrolladores construir aplicaciones escalables, en tiempo real y basadas en eventos utilizando JavaScript tanto en el lado del cliente como en el servidor.

Express es un framework web rápido y minimalista para Node.js que simplifica el proceso de creación de aplicaciones web. Se destaca por su enfoque en la simplicidad y la flexibilidad, lo que lo convierte en una opción popular para desarrolladores de Node.js de todos los niveles de experiencia. Express proporciona una serie de funciones y características que facilitan la creación de servidores web y el manejo de rutas, solicitudes y respuestas HTTP.

Una de las principales fortalezas de Express es su estructura modular y su arquitectura de middleware. Permite a los desarrolladores crear aplicaciones web de manera modular, agregando y configurando fácilmente el middleware necesario para manejar diferentes aspectos de la aplicación, como el enrutamiento, la autenticación, el manejo de errores, entre otros. Esto brinda flexibilidad y extensibilidad, ya que se pueden agregar o eliminar middleware según las necesidades específicas del proyecto.

Express también se destaca por su simplicidad y su enfoque en la escritura de código limpio y conciso. Proporciona una API intuitiva y fácil de usar para definir rutas y manejar las solicitudes y respuestas HTTP. Además, cuenta con una gran cantidad de recursos y complementos de la comunidad, lo que facilita la ampliación de las funcionalidades de la aplicación.

En resumen, Express es un framework web minimalista y flexible para Node.js. Su estructura modular y su arquitectura de middleware permiten a los desarrolladores crear aplicaciones web de manera rápida y eficiente. Su enfoque en la simplicidad y la escritura de código limpio lo convierte en una opción popular para el desarrollo de aplicaciones web en Node.js.

#### PHP

PHP es un lenguaje de programación de código abierto ampliamente utilizado en el desarrollo web. Es especialmente adecuado para crear aplicaciones web dinámicas y sitios web interactivos. PHP se ejecuta en el lado del servidor, lo que significa que el código PHP se procesa en el servidor antes de enviar la respuesta al navegador del cliente. Una de las principales fortalezas de PHP es su capacidad para interactuar con bases de datos y generar contenido dinámico en función de las solicitudes del usuario.

En resumen, PHP es un lenguaje de programación ampliamente utilizado en el desarrollo web, mientras que Flight es un framework ligero y minimalista para PHP. Flight se enfoca en la simplicidad y el modularidad, proporcionando solo las características esenciales para el desarrollo de aplicaciones web. Es ideal para proyectos pequeños o cuando se busca una solución ágil y liviana que permita un desarrollo rápido y eficiente.

#### CSS

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para definir la presentación y el aspecto visual de una página web. Junto con HTML, forma parte de los pilares fundamentales del diseño web. CSS permite controlar la apariencia de los elementos HTML, como el tamaño, color, tipo de fuente, espaciado, bordes y fondos, entre otros aspectos.

Una de las principales ventajas de CSS es su capacidad para separar el contenido de la presentación. Esto significa que el contenido de una página web se puede estructurar en HTML y luego se puede aplicar el estilo deseado utilizando CSS. Esta separación permite una mayor flexibilidad y mantenibilidad en el diseño, ya que los cambios en el estilo pueden aplicarse de manera centralizada a través de una hoja de estilos CSS, sin necesidad de modificar el código HTML.

CSS ofrece una amplia gama de selectores y propiedades que permiten un control detallado sobre la apariencia de los elementos de una página web. Además, con la introducción de CSS3, se han agregado numerosas características avanzadas, como animaciones, transformaciones, sombras y gradientes, que brindan aún más posibilidades creativas en el diseño web. En resumen, CSS es una herramienta esencial para dar vida y estilo a las páginas web, brindando la capacidad de crear diseños atractivos y visualmente agradables.

#### HTML

HTML (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para crear la estructura y el contenido de las páginas web. Es la base fundamental de la mayoría de los sitios web en Internet. HTML utiliza etiquetas y elementos para definir la semántica y la organización del contenido, como encabezados, párrafos, enlaces, imágenes, tablas y formularios.

Con HTML, se puede estructurar el contenido de una página web de manera jerárquica, permitiendo definir encabezados y suben cabezados para organizar la información de manera lógica. Además, se pueden crear enlaces para conectar diferentes páginas y recursos, lo que facilita la navegación en el sitio web. HTML también permite la inclusión de elementos multimedia, como imágenes y videos, para enriquecer la experiencia visual de los usuarios.

Una de las características más importantes de HTML es su naturaleza de lenguaje de marcado, lo que significa que se centra en describir la estructura y el significado del contenido en lugar de su apariencia visual. Esto permite una mayor separación entre el contenido y la presentación, lo que facilita la adaptabilidad y el mantenimiento de los sitios web. En conjunto con CSS y JavaScript, HTML proporciona las bases para la creación de sitios web interactivos y dinámicos.

### BASE DE DATOS

#### MySQL

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado en el desarrollo de aplicaciones web y software en general. Se destaca por su confiabilidad, rendimiento y escalabilidad, lo que lo convierte en una opción popular tanto para proyectos pequeños como para aplicaciones empresariales de gran envergadura.

MySQL permite almacenar, administrar y recuperar datos de manera eficiente. Utiliza el lenguaje de consulta estructurado (SQL) para interactuar con la base de datos, lo que facilita la creación, modificación y eliminación de tablas, así como la manipulación de los datos almacenados en ellas. Además, MySQL ofrece características avanzadas, como transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), que garantizan la integridad y la coherencia de los datos.

Con su arquitectura cliente-servidor, MySQL permite la conexión simultánea de múltiples clientes a la base de datos, lo que facilita el acceso y el procesamiento de datos por parte de distintas aplicaciones. Además, ofrece opciones de replicación y clustering para mejorar la disponibilidad y la capacidad de respuesta en entornos de alta demanda. En resumen, MySQL es una poderosa herramienta de gestión de bases de datos que brinda flexibilidad, rendimiento y confiabilidad para el desarrollo de aplicaciones basadas en datos.

### HERRAMIENTAS CASE

#### Draw.io

Draw.io es una herramienta de creación de diagramas en línea que permite a los usuarios diseñar y visualizar de manera intuitiva diversos tipos de diagramas, como diagramas de flujo, organigramas, diagramas de red, diagramas UML y muchos otros. Es una aplicación de diagramación versátil y fácil de usar que no requiere instalación, ya que se ejecuta directamente en el navegador web. Con una amplia gama de formas y elementos gráficos predefinidos, los usuarios pueden arrastrar y soltar componentes para construir sus diagramas de forma rápida y sencilla. Además, draw.io ofrece una interfaz intuitiva y una amplia variedad de herramientas de edición para personalizar los diagramas, agregar texto, aplicar colores y estilos, y conectar elementos con líneas y flechas. La capacidad de colaboración en tiempo real permite a varios usuarios trabajar simultáneamente en un mismo diagrama, lo que facilita la colaboración en proyectos y presentaciones. draw.io es una herramienta muy útil para profesionales, estudiantes y cualquier persona que necesite crear y compartir diagramas de manera efectiva y profesional.

#### Git

Git es un sistema de control de versiones ampliamente utilizado que permite a los desarrolladores gestionar eficientemente el seguimiento de los cambios en el código fuente de un proyecto. Proporciona un entorno distribuido en el que múltiples personas pueden colaborar en un mismo proyecto, facilitando la gestión de ramas de desarrollo, la fusión de cambios y el seguimiento de versiones. Git ofrece una serie de características poderosas, como la capacidad de deshacer cambios, revertir a versiones anteriores, realizar ramificaciones y fusiones sin problemas, y realizar seguimiento de conflictos en la fusión de código. Además, su arquitectura descentralizada permite a los desarrolladores trabajar de forma independiente en sus propios repositorios locales y luego sincronizar los cambios con un repositorio centralizado. Esto facilita el desarrollo en equipos distribuidos y la colaboración en proyectos de software de cualquier tamaño. Git se ha convertido en una herramienta esencial en el desarrollo de software moderno, brindando a los equipos un control de versiones eficiente, una historia de cambios clara y una forma segura de gestionar el código fuente.

#### Chat GPT-3.5

Chat GPT-3.5 es un modelo de lenguaje desarrollado por OpenAI que utiliza la arquitectura GPT-3 para generar respuestas en lenguaje natural a partir de una entrada de texto. Es capaz de comprender y generar contenido coherente y contextualmente relevante en una amplia variedad de temas. Chat GPT-3.5 ha sido entrenado en una gran cantidad de datos y posee un conocimiento extenso en áreas como ciencia, tecnología, cultura y más. Su capacidad para generar respuestas humanas y su flexibilidad para adaptarse a diferentes situaciones hacen de Chat GPT-3.5 una herramienta poderosa para la generación de texto y la interacción con usuarios en conversaciones virtuales y asistencia en tareas de escritura y comprensión de lenguaje natural.

#### Postman

Postman es una herramienta de colaboración y desarrollo de API que permite a los desarrolladores crear, probar, documentar y compartir fácilmente APIs. Con una interfaz intuitiva y amigable, Postman simplifica el proceso de enviar solicitudes HTTP y visualizar las respuestas, lo que facilita el desarrollo y la depuración de APIs.

Además de ser una plataforma para probar y depurar APIs, Postman ofrece características avanzadas como la posibilidad de crear entornos de prueba personalizados, automatizar flujos de trabajo, generar documentación detallada de API y colaborar con equipos en proyectos de desarrollo. Con su amplia gama de funcionalidades, Postman se ha convertido en una herramienta indispensable para los desarrolladores y equipos de desarrollo de software, mejorando la eficiencia y la calidad en el proceso de desarrollo de APIs.

# CAPÍTULO 9

1. Flujo de trabajo: Prueba

## Plan de prueba

### Caja Negra

Para realizarlas las pruebas al Asistente Virtual Avanzado para reducir el tiempo de espera de los clientes, se tomarán los casos de uso principales.

Aplicando el MÉTODO DE LA CAJA NEGRA.

Se realizarán los siguientes pasos:

• Prototipo de la interfaz de prueba.

• Identificar las entradas.

• Resultados.

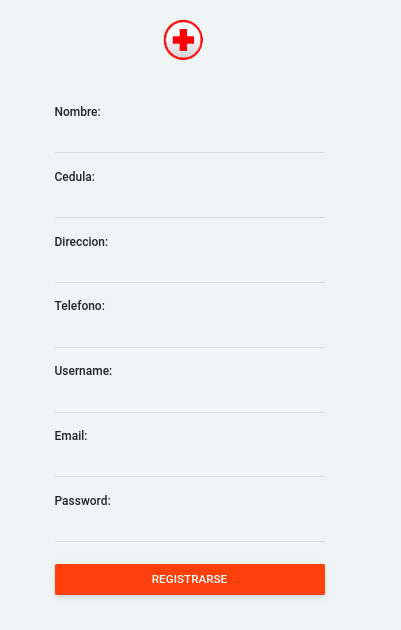
• Condiciones.

• Procedimientos de Prueba.

A continuación, se describen los casos de uso sobre los que se realizara los casos de prueba.

#### Metodología de la Caja Negra

###### Gestionar Usuario



1. **Entrada**

Habilitar la interfaz de registrar usuario

1. **Resultado**

Se registra un usuario

1. **Condición**

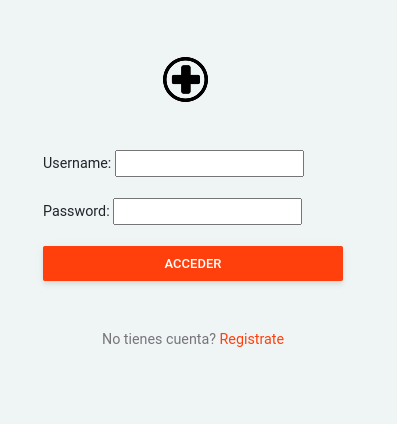
Debe tener los permisos necesarios para crear un usuario según su tipo

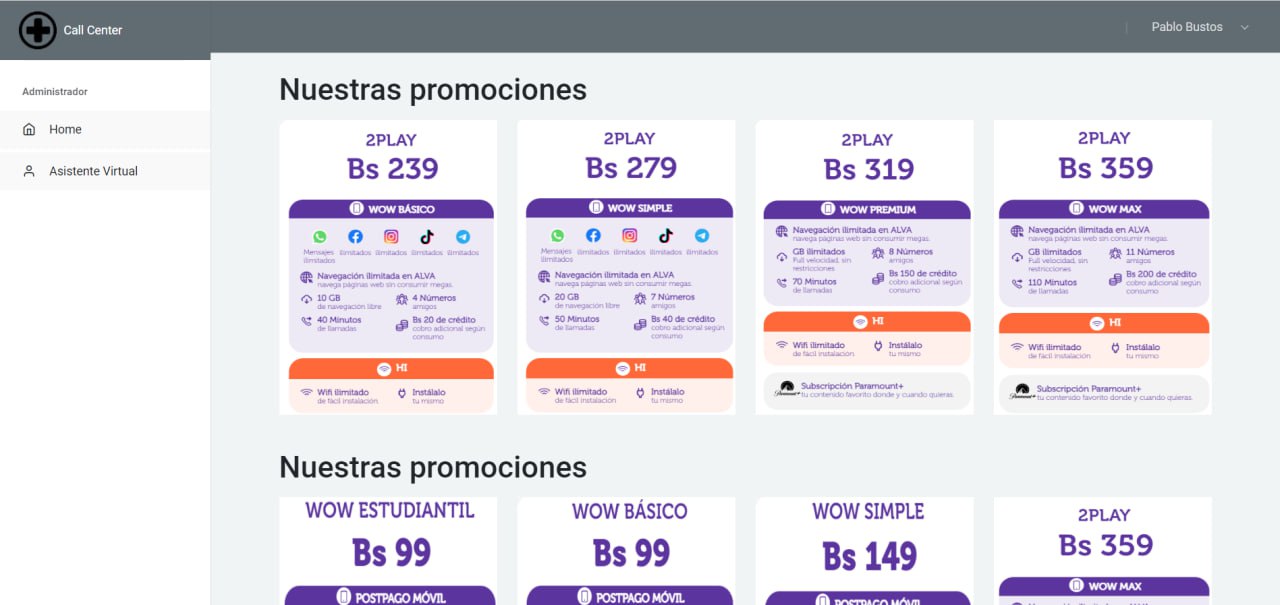
Debe haber llenado el formulario

1. **Procedimiento de prueba**

Debe presionar el botón Registrarse

###### Login





1. **Entrada**

Debe habilitar la interfaz de Login

1. **Resultado**

Si las credenciales son correctas entrara a la interfaz inicial

Si las credenciales son incorrectas se lo volverá a redireccionar al login

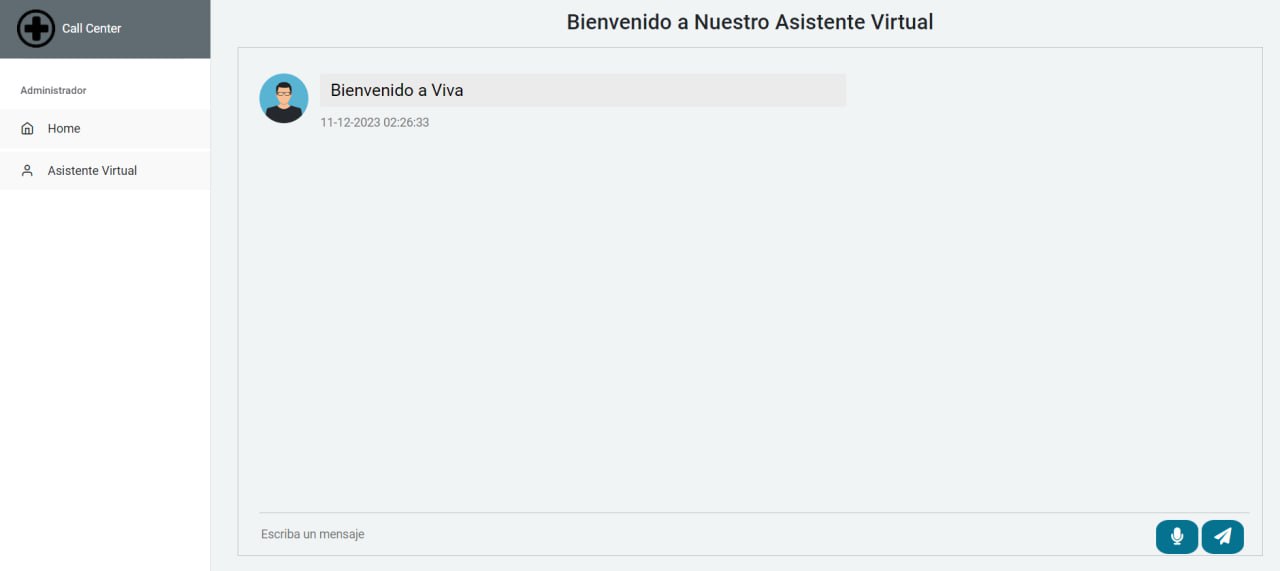
1. **Condición**

Debe haberse registrado previamente al usuario

1. **Procedimiento de prueba**

Presionar el botón Acceder

###### Administrar interacción



1. **Entrada**

Debe habilitar la interfaz de interacción

1. **Resultado**

De primera instancia tendrá un saludo de bienvenida

Ahora cuando el usuario envié un mensaje este tendrá una respuesta

1. **Condición**

Debe haberse registrado previamente al usuario

1. **Procedimiento de prueba**

Presionar el botón de enviar

### Caja Blanca

1. **Configuración Inicial:**

Integra el código en un sistema de control de versiones.

Asegúrate de tener SonarQube y sonar-scanner configurados.

1. **Análisis de Código:**

Ejecuta sonar-scanner para analizar el código.

1. **Revisión en SonarQube:**

Accede al panel de SonarQube para revisar resultados.

Aborda problemas críticos y áreas de mejora identificadas.

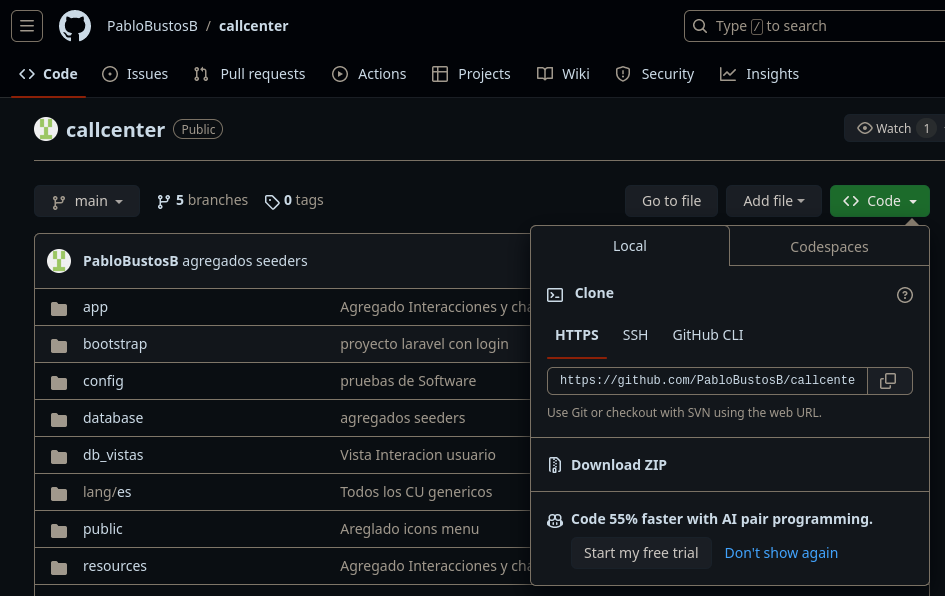
1. **Iteración Continua:**

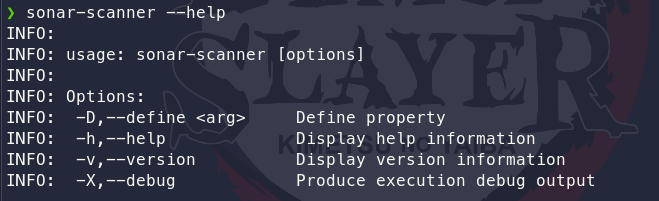
Repite el análisis y la revisión con cada cambio en el código.

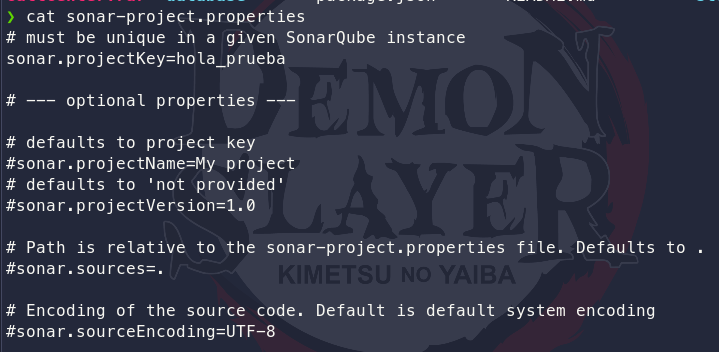
Corrige problemas y mejora la calidad de manera iterativa.

Estos pasos te proporcionan una metodología básica para realizar pruebas de caja blanca utilizando SonarQube y sonar-scanner. Puedes repetir estos pasos de manera continua durante el desarrollo para mantener la calidad del código.

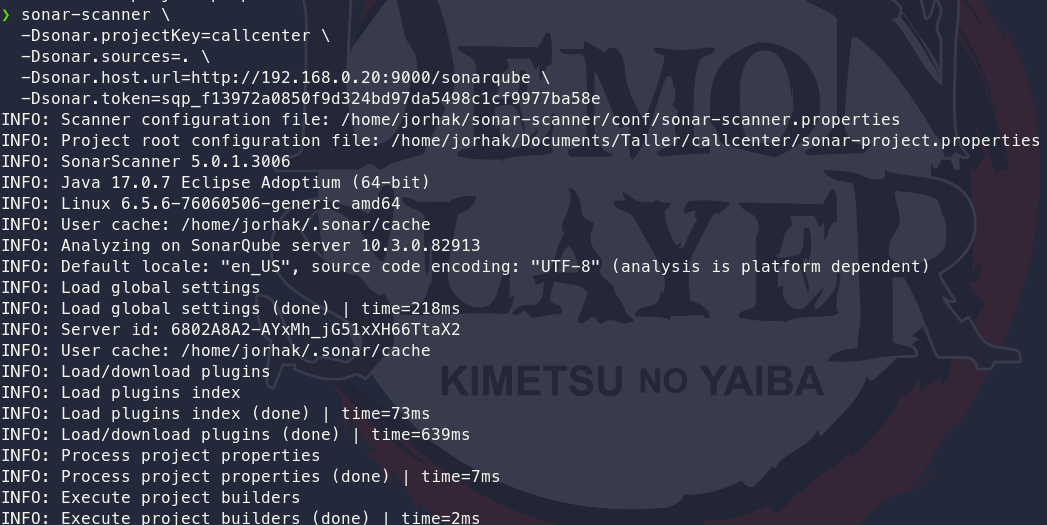
#### Configuración inicial





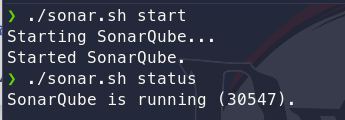


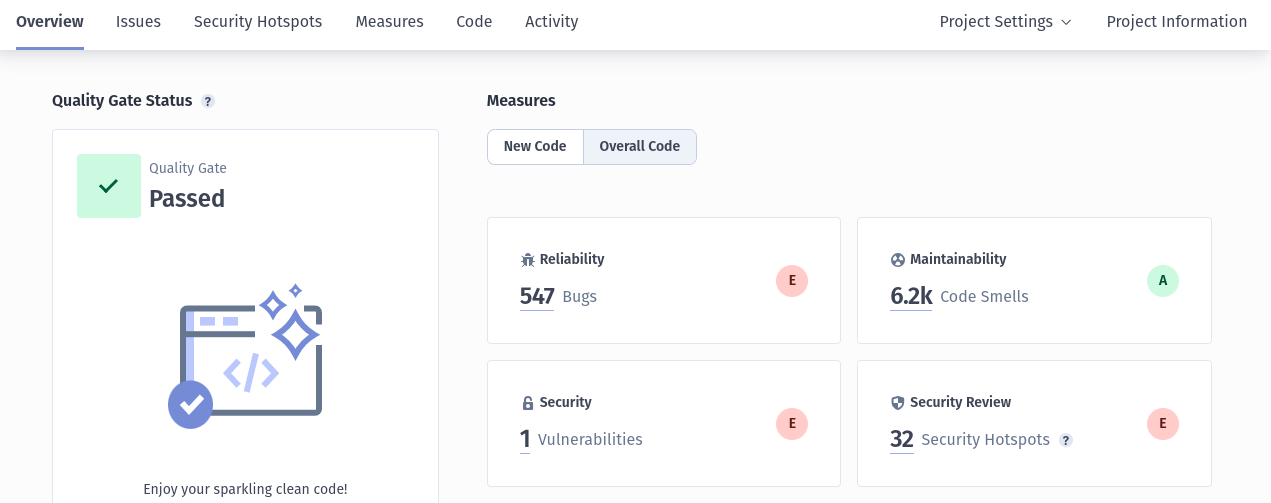
#### Análisis del código

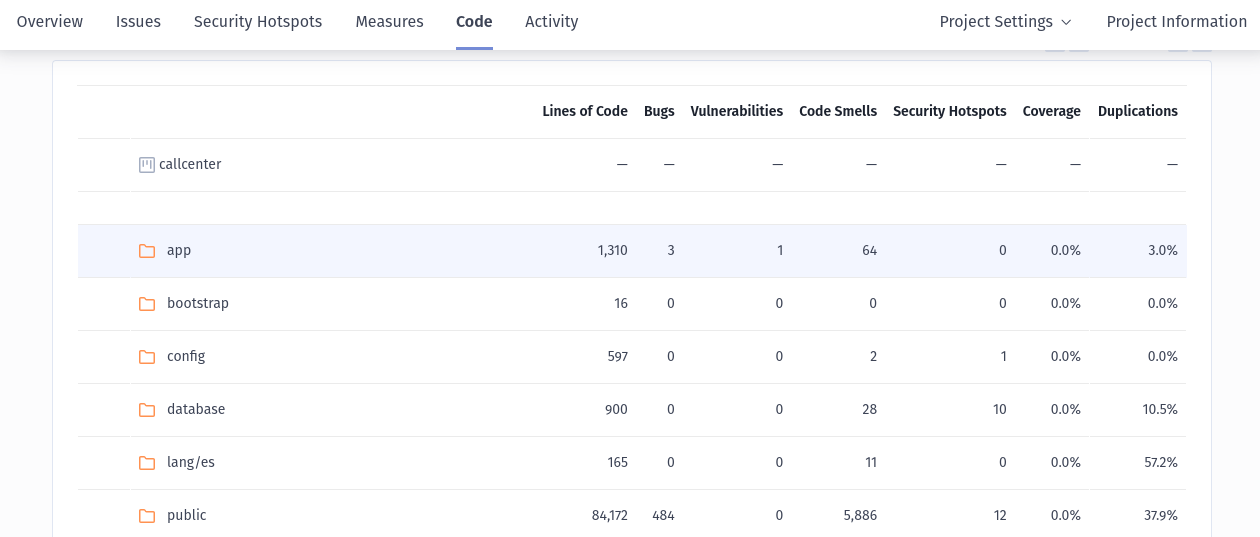


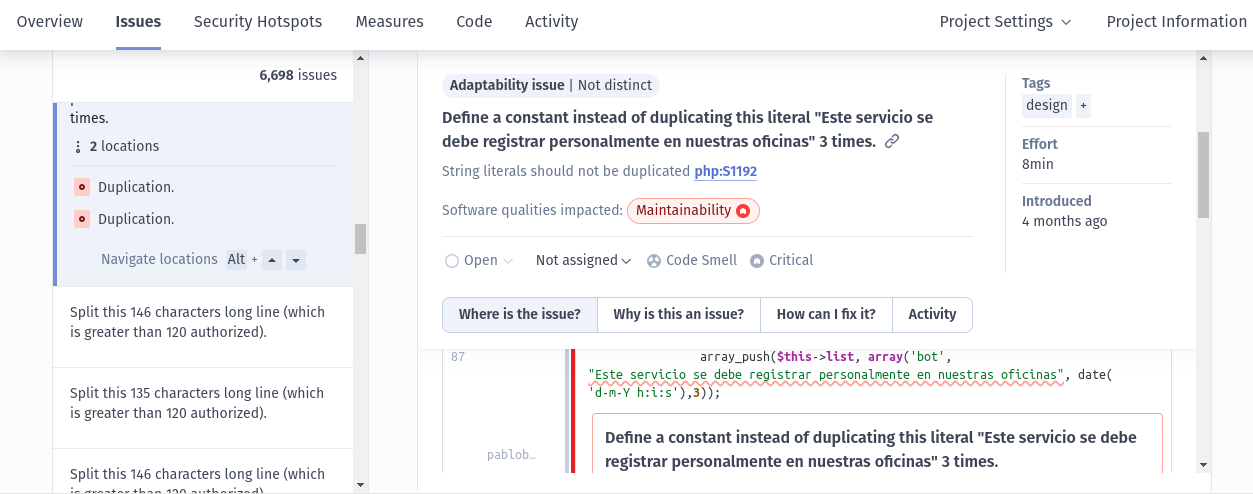
#### Revisión en SonarQube

**Iniciar servicios**

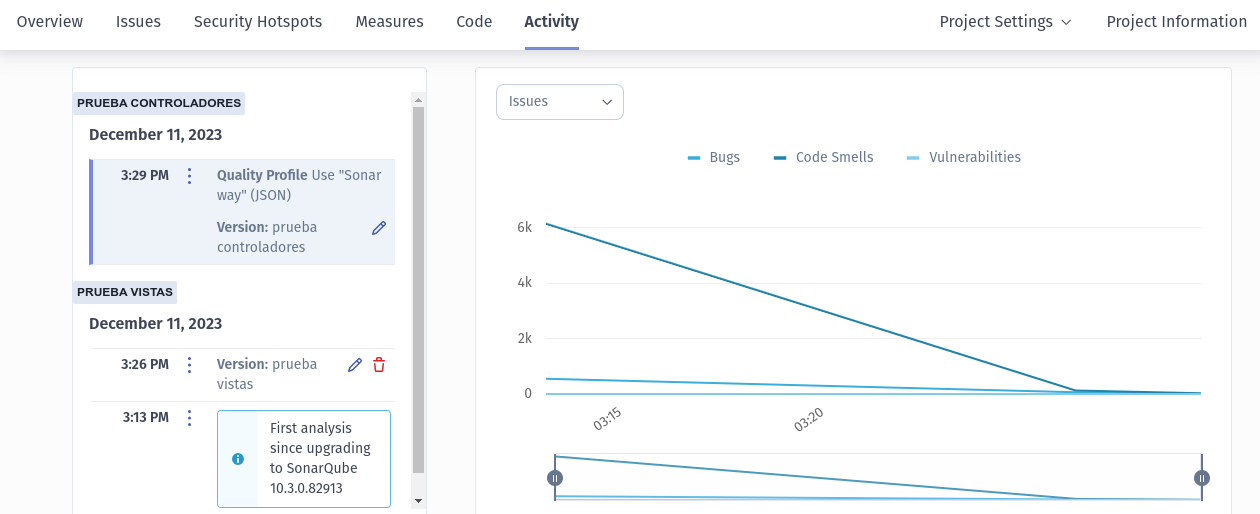








#### Iteración continua



### TDD (Desarrollo dirigido por pruebas)

TDD (Desarrollo Dirigido por Pruebas, por sus siglas en inglés Test-Driven Development) es una metodología de desarrollo de software en la que las pruebas automáticas son escritas antes de que el código de producción sea implementado. En el contexto de PHP, TDD implica seguir un ciclo de desarrollo específico conocido como el ciclo de "Rojo, Verde, Refactor".

Aquí tienes una descripción del ciclo TDD en PHP:

1. **Rojo (Red):**

Antes de escribir el código de producción, escribes un caso de prueba que describe la funcionalidad deseada.

Este caso de prueba inicialmente falla (da un resultado "rojo") porque la funcionalidad aún no está implementada.

1. **Verde (Green):**

Ahora, implementas el código mínimo necesario para que el caso de prueba pase.

La idea es escribir solo lo suficiente para que el caso de prueba pase, sin preocuparse por la calidad o la optimización.

1. **Refactorización (Refactor):**

Después de que el caso de prueba ha pasado satisfactoriamente, puedes refactorizar el código para mejorarlo sin cambiar su comportamiento.

Las pruebas automatizadas aseguran que cualquier cambio no intencional en el comportamiento sea rápidamente detectado.



# CAPITULO 10

1. COSTOS

## POSIBLES COSTOS

Para desarrollar el Asistente Virtual para reducir el tiempo de espera de los clientes de la empresa VIVA, se necesitará lo siguiente:

### Hardware

El hardware requerido para que el computador soporte el Asistente Virtual para reducir el tiempo de espera de los clientes de la empresa VIVA, oscila bajo los siguientes costos unitarios:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Versión** | **Capacidad** | **Costo ($)** | |
| **Procesador** | Intel i5-9150 | 3.5 GHz | | 113,00 $ |
| **Memoria RAM** | KingstoneDDR3 | 4GB | | 37,00 $ |
| **Tarjeta de video** | GeForce GTX 650 Ti | 2GB | | 120,00 $ |
| **Disco duro** | Seagate Desktop HDD | 1TB | | 57,00 $ |
| **Placa base** | ASRock 960GM-VGS3 FX | - | | 42,00 $ |
| **Case combo** | Delux DLC-MU305 | - | | 50,00 $ |
|  | | | | 419,00 $ |

### Software

Las licencias requeridas para ciertas plataformas de funcionamiento continuo irrestricto, y de seguridad oscilan bajo los siguientes costos:

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Versión** | **Costo ($)** |
| **Sistema operativo** | Windows 10 Pro 64 Bits | 319,99 $ |
| **Gestor de base de datos** | MySQL | 00 $ |
| **Antivirus** | ESET Smart Internet Security 8 | 79,99 $ |
| **Antivirus para servidor** | ESET END Point Security | 39,99 $ |
| **Lector de documentos** | ADOBE Reader PDF xi | Free |
|  | | 2439,97 $ |

### Recursos humanos

El recurso humano es una parte fundamental para el desarrollo del Sistema Software y su costo de trabajo se especifica en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **Ocupación** | **Cantidad** | **Costo/Mes - C/U** |
| **Analista** | 2 | 300,00 $ |
| **Diseñador** | 2 | 400,00 $ |
| **Desarrollador** | 2 | 300,00 $ |

* 1. Bibliografía

Bagnato, J. (2022). *Principales Algoritmos usados en machine learning, Aprende Machine Learning.* Recuperado el 01 de Diciembre de 2023, de https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/

Bagnato, J. (2022). *Procesamiento del Lenguaje Natural (NLP), Aprende Machine Learning.* Recuperado el 01 de Diciembre de 2023, de https://www.aprendemachinelearning.com/procesamiento-del-lenguaje-natural-nlp/

Booch G., R. J. (1999). *Booch G., Rumbaugh J., Jacobson I.,.* Madrid: Pearson Educación S.A.

Camps, G. (2022). *Procesamiento del Lenguaje Natural (NPL) con python, Damavis Blog - Data - Machine Learning - Visualization.* Recuperado el 01 de Diciembre de 2023, de https://blog.damavis.com/procesamiento-del-lenguaje-natural-nlp-con-python/

Germán, R. (Septiembre de 2019). *Cómo construir un asistente de voz con herramientas de código abierto como Rasa y Mozilla.* Recuperado el 26 de Septiembre de 2023, de https://planetachatbot.com/tutorial-como-construir-asistente-voz-con-herramientas-de-codigo-abierto-rasa-y-mozilla/

Germán, R. (2021). *Técnicas de Análisis de Sentimientos Aplicadas a la Valoración de Opiniones en el Lenguaje Español - UNLP*. Recuperado el 01 de Noviembre de 2023, de http://sedici.unlp.edu.ar/bitstream/handle/10915/130344/Documento\_completo.pdf

Gonzales, M. (Julio de 2020). *Analítica de Textos en la Detección de Sentimientos Aplicada a un Cali Center Automotriz.* Recuperado el 11 de Noviembre de 2023, de https://tecnocultura.org/index.php/Tecnocultura/article/view/247/268

IBM. (2023). *IBM Cloud Services*. Obtenido de IBM: https://www.ibm.com/mx-es/topics/recurrent-neural-networks

León Albán, J. A. (18 de Marzo de 2021). *Desarrollo de un sistema software inteligente que mejore la gestión de órdenes de los usuarios de la empresa ATV Cable de la ciudad de Latacunga provincia de Cotopaxi. Latacunga: Universidad de las fuerzas armadas.* Recuperado el 25 de Septiembre de 2023, de http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/24372/T-ESPEL-SOF-0040.pdf

Moreno, A. (06 de Noviembre de 2018). *Procesamiento del lenguaje natural ¿Qué es? - IIC, Instituto de Ingeniería del Conocimiento.* Recuperado el 01 de Diciembre de 2023, de https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/

VIVA, S. (Septiembre de 2019). *Organigrama septiembre 2019*. Recuperado el 26 de Septiembre de 2023, de https://es.scribd.com/document/449080652/9-Organigrama-Septiembre-2019-pdf