



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**BITACORA DE ATENCIÓN A CLIENTES DE TELEFONÍA
MÓVIL**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

MARIANA DOLORES MARTINEZ TORRES

DANIELA ALEJANDRA CONTRERAS VALENCIA

ADAN ALBERTO FIGUEROA MIRANDA

DIRECTOR: M.I. JUAN CARLOS ROA BEIZA



Ciudad Universitaria, México D.F., Agosto 2015



DEDICATORIA Y AGRADECIMIENTOS.

*Esta tesis se la dedico a mi hijo **Axel Ali Pérez Contreras** mi guerrero por haber llegado a mi vida y enseñarme a luchar y ser perseverante, por traer los más maravillosos momentos de mi vida por llegar a descubrir muchas cosas de mí, por tu sonrisa que me da paz, tranquilidad ,armonía, felicidad, te amo.*

Quiero agradecer a mi mama Silvia Rita Valencia Guerrero por siempre estar a mi lado por tu apoyo incondicional y consejos por ser mi guía y un gran ejemplo de una gran mujer a mi papa Gerardo Contreras Hernández por enseñarme a que el mejor camino no siempre es el más fácil y por tu apoyo para lograr ser alguien en la vida, a Ali Pérez Morales por recorrer este camino conmigo y creer en mí, a mis hermanas Anabell Conteras Valencia y Laura Angélica Conteras Valencia por su compañía y apoyo, a mi amiga y compañera de carrera Yessica Arredondo Guzmán por todo lo que recorrimos en la facultad de ingeniería para llegar hasta aquí, por sus apoyo y consejos .

Al Maestro en Ingeniería Juan Carlos Roa Beiza por ser mi guía en la elaboración de esta tesis a mis compañeros de tesis Adán Alberto Figueroa Miranda y Mariana Dolores Martínez Torres por trabajar en conjunto para lograr este objetivo, al programa PAT por la oportunidad que me dio para culminar mis estudios y a la UNAM por todo lo que me brindo para llegar hasta aquí.

DANIELA ALEJANDRA CONTRERAS VALENCIA.



DEDICATORIA

A Zoe Michelle Burgos Torres (Mi esposa)

Doy gracias a Dios por conocerte en mi vida, mi compañera de vida. Tu total apoyo en las decisiones que he tomado. Tu amor me ha permitido saber el sentido de mi vida. TE AMO

A Gabriela Miranda Hernández (Mami).

Gracias por darme las armas necesarias para afrontar la vida y por creer en mí. Siempre has estado ahí cuando te he necesitado, y gracias por darme la vida.

A Lourdes Rivas, Martha Hernández, José Miranda y Jorge Figueroa (Mis queridos abuelos).

A mis queridos abuelos, les agradezco todo su amor y cariño cuando más los necesite, por todos sus cuidados y enseñanzas. Muchas Gracias

A mis primos

Todos ustedes son como mis hermanos y hermanas, he tenido el gusto de tenerlos en mi vida. Son la motivación de alcanzar metas en la vida.

A mi familia Miranda Hernández

Por su enorme amor y cariño siempre me cuidaron y protegieron cuando más necesite, nunca me sentí solo gracias a ustedes.

A mi familia Figueroa Rivas

Por todo su cariño y amor les agradezco el confiar en mí.

A mis amigos

Por qué los amigos se tienen en las buenas y en las malas, gracias por su amistad. Por los momentos bueno y malos que hemos pasado.

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.



Agradecimientos

*A la **Universidad Nacional Autónoma de México** y a la **Facultad de Ingeniería** por darme la oportunidad de darme las armas necesarias para defenderme en la vida pertenecer a esta excelente institución académica.*

*Al **PAT** por darme las herramientas necesarias para poder titularme.*

*A mi director de tesis el **M.I. Juan Carlos Roa Beiza**, por su excelente ayuda para poder completar el ciclo de mi carrera universitaria.*

A mis compañeras de tesis, gracias por tener el mismo objetivo en este proyecto, les deseo mucho éxito.

ADAN ALBERTO FIGUROA MIRANDA



Dedicatoria

A mis padres Alicia Torres Estrada y Carlos Martinez Cazares por tener paciencia y creer que algún día este momento llegaría, aquí estoy, culminando una de las etapas más importantes de mi vida. Todos los principios y valores que me enseñaron ayudaron a ser lo que ahora soy.

A mis abuelos + Amador y Leandra por todas sus enseñanzas y todo su cariño, se que están contentos por esta meta que hoy alcanzo, sin ustedes nada de esto hubiera sido posible los quiero y los extraño

A mis dos grandes ejemplos mi hermano Carlos y mi primo Juan, gracias por impulsarme siempre para afrontar cualquier cosa que se me presente en el camino.

A la familia Torres Estrada, la mejor familia que me pudo tocar, gracias a su apoyo y cariño me pude levantar de cada caída para seguir adelante, ustedes son la base de lo que soy ahora, su apoyo y cariño son las armas que me hacían sentir segura de poder vencer cualquier obstáculo. Sin ustedes no estaría aquí.

A mi madrina Dolores, por todas las aventuras que pasamos juntas y todos los obstáculos que me ayudaste a superar, nadie mejor que tú conoce esta historia.

A mis sobrinos Fernanda, Carla, Ana, Emiliano y mi primo Jose Angel por cada alegría que le imprimen a mis días y por impulsarme a querer ser siempre un buen ejemplo para ustedes

A Christian Rojas Ponce, porque impulsaste la idea de cerrar este ciclo, estuviste conmigo y fuiste mi cómplice en el proceso. Gracias por tu apoyo y el de tu familia.



Agradecimientos

*A la **Universidad Nacional Autónoma de México** y a la **Facultad de Ingeniería** por ser mi alma mater y por abrirme las puertas para lograr culminar una carrera universitaria.*

*Al **PAT** por brindarme una oportunidad de desarrollar este proyecto de tesis de una manera óptima y concisa.*

*A mi director de tesis el **M.I. Juan Carlos Roa Beiza**, por guiarnos durante todo el proceso y brindarnos siempre su apoyo.*

A mis compañeros de tesis, gracias por su compromiso para lograr este fin común, fue un placer ir en el mismo barco.

MARIANA DOLORES MARTINEZ TORRES



ÍNDICE TEMÁTICO

CAPITULO 1

ANTECEDENTES

1.1.- INTRODUCCION.....	1
1.2.- CONCEPTOS GENERALES DE LOS TÉRMINOS GENERALES USADOS EN UN CALL CENTER.....	6
1.3.- NORMATIVIDAD DE LOS CALL CENTER.....	11
1.4.- DESCRIPCIÓN BÁSICA DE UN CALL CENTER.....	14
1.5.- ACOTAMIENTO DEL PROBLEMA A RESOLVER.....	18

CAPITULO 2

MARCO TEÓRICO

2.1.- CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES Y METODOLOGÍA MVC.....	21
2.2.- CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.....	26
2.3.- CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL LENGUAJE DE PROGRAMACIÓN PHP.....	29
2.4.- CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL MANEJADOR DE BASE DE DATOS MYSQL.....	33



2.5.- CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL AMBIENTE CLIENTE SERVIDOR.....	36
---	----

CAPITULO 3

PLANTEAMIENTO DEL PROBLEMA

3.1.- ANÁLISIS DEL PROBLEMA.....	42
3.2.- REQUERIMIENTOS GENERALES Y PARTICULARES DE LA APLICACIÓN.....	47
3.3.- ANÁLISIS Y POSICIONAMIENTO DE LOS LENGUAJES PARA EL BACK-END Y FRONT-END	50
3.4.- APLICACIÓN Y SEPARACIÓN DE LOS MÓDULOS SEGÚN LA METODOLOGÍA MVC	64
3.5.- DISEÑO A NIVEL DE BLOQUES DE LA INTERFAZ DEL USUARIO.....	67

CAPITULO 4

ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1.- MODELO ESENCIAL AMBIENTAL DEL SISTEMA	72
4.1.1 DIAGRAMA DE CONTEXTO DEL SISTEMA	72
4.1.2 DIAGRAMA DE ENTIDAD- RELACIÓN DEL SISTEMA.....	74
4.1.3. DIAGRAMA DE CASOS DE USO, CLASES Y DE PROCESOS.....	77
4.1.3.1 DIAGRAMA DE CASOS DE USO	77



4.1.3.2 DIAGRAMA DE CASOS DE CLASES	25
4.1.3.2 DIAGRAMA DE CASOS DE PROCESOS	82
4.1. 4.- DICCIONARIO DE DATOS	88
4.1. 5.- NORMALIZACIÓN DE LA BASE DE DATOS	92
4.2.- CREACIÓN DE LA BASE DE DATOS	101
4.3.- DESARROLLO DEL FRONT-END	109
4.4.- GENERACIÓN DE PRUEBAS Y MANTENIMIENTO	114
4.5.- GENERACIÓN DE REPORTES Y COMPROBANTES	129
CONCLUSIONES.....	132
BIBLIOGRAFÍA.....	134

Capítulo I. Entorno del Problema

1.1 Introducción.

Los **call centers** (centros de llamadas) nacieron de la oportunidad de prestar un servicio inmediato al cliente a través del teléfono. Al principio era principalmente informativo y tenía un carácter de servicio accesorio a la oferta principal del producto. Sin embargo, su utilización se expandió considerablemente, debido principalmente a dos factores:

- Fuerte competencia, que convirtió un servicio de lujo en un canal habitual y necesario de contacto con el cliente.
- Fuerte demanda del cliente particular, que cada vez goza de menos tiempo de ocio y por tanto le da más valor a su tiempo libre.

Hoy en día gran parte de los servicios a los que tenemos acceso utilizan las líneas telefónicas como medio de comunicación recurrente todo con el fin de estar en mayor contacto con el usuario y poder ofrecerle un mejor servicio.

Ejemplos de servicios que puede ofrecer un call center

- Estudios de mercado
- Encuestas telefónicas
- Servicios de gestión de cobro
- Actualizaciones de datos
- Atención de reclamos
- Venta de productos y/o servicios
- Atención de pedidos
- Manejo del cliente (CRM)



- Telemarketing
- Otros

Pero todo debe tener un orden y una estructura sobre la cual establecerse para tener el mayor control posible de toda la información que se puede llegar a generar de una llamada telefónica.

Se tienen diferentes tipos de información, alguna que es confidencial y no se puede compartir tan fácilmente y otra que no es tan indispensable pero nos sirve para generar estadísticas y saber hacia dónde vamos.

Debido a la saturación que encontramos cada que necesitamos resolver algún problema con nuestra línea de teléfono celular nos dimos a la tarea de buscar una forma de agilizar el proceso de atención dentro de un call center.

En primera instancia debemos conocer la terminología que se usa dentro de estos espacios de trabajo para lograr un acoplamiento ágil y eficaz de nuestra aplicación, Así mismo se abordara las leyes y mejores prácticas que existe alrededor de los call centers, logrando establecer de manera clara los puntos a los que debemos enfocar el desarrollo.

El objetivo que se plantea para este trabajo es analizar, desarrollar e implementar una bitácora de atención a clientes de telefonía móvil, en donde el ejecutivo será capaz de gestionar correctamente los diversos problemas que se presenten en el servicio para canalizarlos al área correspondiente en menor tiempo y así lograr una mayor satisfacción hacia el cliente. Esta bitácora de atención minimizara tiempos de procesamiento de datos, así como tiempos de respuesta a los problemas de los usuarios.



Se dará a conocer la metodología utilizada para lograr el mejor desempeño de acuerdo a las necesidades del usuario, mostrando los beneficios que nos otorga en comparación con las otras metodologías disponibles. Se mostrara el seguimiento de cada uno de los pasos que marca esta metodología ilustrando así el cumplimiento de la misma durante todo el trabajo.

Para el entorno de la aplicación se utilizara un diseño en el cual el ejecutivo del call center pueda interactuar de una manera intuitiva y pueda evitar los errores humanos haciendo el uso de catálogos pre-capturados.

El sistema será desarrollado con software libre que se ajuste a los requerimientos debido a la flexibilidad que sus características entregan.

La base de datos se trabajara con My SQL, PHP en conjunto con Apache por lo que se analizaran las características principales de estas herramientas así como las ventajas y desventajas que nos ofrecen.

Por otra parte se revisaran a detalle las bases de datos relacionales explicando sus ventajas, desventajas estructura y características principales por qué se decidió trabajar con ellas.

Se profundizara dentro del entorno de la programación orientada objetos tomando en cuenta las bondades que esta le puede brindar al proyecto en comparación con otro tipo de programación.

A su vez se explicaran las características a explotar de la programación en PHP y por qué se decidió trabajar con ella.

La aplicación debido a la cantidad de peticiones que tendrá necesita un ambiente cliente servidor por lo que se explicara su funcionamiento y características principales.



Se darán a conocer los requerimientos tanto generales como particulares que nos permitieron ir eligiendo de manera más puntual todas las herramientas que se utilizaron en este trabajo.

Debido a que la información que se maneja es sensible se garantiza su resguardo de manera vital para el desarrollo, ofreciendo con esto poder consultar un histórico y generar reportes en formato de Excel para que sea más fácil su manejo y su presentación en caso de así requerirse.

Se mostrara tanto la estructura que se manejó dentro del cliente y dentro del servidor para lograr la mejor comunicación y a su vez la mejor vista ante el usuario.

Mostrando el diseño y la construcción de la aplicación según los módulos que marca la metodología que se eligió para llevar a cabo el desarrollo.

Se debe explicar la interacción que tenga el usuario con la aplicación para lo cual se usara un diagrama de bloques que ejemplifique de una forma clara y precisa todos los puntos a los que el usuario tiene acceso y los resultados que puede obtener de cada uno de ellos.

La base de datos juega un papel vital dentro de esta aplicación, no solo por el tipo de información si no porque sin ella no se podría llevar a cabo ninguna de las funciones y por consiguiente mejoras que se pretenden lograr.

Para ello se tiene que llevar a cabo desde el diagrama entidad relación, la normalización y todos los detalles que nos ayuden a garantizar el correcto funcionamiento.

Procurando tener siempre los recursos disponibles para evitar caídas innecesarias.

Se plantea un esquema de pruebas para asegurar el buen funcionamiento antes de entregar la aplicación.



Así como un plan de mantenimiento adecuado de acuerdo a la evolución que se vaya presentando.

Se podrán generar reportes con la información recopilada y almacenada en la base de datos que serán útiles para saber el comportamiento que va teniendo el call center en su primer contacto con el cliente.

Resultados Esperados.

Se pretende desarrollar una bitácora de atención a clientes de telefonía móvil, la cual permita capturar y guardar la información primordial sobre la atención al cliente en una base de datos para poder realizar filtros de datos los cuales permitan tener estadísticas del call center y del usuario. Con los siguientes objetivos particulares.

- Medir el tiempo de solución al usuario.
- Agilizar tiempo de captura.
- Almacenar datos para la generación de estadísticas del servicio.
- Mejorar el seguimiento del reporte (agendar callback al cliente)
- Mejorar la imagen de la compañía ante el cliente
- Dar la impresión de que el cliente es primero
- Verificar el tipo de fallas más comunes
- Generar un servicio proactivo
- Ser líderes en la solución de problemas que el cliente reporta



1.2 Conceptos de los términos generales usados en un call center.

ACD: Automatic Call Distributor. (Distribuidor Automático de Llamadas.) En el ámbito de Centros de llamadas es un recurso de telefonía que administra las llamadas telefónicas (entrantes y salientes) y las distribuye a los Agentes, basándose en reglas y algoritmos configurables por el usuario, tales como Balanceo de Carga, Basado en Habilidades, etc.

ANI: Automatic Number Identification (Identificación Automática del Número Llamante) Servicio brindado por las compañías telefónicas mediante el cual al recibirse una llamada, se obtiene simultáneamente el número telefónico del cual proviene.

Back Office: (Servicio de gestión). Organización y procesos de la empresa en los que se basan las operaciones anteriores a la venta y las de posventa.

CallerID: (*Identificación del número llamante*). Característica de todos los teléfonos digitales (y algunos analógicos) de poder desplegar el nombre y/o número telefónico de la parte llamante en una interacción telefónica. La característica debe ser activada por la compañía telefónica. En el ámbito de Call Centers el CallerID es utilizado para acceder a la información almacenada del usuario llamante.

CallFlow:(Flujo de Llamadas) Representación gráfica de un proceso que determina el ciclo de vida de una llamada en un Call Center.

Cola de Llamadas: Cantidad de llamadas que llegan al centro, en espera de ser atendidas por un operador, en un período de tiempo determinado.



CRM: Customer Relationship Management. (*Gestión de las Relaciones con el Cliente*). Es una filosofía empresarial orientada a optimizar las relaciones con sus clientes haciendo uso de los más diversos canales y tecnologías disponibles. Cada etapa de las relaciones cliente/empresa está almacenada en el sistema, de modo que sea posible que los operadores tengan acceso a las informaciones referidas a las experiencias acumuladas de los clientes en todo momento teniendo siempre una visión de conjunto completa del mismo.

DNIS: Dialed Number Identification Service. (*Servicio de identificación del número marcado.*) El Servicio de DNIS se utiliza básicamente en centros de llamadas donde se brindan varios servicios atendidos por el mismo grupo de personas. Es un servicio telefónico que permite identificar el número marcado originalmente por el llamante cuando varios números acaban en un mismo circuito (troncal telefónico). El número marcado es pasado al dispositivo destino de la llamada el cual puede actuar en función de ese dato a la hora de enrutar, encolar o tratar la llamada en general. Se utiliza típicamente para dar un tratamiento diferenciado a los usuarios que llaman a un Call Center.

Estaciones: Son módulos de trabajo que comprenden computadoras, conectadas a un servidor, sistemas de comunicación y de cobro, que permiten que el agente realice la labor de contacto con los clientes especificados en una base de datos.

Firewalls: Hardware o software utilizado en las redes para prevenir algunos tipos de comunicaciones prohibidas por las políticas de red, las cuales se fundamentan en las necesidades del usuario. Denominado también check-point que preserva a la intranet de ataques de intrusos que pudieran accederla. Su objetivo básico es asegurar que todas las comunicaciones entre dicha red e Internet se realicen conforme a las políticas de seguridad de la organización que lo instala.



Gateway: (Puente). Sistema de información que transfiere información entre sistemas o redes incompatibles.

Guion: Son las instrucciones que se utilizan en una campaña para dirigir la interacción de un asesor con el cliente en línea.

Inbound: Llamada entrante.

ISO 9000: se refiere a una serie de normas universales que define un sistema de “Garantía de Calidad” desarrollado por la Organización Internacional de Normalización (ISO) y adoptado por 90 países en todo el mundo, que se pueden aplicar en cualquier tipo de organización (empresa de producción, empresa de servicios, administración pública, entre otros).

IVR: Interactive Voice Response. (Respuesta Interactiva de Voz). Aplicación de software que permite a un usuario interactuar con una entidad sin necesidad de ser atendido por un operador. Acepta una combinación de entrada de voz y teclas al tacto provenientes de un teléfono brindando respuestas apropiadas en forma de voz, fax, callback, e-mail o algún otro medio. Su ventaja principal es la de liberar los recursos humanos corporativos disminuyendo simultáneamente los tiempos de espera de los usuarios.

Java: Lenguaje de programación desarrollado por la empresa Sun para la elaboración de aplicaciones exportables a la red (applets) y capaces de operar sobre cualquier plataforma.

LAN: Local Area Network. (Red de área local). Es una red de datos que cubre un área limitada. Usualmente inscrita en un solo edificio u edificios cercanos.

Marcación Predictiva: Aplicación que permite la realización de campañas masivas salientes a partir de una lista de contactos a los cuales se quiere localizar. El sistema



realiza automáticamente llamadas salientes y transfiere las llamadas respondidas a los agentes. Cuando el sistema de marcado detecta tono de ocupado o no obtiene respuesta, coloca al número nuevamente en la cola según reglas de marcación previamente establecida y específica para cada campaña.

Monitoreo Remoto: A través de este sistema el cliente puede escuchar, desde su localidad, a los asesores realizando las operaciones de telemarketing. Esto le permite evaluar la operación y modificar o mejorar el guion de telemarketing, de considerarlo necesario.

Nivel de Servicio: Indicador porcentual que mide la eficiencia de un Call Center en relación a las llamadas atendidas frente al total de llamadas producidas. Un indicador del 80%, por ejemplo indica que se atienden 8 de cada 10 llamadas que llegan o son emitidas por un Call Center. También indica la probabilidad de que una llamada no sea conectada a un sistema debido a que todos los troncales están ocupados.

Outbound: Llamada saliente.

PBX: Private Branch Exchange. (Sistema de conmutación privado) el cual tiene conexiones internas (extensiones) y conexiones externas (troncales, enlaces privados, etc.). El mismo conmuta las llamadas tanto entrantes como salientes y proporciona acceso a un computador desde una terminal de datos. Resultan una parte indispensable de un centro de contactos pues provee las conexiones necesarias para poder utilizar marcadores predictivos y ACD.

Scripting: Funcionalidad de una solución de Call Center que permite disponer de un guion estructurado que utiliza el Agente para guiar y estandarizar el diálogo telefónico.

SMS (Short Message System): (Sistema de Mensajes Cortos). Procedimiento de envío y recepción de mensajes escritos de pequeño tamaño a través del teclado y la pantalla de los teléfonos móviles.



Speaker Verification: (*Verificación del llamante.*) Tecnología biométrica que permite comprobar la identidad de una persona mediante la identificación de su timbre de voz. Puede ser usada como sustitutiva o complementaria de otras formas de identificación (contraseñas, huellas digitales, etc.). Es sumamente útil en el ámbito de Call Center donde no se puede aplicar ningún otro tipo de sistema biométrico.

Tasa de Abandono: Por ciento de llamadas abandonadas por el sistema, en el momento que el cliente atiende la llamada. Esta medida es importante, dado que mientras mayor es el por ciento de abandono, mayor será el descontento de los clientes cuando al final sean atendidos por un operador del centro, o muy posiblemente su compañía haya perdido más de un cliente sin darse cuenta.

Tiempo de Espera: En este renglón se mide el tiempo que los representantes esperan para que el sistema contacte a un cliente y le transfiera la llamada; mientras mayor es la espera, menor será la productividad del operador.

Velocidad promedio de Contestación de Llamadas (ASA): Esto es el tiempo en promedio que se tarda el centro en contestar las llamadas entrantes.

VoIP (Voice over Internet Protocol): (Voz sobre Protocolo Internet.) es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP (Internet Protocol). Esto significa que se envía la señal de voz en forma digital en paquetes en lugar de enviarla en forma de circuitos como una compañía telefónica convencional o PSTN.

Web Collaboration: (Colaboración en Web.) Permite ofrecer asistencia desde una página Web a través de Internet, ayudando a clientes y usuarios en la navegación. Permite al agente mostrarle al cliente la información adecuada (Web Collaboration), conversar con él en línea (Chat) y/o ayudarlo a rellenar formularios de forma colaborativa.



Wrap-Up: En el ámbito de un Call Center es una medida interna para el tiempo que necesita un representante (agente) para concluir el trabajo administrativo relacionado con una llamada que acaba de atender. Es el trabajo que sigue a la finalización de una llamada. Esto incluye, en el seguimiento de una interacción, al trabajo posterior a la misma: rellenar formularios, efectuar otras llamadas, etc. Mientras está en tiempo de Wrap-Up el agente no se encuentra disponible para recibir llamadas.

1.3 Normatividad de los Call Center.

El gran poder que han adquirido las empresas de Call Center en su manejo de situaciones y control de los usuarios/ clientes ha dado pie a la reacción de normas.

La apertura de la economía mexicana a las corrientes de la globalización a partir del Tratado de Libre Comercio con Estados Unidos y Canadá, en 1994, permite la definición de grandes sectores económicos que serán expuestos a la competencia internacional y, sobre todo, a la inversión foránea. El proceso de apertura se dio a lo largo de ciclos de un importante debilitamiento de la economía: elevación de la tasa de interés, de la inflación, caída del poder de compra de la población y crisis de las empresas. Así, las grandes empresas tanto las nacionales como las que se incorporaban a la economía nacional, especialmente en el sector de servicios, debieron hacer un uso muy rápido de los Call Centers para competir en sus respectivos mercados. Por ejemplo, bancos, para ofrecer masivamente tarjetas de crédito, compañías aseguradoras, para la oferta de seguros, o la competencia entre empresas telefónicas y posteriormente servicios de internet. Ello originó una fase de rápido crecimiento con débiles bases profesionales en la gestión de los Call Centers.



La Norma de Excelencia en Centros de Contacto (NECC), es una iniciativa del Instituto Mexicano de Teleservicios que siguiendo los preceptos de su misión impulsa el desarrollo y profesionalización de la Industria a través de un Programa de Certificación. El programa como tal, permite analizar y evaluar el desempeño de los Centros de Contacto, haciendo uso de parámetros globales enfocados en la operación y administración que especifican cuales son los preceptos a través de los cuales se debe regir la industria.

Modelo Global CIC. Certificación Internacional-CCA.

La Certificación de Call / Contact Centers / Áreas de Interacción con clientes bajo el Modelo Global CIC, tiene como antecedente la Norma de Excelencia de Centros de Contacto (NECC), que ha sido un mecanismo importante para establecer un sistema de gestión que ayuda a mejorar la calidad y satisfacción de los clientes de las organizaciones certificadas. La transición hacia el Modelo Global CIC plantea un enfoque de negocio para responder de manera eficiente a la voz del cliente.

El Modelo Global CIC ha sido diseñado para evaluar, orientar y certificar; el trabajo de áreas u organizaciones que interactúan con los clientes proporcionando servicios ya sea en forma directa o en representación de organizaciones que las contratan; integra la experiencia de la industria capitalizada a través del Instituto Mexicano de Teleservicios como organismo especializado en las diversas áreas y prácticas de la industria, que en conjunto con el Instituto para el Fomento a la Calidad Total, asociación civil, quien otorga el Premio Nacional de Calidad, han diseñado este Modelo avalado por el Consejo Académico de la Industria de Centros de Contacto CAICC.

La Misión del Modelo Global CIC es guiar e impulsar, a nivel internacional, el desarrollo de las Organizaciones y Áreas dedicadas a la Interacción con Clientes en sus esfuerzos de mejora continua, para lograr un desempeño de calidad desde la



perspectiva del cliente, y proveer un marco de referencia que contribuya a la profesionalización de dicha industria. Es por ello que en noviembre de 2013, el IMT formalizó una alianza con la Customer Contact Association (CCA), quien es líder independiente y autoridad en estrategias de operación y contacto con el cliente, con sede en Glasgow y presencia en Europa.

Se trata de una industria de servicios que tiene un conjunto de transformaciones constantes, con algunos procesos particulares de carácter decisivo que se encuentran en el ámbito de la tecnología y que impactan al trabajo y a las empresas de tercerización. En la actualidad se puede definir el siguiente cuadro: la industria de CC experimenta cambios por una tercera oleada tecnológica basada en una diversificación de canales comunicacionales, una mayor integración de las TIC al funcionamiento general de las empresas y también una caída de los costos de infraestructura tecnológica. Las empresas de tercerización se expanden y generan nuevas inversiones y oferta laboral en diversos países, la fuerza de trabajo se diversifica en sus competencias para atender nuevas formas del mercado, en donde la masificación individualizada adquiere un papel importante, y, finalmente, aparecen los primeros rasgos de un proceso de regulación que puede tener influencia en el futuro el desarrollo general de la industria.

Los datos recientes sobre la situación de la industria de CC en México apuntan a un escenario de crecimiento, tanto en empleos como en la diversificación de mercados y en expansión territorial. Resalta una nueva tendencia hacia la construcción de centros de producción de gran capacidad y con alta densidad tecnológica, como síntesis de una transformación tecnológica, especialización laboral, ampliación de mercados y expansión territorial de las empresas de CC.



1.4 Descripción básica de un call center

Call center es una noción de la lengua inglesa que puede traducirse como centro de llamadas. Se trata de la oficina donde un grupo de personas específicamente entrenadas se encarga de brindar algún tipo de atención o servicio telefónico.

Los trabajadores de un call center pueden realizar llamadas (para tratar de vender un producto o un servicio, realizar una encuesta, etc.) o recibirlas (para responder a las inquietudes de los clientes, tomar pedidos, registrar reclamos). En algunos casos, el call center se especializa en una de las dos tareas (realizar o recibir los llamados) mientras que, en otros, cumplen con ambas funciones.

Es importante destacar que el call center puede ser operado por la propia compañía o por una empresa externa. Hay firmas que se dedican a establecer centros de llamadas (con la infraestructura necesaria y el personal entrenado) y comercializan dicha prestación.

La principal ventaja que ofrece un call center a una empresa es que centraliza la atención. Si no se cuenta con un call center, todas las llamadas llegarán a distintas oficinas y resultará más complicado decidir cómo se canalizan y se registran. El call center, en cambio, tiene como única función facilitar la comunicación. Los operarios están capacitados para resolver los asuntos por su propia cuenta y recién derivan la llamada a un ejecutivo en casos excepcionales.

EL GERENTE GENERAL: Es la máxima autoridad ejecutiva, se encarga de planificar, organizar, dirigir, controlar, coordinar, analizar, calcular y deducir el trabajo de la empresa.

EL SUBGERENTE GENERAL: Es la persona que responde por todas las operaciones realizadas en la compañía, ya sea de logística, financiera y de tipo operacional



asimismo debe tomar decisiones que afectan el desarrollo de la compañía y velar por el buen desempeño de nuestros agentes. Además filtra asuntos hacia el gerente.

EL GERENTE DE PROYECTOS: Es la persona que se dedica a planificar, organizar y analizar los proyectos que se establezcan para un futuro asimismo debe brindar las posibles alternativas para que el gerente general tome la decisión según sus criterios.

ASISTENTE ADMINISTRATIVA: Es la persona encargada de las tareas de oficina que el área le exige la compañía tales como organizar, archivar, y colaborar al departamento de finanzas, asimismo es responsable de toda la parte de suministro de esta oficina y demás.

EL REPRESENTANTE LEGAL: Es la persona que se dedica a resolver toda situación de tipo legal que se presente en la compañía asimismo debe brindar asesoría a la parte administrativa sobre los posibles aspectos o parámetros legales que se deben tomar en cuenta antes de tomar una decisión.

EL CONTADOR: Es la persona que vela por que se encuentren en orden todas las situaciones de tipo financiero económico y comercial que surgen dentro y fuera de la compañía, además, debe dar informe sobre todos los movimientos de tipo económico y financiero.

ASISTENCIA TÉCNICA Y SISTEMAS: Es el área persona que vela porque todos los sistemas y/o herramientas técnicas funcionen correctamente para que la compañía tenga un buen desempeño con respecto a las ventas realizadas por los agentes, asimismo debe trabajar en conjunto con el área comercial para que dicha área trabaje de una manera óptima y sin percances.

EL SUPERVISOR: Es la persona que se dedica a guiar, apoyar y asistir al personal que labora en el CC tanto en el área comercial como en el departamento de asistencia técnica.



AGENTE COMERCIAL: Es la pieza clave para que la compañía funcione, es la persona que tiene contacto con nuestro cliente vía telefónica y debe realizar y cerrar la venta de nuestros productos.

Todos los puestos anteriores se muestran en la figura 1.4.1

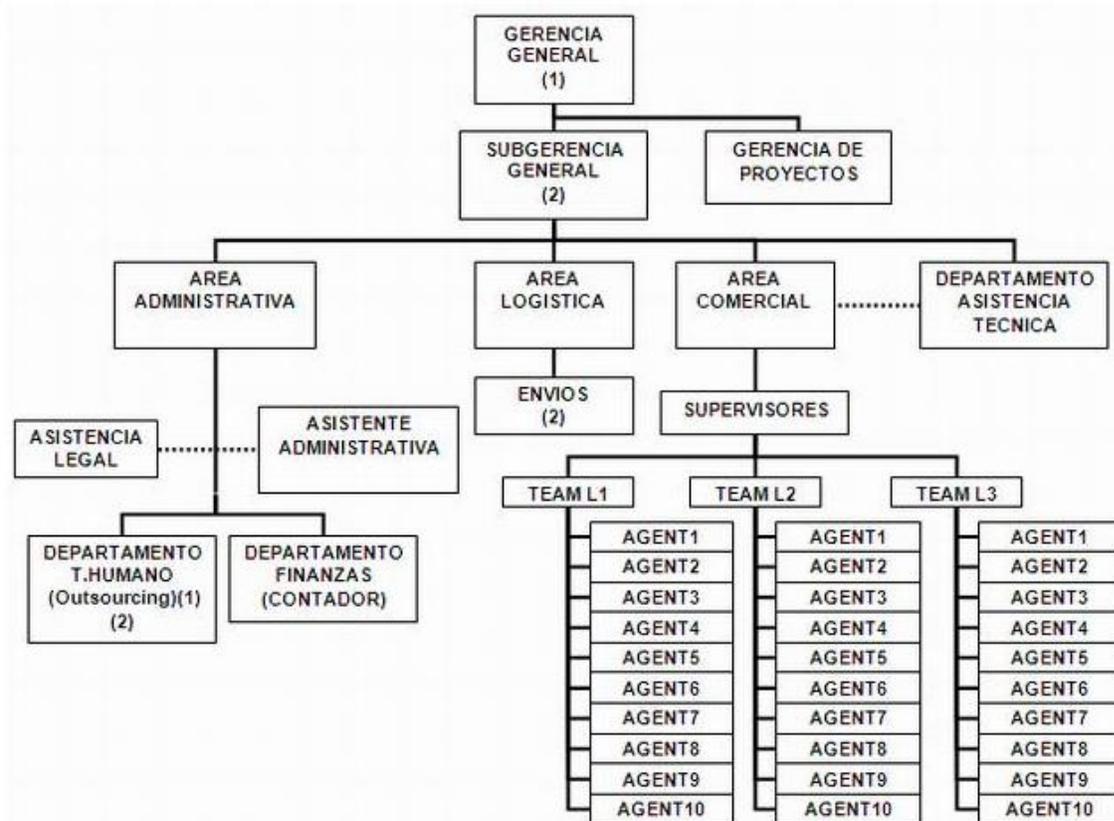


Fig.1.4.1 Organigrama General



Reglamento para el call center.

- En el puesto de trabajo es permitido tener una botella con líquido, pero no es permitido el consumo de alimentos o masticar chicle en el área del call center.
- Los bolsos y maletas deben ser guardados en el locker, antes de ingresar al área del call center.
- Está prohibido tener en el puesto de trabajo maquillaje, espejos, limas, depiladores u objetos de uso personal.
- No es permitido decorar el puesto con afiches, fotos grandes, porta retratos ni porcelanas.
- Prohibido presentarse a su respectivo turno en estado de embriaguez, bajo los efectos de drogas psicoactivas o amanecido.
- No está permitido escuchar música mediante audífonos en el puesto de trabajo y en especial cuando se está en una llamada telefónica.
- Se debe hacer uso adecuado de las salas de espera, los cuales están destinadas a la atención de titulares.
- Se debe mantener un grado de respeto y cordialidad en el trato con todas las personas de la compañía aunque no pertenezcan a su respectivo grupo de trabajo.
- Los baños y las zonas de cafetería se deben conservar en perfecto estado después de su uso, (puertas cerradas, luz apagada, llave bien cerrada).
- Mantener buena disciplina en general. Normas relativas a la presentación personal.
- Está prohibido en el horario de lunes a viernes el uso de prendas informales como: tenis, playeras, bermudas, leguins de colores llamativos o con estampados, botas con peluche o sandalias.



- No se permite los aretes en hombres y piercing tanto en hombre como en mujeres.
- No se permite el cabello largo, ni colas en hombres.
- Se debe portar el carné en un lugar visible todo el tiempo.
- Los teléfonos celulares de uso personal deben permanecer guardados en los casilleros, y solo pueden ser utilizados con autorización del Director o en los horarios de descanso.
- Tiempos para ir al baño y comer están asignados por el supervisor.
- Si se está enfermo se debe reportar al supervisor (gripe, catarro, garganta...) para no dañar la imagen de la empresa.

1.5 Acotamiento del problema a resolver.

La aplicación que se desarrollara cubre en su totalidad las expectativas de un sistema para un agente de un call center, el cual manejara dos puntos importantes dentro del call center, permitirá guardar información sobre las llamadas al agente lo que evitara utilizar documentos de Excel, Word hasta hojas sueltas para el intercambio de Información y se podrán realizar estadísticos para la calidad de la atención y el seguimiento de los asuntos pendientes del agente, creando seguridad al agente para concluir cada incidencia.

Esto ayudara a la productividad de los agentes y al buen almacenamiento de la información, se podrán realizar filtros de casos particulares y exportar la información.

Dentro de los campos básicos que tendrá la bitácora de atención están:

- Forma de contacto:

Seleccionara correo



- Envío de encuesta:

Seleccionará si será necesario o no enviar la encuesta.

- RFC:

Ingreso del RFC del cliente.

- Representante legal:

Ingreso del representante legal de la línea.

- DN:

Ingreso de DN del cliente.

- ¿Es llamada cortada?:

Selección 'SI' 'NO'.

- Ciclo de facturación:

Al seleccionar un ciclo de facturación se habilita la opción de Enviar información de planes y Referencias:

- Encuesta:

Estatus, si se selecciona Acepta Transferencia, se habilita una encuesta de calidad.

- Selecciona una tripleta:

Selección de la tripleta de acuerdo a cada escenario:

- ¿Se solucionó la petición?

Selección del estatus.

- Agendar Callback al cliente:

Si es necesario agendar CallBack

- Comentarios y Guardar información:

La aplicación tendrá la opción de modificar los campos de status y comentarios para la actualización o corrección del registro.

Los filtros que se podrán realizar en la aplicación son:



-
- Mis registros.
 - Por Fecha, status y plataforma

La cual podrá ser exportada a Excel para una mejor manipulación de la información.

La aplicación será consultada vía web para asegurar el acceso desde cualquier centro de atención, la portabilidad permitirá un acceso centralizado y disminuirá el tiempo de actualizaciones.

Capítulo 2 Marco Teórico.

2.1 Características, ventajas y desventajas de las bases de datos relacionales y metodología MVC (modelo–vista–controlador).

Definición de una Base de Datos Relacional.

En una Base de Datos Relacional, todos los datos visibles al usuario están organizados estrictamente como tablas de valores, estructuradas en registros (filas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre del campo, tipo y longitud; a éste campo generalmente se le denomina ID, identificador o clave; todas las operaciones de la base de datos operan sobre estas tablas, a esta manera de construir bases de datos se le denomina modelo relacional .

Características de las bases de datos relacionales.

- Su modelo se basa en el álgebra relacional
- Independencia lógica y física de los datos
- Se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro
- Cada atributo tiene un nombre diferente en las tablas
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- El orden de las tuplas y de los atributos no importa.



- El número de filas (tuplas) recibe el nombre de cardinalidad.
- El número de atributos de una tabla recibe el nombre de grado.
- Al conjunto de valores válidos representados por un atributo recibe el nombre de dominio.
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- La información es atómica, es decir, cada campo de un registro tiene uno y sólo un valor.
- Debe haber consistencia en la información, es decir, no debe haber incongruencias en los datos que se almacenan.
- La información debe ser no redundante, es decir, se debe de eliminar al máximo la existencia de duplicidad de los datos.
- Las relaciones entre dos o más tablas se establecen en virtud de los valores de datos comunes contenidos en las tablas correspondientes.
- Los sistemas de bases de datos relacionales suelen tener asociados lenguajes de consulta de alto nivel, que facilitan y optimizan la realización de búsquedas y actualizaciones en la base de datos.

Ventaja de una base de datos relacional:

- Se puede acceder a los datos a través de llaves.
- Independencia de los datos.
- Se pueden agregar posteriormente datos al sistema sin tener que reescribir lo que ya tenemos.



- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves, por lo tanto no tenemos duplicación de datos, y tampoco tenemos grandes cantidades de “celdas vacías”
- Al eliminar un registro se elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.
- Existencia de numerosos sistemas comerciales entre los que se puede escoger y conseguir apoyo técnico.

Desventajas de una base de datos relacional:

- El acceso a los datos puede llegar a ser lento, dependiendo de la robustez de los sistemas de hardware.
- Presentan deficiencias con datos gráficos, multimedia y sistemas de información geográfica.
- No se manipulan de forma manejable los bloques de texto como tipo de dato.

Metodología MVC (Modelo Vista Controlador)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

Descripción de sus componentes:

- El controlador. Es el elemento encargado de recibir y procesar las peticiones HTTP. Su responsabilidad es retornar una vista concreta con el modelo correcto.
- El modelo. Corresponde con la representación de los datos/entidades de negocio que modelan la aplicación.
- La vista. El contenido que va a ser devuelto como consecuencia de la petición HTTP. La vista recibe el modelo para poder realizar la presentación de los datos.

El siguiente diagrama de la figura 2.1.1 ilustra el proceso.

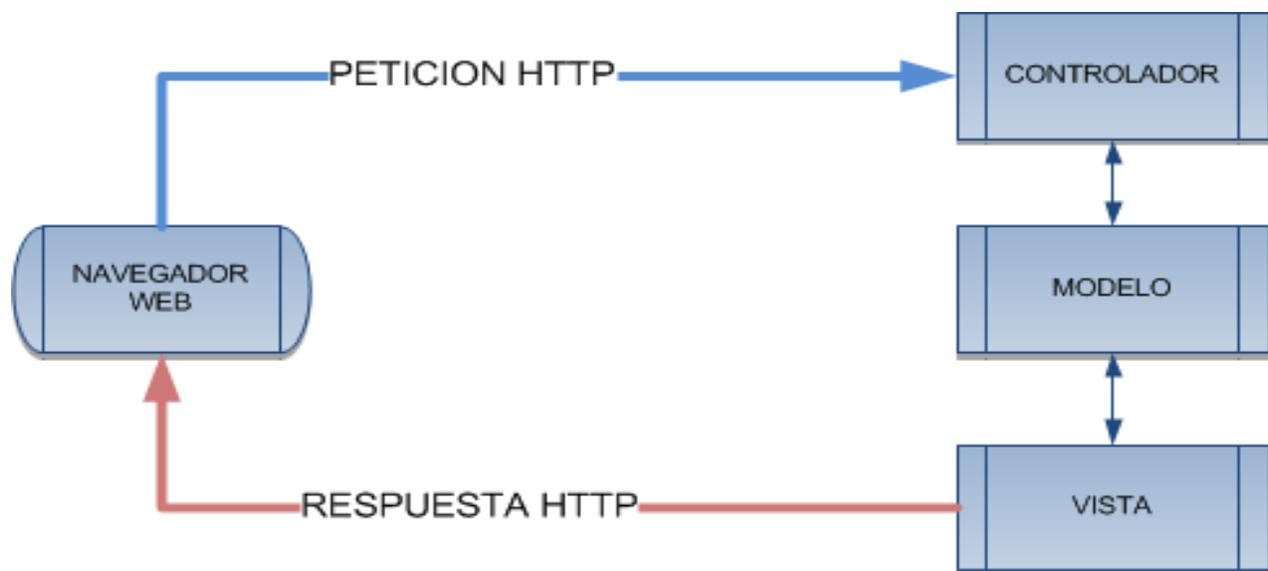


Figura 2.1.1 Proceso de los componentes del modelo MVC

.Ventajas:

- Clara separación entre interfaz, lógica de negocio y de presentación
- Sencillez para crear distintas representaciones de los mismos datos.



- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar un desarrollo guiado por pruebas (TDD).
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.
- Sus vistas muestran información actualizada siempre.
- El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- Las modificaciones a las vistas no afectan en absoluto a los otros módulos de la aplicación.
- MVC es bastante utilizado en la actualidad en marcos de aplicación orientados a objeto desarrollados para construir aplicaciones de gran tamaño; Java Swing, Apache Struts, Microsoft ASP.NET, las transformaciones XSL.

Desventajas:

- Tener que ceñirse a una estructura predefinida, lo que a veces puede incrementar la complejidad del sistema. Hay problemas que son más difíciles de resolver respetando el patrón MVC.
- La curva de aprendizaje para los nuevos desarrolladores se estima mayor que la de modelos más simples como Webforms.
- La distribución de componentes obliga a crear y mantener un mayor número de ficheros.
- El tiempo de desarrollo de una aplicación que implementa el patrón de diseño MVC es mayor, al menos en la primera etapa.
- Requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los módulos de una aplicación.



- Es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

2.2 Características, ventajas y desventajas de la programación orientada a objetos

La Programación Orientada a Objetos supone un cambio en la concepción del mundo de desarrollo de software, introduciendo un mayor nivel de abstracción que permite mejorar las características del código final.

En forma general, las aportaciones de este paradigma se pueden resumir en: Conceptos de clase y objeto, que proporcionan una abstracción del mundo centrada en los seres y no en los verbos.

Los datos aparecen encapsulados dentro del concepto de clase. El acceso a los datos se produce de manera controlada e independiente de la representación final de los mismos. Como consecuencia, se facilita el mantenimiento y la evolución de los sistemas, al desaparecer las dependencias entre distintas partes del sistema. Mediante conceptos como la composición, herencia y polimorfismo se consigue simplificar el desarrollo de sistemas.

La composición y la herencia nos permiten construir clases a partir de otras clases, aumentando en gran medida la reutilización.

Características principales

- Objeto tiene el mismo significado que un nombre o una frase nominal Es una persona, un lugar o una cosa
- Atributos (características que los describen)



- Clase es una caracterización abstracta de un conjunto de objetos

Los principios en que se apoyan las tecnologías orientadas a objetos son:

- Objetos: como instancia de una clase
- Métodos: Es la implementación de un algoritmo que representa una operación o función que un objeto realiza. El conjunto de los métodos de un objeto determinan el comportamiento del objeto.
- Mensajes: Los mensajes son llamados a métodos de un objeto en particular.

Las características que ayudan a definir un objeto son:

- Encapsulamiento: Ocultamiento de información, datos o funciones especiales a los usuarios. En el caso de la programación, el encapsulamiento es lo que permite que tanto la estructura (campos) como el comportamiento (métodos) se encuentren dentro del mismo cuerpo de código de la clase con la que se crean los objetos. Dentro de la clase se deben agrupar tanto la información o datos de los campos como las operaciones, métodos o funciones que operan sobre esta información.
- Modularidad: Es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y vagamente conexos.
- Abstracción: Capacidad del ser humano para entender una situación excluyendo detalles y sólo viéndola a alto nivel.
- Polimorfismo: Esta propiedad indica que un elemento puede tomar distintas formas. Podemos definirlo como el uso de varios tipos en un mismo componente o función. Por ejemplo, una función que suma dos operandos, la cual maneja, o dos números o dos cadenas, para retornar un total de una suma o de una concatenación.

Las clases se organizan para modelar el mundo real en las siguientes relaciones:

- Herencia: (generalización/especialización): Propiedad que permite a los objetos ser construidos a partir de otros; es recibir de un módulo superior sus características,



tales como atributos o funciones (campos y métodos o comportamientos), para usarlos en el módulo actual. Heredar es compartir atributos.

- Asociación: el momento en que dos objetos se unen para trabajar juntos y así, alcanzar una meta.

Ventajas de la programación orientada a objetos

- Reutilización: las clases se construyen de manera que se puedan adaptar a los otros sistemas. Un objetivo fundamental de las técnicas orientadas a objetos es lograr la reutilización masiva al construir el software.
- Estabilidad: Las clases diseñadas para una reutilización repetida se vuelven estables.
- Un diseño más rápido: Las aplicaciones se crean a partir de componentes ya existentes.
- Integridad: Las estructuras de datos (los objetos) sólo se pueden utilizar con métodos específicos.
- Mantenimiento más sencillo: El programador encargado del mantenimiento cambia un método de clase a la vez. Cada clase efectúa sus funciones independientemente de las demás.
- Una interfaz de pantalla sugestiva para el usuario: Hay que utilizar una interfaz de usuario gráfica de modo que el usuario apunte a iconos o elementos de un menú desplegado, relacionados con los objetos.
- Migración: Si se ajustan de manera correcta pueden adaptarse fácilmente



Desventajas de la programación orientada a objetos

- No todos los programas pueden ser modelados con exactitud
- Curvas de aprendizaje largas
- Dificultad en la abstracción
- Limitaciones para el programador.
- Tamaño excesivo en las aplicaciones resultantes.
- Cuando se heredan clases a partir de clases existentes se heredan de forma implícita todos los miembros de dicha clase aun cuando no todos se necesiten, lo que produce aplicaciones muy grandes que no siempre encajan en los sistemas con los que se disponga.
- Velocidad de ejecución (cómo resultado del punto anterior).

2.3 Características ventajas y desventajas de la programación PHP



Figura 2.3.1 Logo PHP

En la figura 2.3.1 se presenta el logo de PHP, (acrónimo de " Hypertext Preprocessor") es un lenguaje de código abierto, adecuado para desarrollo web y que puede ser incrustado en HTML.

Se utiliza para generar páginas web dinámicas. Los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

El navegador PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor.

El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero, para que páginas creadas en este lenguaje funcionen el servidor donde están almacenadas si las debe soportar. Vea figura 2.3.2

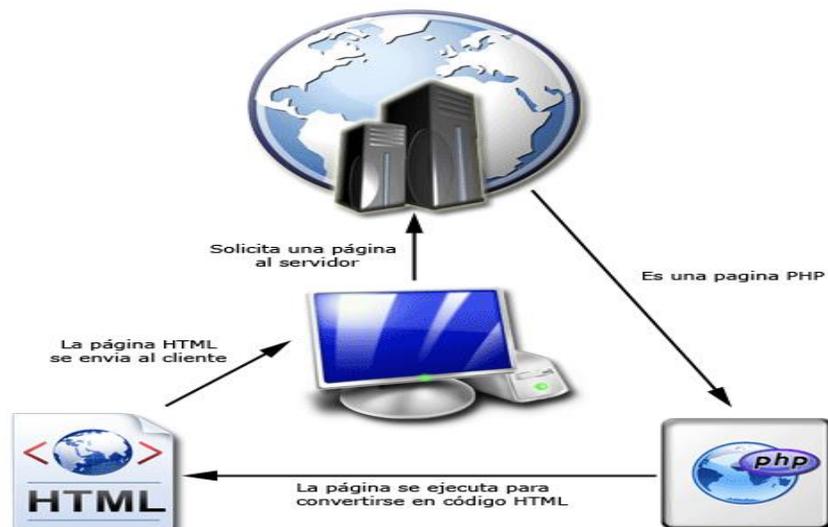


Figura 2.3.2 Funcionamiento de PHP



Características:

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad.

- **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini
- **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Ventajas

- **Multiplataforma:** se trata de un lenguaje que se puede lanzar en casi todas las plataformas de trabajo (Windows, Linux, Mac,...)



- Abierto y gratuito: Pertenece al software licenciado como GNU, la licencia del sistema Linux; lo que permite su distribución gratuita y que la comunidad mejore el código.
- Permite las técnicas de Programación Orientada a Objetos. (Clases y herencia.).
- Gran compatibilidad con los diferentes manejadores de bases de datos
- Extensiones. Dispone de un enorme número de extensiones que permiten ampliar las capacidades del lenguaje, facilitando la creación de aplicaciones web complejas.
- Se puede introducir código PHP en las etiquetas de HTML

Desventajas

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y lo delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación



2.4 Características, ventajas y desventajas del manejador de Base de Datos MySQL.

Introducción

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL. El lenguaje de programación que utiliza **MySQL** Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. En la figura 2.4.1 se presenta el logo de MySQL.



Fig.2.4.1 MySQL



Historia de MySQL

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL. El origen del nombre MySQL no se sabe con certeza de donde proviene, por un lado se dice que en sus librerías han llevado el prefijo “my” durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido cual de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos.

Características principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

Las características principales son:

- El principal objetivo de MySQL es velocidad.
- Soporta gran cantidad de tipos de datos para las columnas.



- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

Ventajas

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.

Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).
- Reindexado: Si tiene muchos índices, nuestro fichero puede alcanzar nuestro tamaño máximo de fichero antes que el de datos.
- Integridad Referencial: es una limitación que se aplica a una base de datos relacional, en la que los datos y las relaciones entre ellos están organizados en



tablas de filas y columnas, para que no se introduzcan datos inconsistentes. La mayoría de los sistemas gestores de bases de datos relacionales definen reglas de integridad referencial que los programadores aplican cuando crean las relaciones entre dos tablas.

2.5 Características, ventajas y desventajas del ambiente cliente servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

Algunos ejemplos de aplicaciones computacionales que usen el modelo cliente-servidor son el Correo electrónico, un Servidor de impresión y la World Wide Web. Ver Figura 2.5.1

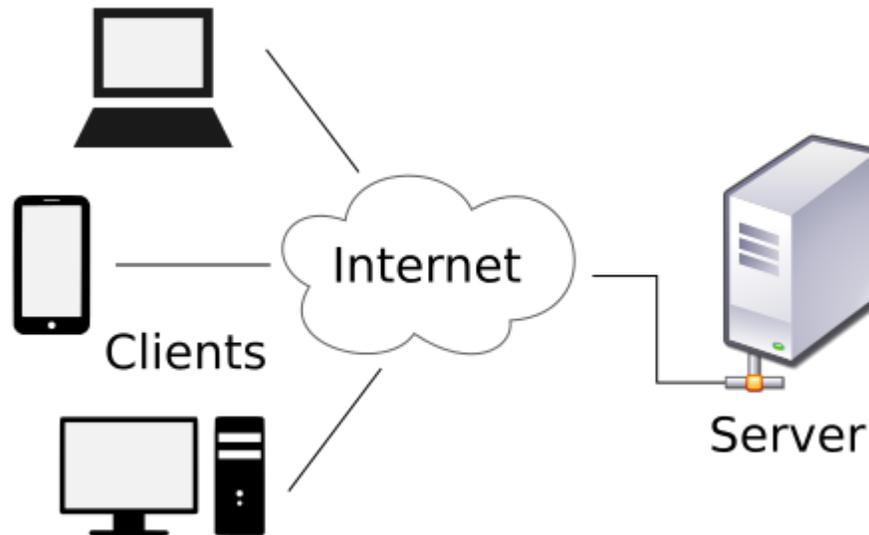


Figura 2.5.1 Diagrama Cliente-Servidor

En la arquitectura Cliente-Servidor sus características son:

- El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no

muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

- Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización.
- La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio. Ver Figura 2.5.2

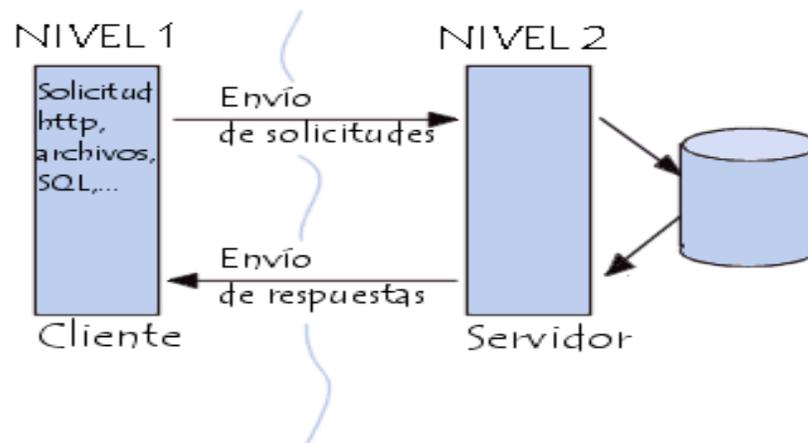


Figura 2.5.2 Arquitectura Cliente-Servidor de dos capas.

- En la arquitectura en 3 niveles, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:
- Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación.

- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere. Ver Figura 2.5.3

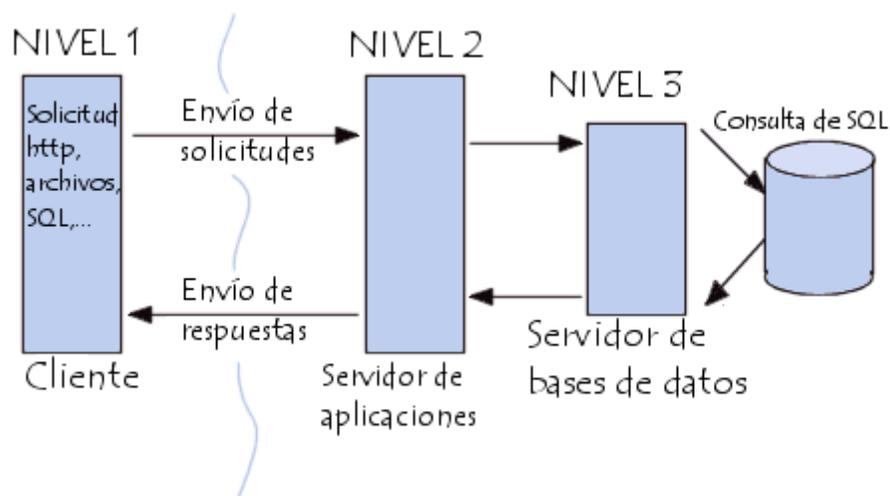


Figura 2.5.3 Arquitectura cliente-servidor de tres capas.

Ventajas

El modelo cliente/servidor se recomienda, en particular, para redes que requieran un alto grado de fiabilidad. Las principales ventajas son:

- Recursos centralizados: Debido a que el servidor es el centro de la red, puede administrar los recursos que son comunes a todos los usuarios, por ejemplo: una base de datos centralizada se utilizaría para evitar problemas provocados por datos contradictorios y redundantes.



- Seguridad mejorada: Ya que la cantidad de puntos de entrada que permite el acceso a los datos no es importante.
- Administración al nivel del servidor: Ya que los clientes no juegan un papel importante en este modelo, requieren menos administración.
- Red escalable: Es posible quitar o agregar clientes sin afectar el funcionamiento de la red y sin la necesidad de realizar mayores modificaciones.
- Think Client (cliente liviano): Se refiere a una aplicación a la que se puede acceder por una interfaz Web (en HTML), que se puede visualizar con un navegador Web en donde toda la lógica comercial se realiza en el lado del servidor. Por eso, al navegador a veces se le denomina cliente universal.
- El origen del término se debe a la ausencia de un lenguaje HTML rico, que sólo permite interfaces con relativamente poca interactividad; esta problemática no se encuentra en el lenguaje JavaScript.
- El hecho de que todo el procesamiento fundamental ocurra en el lado del servidor, con la interfaz gráfica siendo enviada al navegador a cada solicitud, significa que es muy flexible cuando realiza las tareas de actualización. Sin embargo, la aplicación debe poder leer las diferencias a la hora de interpretar el código HTML por los diferentes navegadores. También la ergonomía de la aplicación es limitada.
- Fat Client (cliente pesado): Se utiliza para una aplicación gráfica de cliente que se ejecuta en el sistema operativo del usuario. Un cliente pesado suele tener una mayor capacidad de procesamiento y es posible que tenga una interfaz gráfica sofisticada. Sin embargo, esto conlleva un desarrollo adicional y suele ser una mezcla de la lógica de presentación (interfaz gráfica) con la lógica de la aplicación (potencia de procesamiento).
- Este tipo de aplicación suele instalarse en el sistema operativo del usuario y se debe instalar una nueva versión cuando se realiza una actualización. Para solucionar esto, los programadores de aplicaciones pesadas, por lo general, incorporan una funcionalidad que se ejecuta al iniciar la aplicación y verifica un



servidor remoto para saber si está disponible alguna versión más nueva. De ser así, le indica al usuario que descargue e instale la actualización.

Desventajas

La arquitectura cliente/servidor también tiene las siguientes desventajas:

- Costo elevado: Debido a la complejidad técnica del servidor.
- Un eslabón débil: El servidor es el único eslabón débil en la red de cliente/servidor, debido a que toda la red está construida en torno a él.
- DeadLock (Bloque Mutuo): Es el bloqueo permanente de un conjunto de procesos o hilos de ejecución en un sistema concurrente que compiten por recursos del sistema o bien se comunican entre ellos. A diferencia de otros problemas de concurrencia de procesos, no existe una solución general para los interbloqueos.

Capítulo 3.- Planteamiento del Problema.

3.1 Análisis del problema.

Uno de los principales factores que causan problemas en un call center es el tiempo en el que se atiende al cliente y las múltiples canalizaciones, es por eso necesario contar con un sistema de seguimiento de incidentes que administre y mantenga listas de incidentes, para crear, actualizar y resolver incidentes reportados por usuarios, o inclusive incidentes reportados por otros empleados de la organización para optimizar el tiempo de respuesta. Es importante contar con una base de conocimiento que contenga información de cada cliente, soluciones a problemas comunes y otros datos relacionados, vea figura 3.1.1



Figura 3.1.1 Planteamiento del problema.

1.1. El cliente hace uso del call center para reportar una falla o solicitar información.



2.1 El Agente toma la llamada y escucha lo que requiere el cliente.

2.1.1 El agente resuelve el problema y se cierra el reporte del cliente

2.1.2 El agente no cuenta con las herramientas necesarias para solucionar el problema y canaliza con un especialista al cliente.

3.1 El especialista da seguimiento al reporte hasta su cierre.

4.1 Se almacena la información del reporte desde su inicio hasta el cierre.

5.1 Se extrae información de la BD para obtener estadísticas del agente y de las fallas comunes.

Toda empresa que tenga un área de call center debe ofrecer un servicio eficiente y de calidad.

Algunas compañías deciden tener este departamento integrado dentro de la propia empresa muchas veces se toma esta decisión por desconfianza en una empresa externa, o simplemente porque no se le da importancia a esta área.

Generalmente las empresas que deciden tener un call center dentro de la propia empresa (inhouse) es por un tema de confidencialidad ya que no les gustaría entregar su información como bases de datos, prácticas, precios etc... a un intermediario que pudiera divulgar información.

Una de las principales desventajas de esta decisión es que no se está preparado para llevar un proceso de selección de personal para call center, por lo tanto se pueden cometer errores al seleccionar al personal de esta área.

Otro punto débil en un call center inhouse, es el software especializado con el que se cuenta para la operación, por lo general este tipo de programas demandan una fuerte inversión y no siempre se aprovechan todos los beneficios que la herramienta ofrece por las limitaciones técnicas o de recursos dentro del call center interno.



Por lo general las empresas no fueron diseñada para tener como huésped un call center, por lo tanto no cuenta con una estructura especializada en el monitoreo de llamadas, ni estándares de calidad con los cuales se deba medir al centro, adicionalmente para cualquier refuerzo en las habilidades a desarrollar por parte del personal del call center, es necesario contratar compañías que brinden esta asesoría, lo cual representa inversión adicional.

Podríamos indicar que una ventaja de contar con un call center inhouse, es la velocidad en los tiempos de respuesta y la resolución de problemas del cliente, ya que al estar dentro de la misma compañía y conocer cada departamento es más sencillo el escalamiento correcto de cada caso para su inmediata atención.

Otras empresas prefieren delegarlo y dejar el servicio de atención al cliente en empresas especializadas en este sector, dedicadas exclusivamente a ello.

A continuación, planteamos 4 problemas que se evitan al delegar el servicio de atención telefónica:

- Pérdidas de tiempo en la gestión de problemas reportados y reclamaciones. Un call center debe disponer de los recursos necesarios y personal calificado para gestionar de manera rápida y eficaz los problemas reportados y reclamaciones de los clientes. Si la información que se solicita al cliente no es suficiente para detectar el problema puede ser mal canalizado el reporte lo que provocaría estar transfiriendo al cliente con diferentes especialistas repitiendo información del problema lo que causaría agobio y frustración al cliente o bien si no se anotan los detalles del incidente del cliente puede enviarse mal la información al especialista y provocar nuevamente mayor tiempo de solución al cliente, por eso es tan importante la precisión y el manejo de la información.



- Rompimientos de cabeza para la rentabilización de costes. Tener un servicio de atención al cliente de calidad requiere la dedicación de recursos a tal fin por eso es necesaria una formación rigurosa de las personas que van a realizar este trabajo así como también unas instalaciones adecuadas para ello que nos permitan llevar una gestión del servicio.
- Citas y llamadas perdidas o traspapeladas. Si no se dispone del sistema o la herramienta adecuada para gestionar el volumen de llamadas que entran a diario en una empresa, éstas se pueden perder, o lo que es peor: recoger el recado y perderlo. En cualquier compañía mediana entra un volumen de llamadas importantes a diario, por ello para llevar un seguimiento adecuado de las mismas (recados, mensajes, gestiones, pedidos...) es necesaria una herramienta exclusivamente para tal fin.
- Imagen dañada por el descuido de clientes. Es común que a determinadas horas del día haya un desbordamiento de llamadas; o que se disponga de un servicio de atención al cliente deficiente. Esto supone la decepción, o el enfado de algunos clientes que no dudarán en contar su mala experiencia dañando de manera importante la imagen de la empresa.

El mundo del Call Center se beneficia sin lugar a dudas de todos los avances tecnológicos que se producen a diario, no sólo para facilitar el trabajo o llegar a más clientes, sino para generar más ingresos.

- Reportes y Métricas en Tiempo Real.

Atrás quedaron los días en los que el Call Center requería procesar grandes cantidades de datos manualmente cada cierto tiempo para obtener los resultados de la operación. El poder de la computación actual se refleja principalmente en la capacidad de



procesamiento de datos, encontrar patrones y realizar análisis que a los humanos nos tomaría mucho tiempo realizar.

- Comunicación OmniCanal.

Hasta ahora empezamos a ver una verdadera comunicación multicanal y multiplataforma. En un Call Center el teléfono ya no es el único canal de comunicación, el email no ha sido desbancado y el chat ha llegado para quedarse incluyendo el chat social.

- Software Listo para la Virtualización y la Nube.

Todo Call Center que se tome en serio no sólo sus costos de operación sino el retorno de su inversión, está considerando virtualizar sus sistemas de información, el primer paso es la virtualización de las aplicaciones. Los sistemas tradicionales bajo premisa ofrecían robustez a un alto costo, los nuevos sistemas suelen ser más flexibles, a menor costo y no sacrifican funcionalidad sino que al contrario tienen mucho más que ofrecer.

- Entrenamiento e Incentivos al Empleado.

. Las nuevas plataformas de Software para Call Center permitirán a los Administradores y Supervisores tomar acción durante el momento exacto en el que suceden los errores (ya sea interviniendo en la conversación de voz, o apoyando al agente a través de mensajes de texto en su pantalla) ya que es posible grabar todas las llamadas permitiendo mejorar el entrenamiento del personal del Call Center. Así mismo aquellos agentes que cumplan con las normas de control y productividad de seguro serán incentivos por programas que procuren su permanencia por mayor tiempo en el centro de contacto.

- Análisis Biométrico de la Voz.



Aunque aún es una tecnología reciente, cada vez es más frecuente ver cómo el análisis de voz está siendo tomado en serio, ya sea para identificar estados de ánimo o para identificar a la persona aun es un lujo que no todo Call Center puede justificar, pero es interesante ver lo que sucede en este campo.

- Flexibilidad sin Precedentes.

La VoIP al Call Center ha sido el principal promotor del cambio, aun son muchos los centros de contacto que tienen que migrar a esta tecnología a pesar de que ya tiene más de una década que se empezó con el cambio. De la mano de la VoIP se verán también nuevas tendencias tales como los Agentes Virtuales que trabajan desde sus casas o incluso desde otra ciudad o país, los sistemas fácilmente escalables con flexibilidad increíble a la hora de operar, integrarse con aplicaciones de terceros o permitir ser personalizados a la necesidad de la compañía. Otros beneficios que ya se pueden observar son los sistemas de auto marcado basados en listas que se integran con sistemas de audio respuesta y aplicaciones de terceros a niveles nunca antes vistos.

Definitivamente estaremos viendo cómo algunas tendencias son la consolidación de cambios que se vienen dando hace algunos años, y otras son realmente novedosas por eso es necesario ajustar la realidad del Call Center para aprovechar estas nuevas tendencias, para que los cliente estén más satisfechos.

3.2 Requerimientos generales y particulares de la aplicación.

Un requerimiento es un conjunto de propiedades o restricciones definidas con precisión, que un sistema de software debe satisfacer.



Los requerimientos son pieza fundamental en un proyecto de desarrollo de software, ya que son la base para:

- Planear un proyecto y los recursos que se necesitan para su desarrollo.
- Especificar el tipo de pruebas que se habrán de realizar al sistema.
- Planear la estrategia de prueba a la que habrá de ser sometido el sistema.
- Son el fundamento del ciclo de vida de un proyecto.

Los requerimientos bien formulados deben satisfacer varias características. Si no lo hacen, deben ser reformulados hasta hacerlo.

Algunas características son:

- Necesario: Lo que pida un requerimiento debe ser necesario para el producto.
- No ambiguo: El texto debe ser claro, preciso y tener una única interpretación posible.
- Conciso: Debe redactarse en un lenguaje comprensible por los clientes en lugar de uno de tipo técnico y especializado, aunque aun así debe referenciar los aspectos
- Consistente: Ningún requerimiento debe entrar en conflicto con otro requerimiento diferente, ni con parte de otro, es decir el requerimiento no debe contradecir ningún otro requerimiento.
- Completo: Los requerimientos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle.
- Alcanzable: Un requerimiento debe ser un objetivo realista, posible de ser alcanzado con el dinero, el tiempo y los recursos disponibles.



- Verificable: Se debe poder verificar con absoluta certeza, si el requerimiento satisface las necesidades del cliente o no.

Para el desarrollo óptimo de nuestro sistema, definiremos los requerimientos que nos permitirán cubrir todas las necesidades existentes.

Requerimientos generales.

- El motor de base de datos y servidor web requerido MySQL Community Server (GPL) 5.6.12-log (Minino) y Apache/2.4.4 (Win32) PHP/5.4.14(Minino)
- Debe contar con un módulo para inicio de sesión y autenticación de usuarios
- La autenticación debe ser por medio de un nombre de usuario y contraseña para cada usuario, mismos que serán asignados por el administrador del sistema.
- La aplicación debe trabajar bajo un ambiente web, para asegurar el acceso desde cualquier lugar.
- Se desea presentar la información estructurada mediante menús de ventanas para que cualquier usuario pueda acceder con una mínima capacitación.
- Es necesario que sea posible navegar con el mouse y el teclado.
- Debe ser fácil en su instalación para que no presente rechazo por parte del personal encargado de su instalación.
- Se quiere que el sistema sea actualizable por medio de una interface Web sin necesidad de reinstalar el producto.
- Deberá ser una interfaz sencilla, moderna y amigable, que facilitará el llenado de datos requeridos y se optimice el tiempo de llamada.
- Se requiere que la información se guarde en una base de datos.



Requerimientos particulares.

- La aplicación debe contar con una pantalla de reportes donde se pueda realizar filtros por fecha, status y tipo de plataforma y esta pueda ser exportada a Excel.
- La vista de la consulta debe tener los campos DN, MOTIVO, FECHA, ESTATUS, ID, EJECUTIVO, SUPERVISOR y PLATAFORMA.
- La opción motivos de la falla se requiere mostrar con un menú desplegable.
- Se requiere que la aplicación cuente con un distintivo para las llamadas que requieren seguimiento, el cual contenga los campos de motivo, fecha y hora.
- Se solicita que se puedan modificar registros guardados.
- Se requiere mostrar en pantalla el número de visitantes al sistema.
- La pantalla de la bitácora de atención a clientes de telefonía móvil debe contar con un checkbox para indicar si se solucionó la petición del cliente.

3.3 Análisis y posicionamiento de los lenguajes para el back-end y front-end

Para la construcción de un sistema informático es de suma importancia elegir la metodología y las herramientas adecuadas. Actualmente donde el cambio y desarrollo tecnológico tanto en hardware como software, es constante, la oferta es muy amplia.

Para llevar a cabo una buena elección se tienen que tomar en cuenta diversos factores tales como el tamaño y tipo de la aplicación a desarrollar, el costo-beneficio, el número de usuarios que lo emplearan, el equipo disponible, el tiempo de entrega del sistema y otros factores adicionales. Tomando en cuenta estos factores se analizan algunas opciones de solución, para posteriormente seleccionar la que mejor se ajuste a los requerimientos del sistema que se desea desarrollar.



Back-End (Base de Datos)

Para la elección del back-end se analizaron tres bases de datos MySQL, PostgreSQL y Firebird, todas ellas son de distribución y utilización libre.

Firebird

Es un software de administración de bases de datos relacional, similar en propósito a productos tales como DB2; Oracle ó PostgreSQL. Tiene dos componentes principales el servidor de base de Datos, que vive en la misma máquina que la Base de Datos, y la interface de aplicación, comúnmente referida como la Librería del Cliente. Ver fig 3.3.1

Características principales de firebird

- Integridad referencial
- Arquitectura Multi Generacional
- Muy bajo consumo de recursos
- Soporte nativo para los principales sistemas operativos, incluyendo Windows, Linux, Solaris, MacOS y otros.
- Backups incrementales
- Disponible para arquitecturas de 64bits
- Altos volúmenes de almacenamiento
- Seguridad basada en usuarios/roles.



Fig.3.3.1 Firebird

PostgreSQL

Sistema de base de datos objeto-relacional. El código fuente es libre, usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Ver fig 3.3.2

Características principales de postgresQL

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP ...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (Online/hot backups).
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)

- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

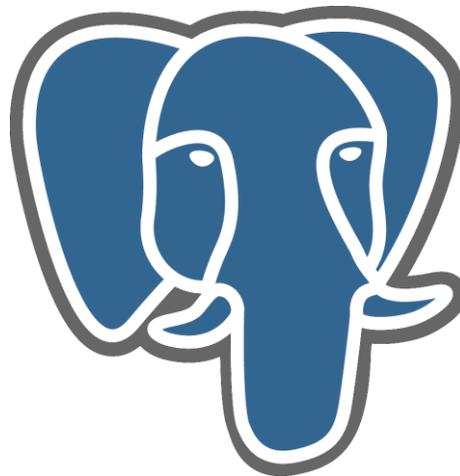


Fig.3.3.2 PostgreSQL

La tabla 3.3.3 muestra la comparación de algunas características de las tres Bases de datos anteriormente mencionadas.

CARACTERISTICAS	FIREBIRD 2.5	MYSQL 5.6	POSTGRESQL 9
Integridad Referencial	Si	Si	Si
Licenciamiento	Gratuito	Gratuito/Cobro	Gratuito



Soporte de 64 bits	Si	Si	Si
Limite por Base de Datos	Limitado	Ilimitado	Ilimitado
Limite por Tabla	Limitados	16 TB en Solaris	32 TB
Número de índices por tabla	Limitados	64	Ilimitados
Limites de Cadenas (VARCHAR)	32765 bytes	65,532 bytes	1 Gb
Soporte de Unicode	Si (UTF-8 y 16)	Si (UTF-8)	Si (UTF-8)
Transacciones	Si, conformidad ACID	Si, conformidad ACID	Si, conformidad ACID
Candados a nivel renglón de Tabla (conurrencia)	Modelo MVCC	Modelo MVCC solo para tablas InnoDB	Modelo MVCC
Funcionalidad de TIMEOUT	No soportada	No soportada	Si soportada
Soporte de Triggers	Soportada	Soportada, con restricciones	Soportada
Escalabilidad/Soporte SMP	Soporte solo en el clasic Server	Soportada	Soportada
Balanceo de cargas	No soportado	Soporte parcial	Soportado



Replicación	Soportado	Soportado	Soportado
Respaldo en Línea (Online Backup)	Soportada con comando en línea	Soportado con la utilería Mysqldump	Puede realizar backups continuos

Tabla 3.3.1 Tabla comparativa de la base de datos.

Front End (Software para programar)

Para la elección del front end se analizaron tres lenguajes de programación PHP, Python y Ruby.

Ruby

Es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Ver figura 3.3.3



Figura 3.3.3 Ruby

Características principales de ruby

- Manejo de excepciones, como Java y Python, para facilitar el manejo de errores.
- Permite escribir extensiones en C para Ruby es más fácil que hacer lo mismo para Perl o Python, con una API muy elegante para utilizar Ruby desde C. Esto



incluye llamadas para embeber Ruby en otros programas, y así usarlo como lenguaje de scripting.

- Puede cargar bibliotecas de extensión dinámicamente si lo permite el sistema operativo.
- Tiene manejo de hilos (threading) independiente del sistema operativo. De esta forma, se tiene soporte multi-hilo en todas las plataformas en las que corre Ruby, sin importar si el sistema operativo lo soporta o no.
- Altamente portable.
- Un verdadero mark-and-sweep garbage collector para todos los objetos de Ruby. Lo que significa que no es necesario mantener contadores de referencias en bibliotecas externas
- Ruby es fácilmente portable: se desarrolla mayoritariamente en GNU/Linux, pero corre en varios tipos de UNIX, Mac OS X, Windows 95/98/Me/NT/2000/XP, DOS, BeOS, OS/2, etc.

Python

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Ver figura 3.3.4

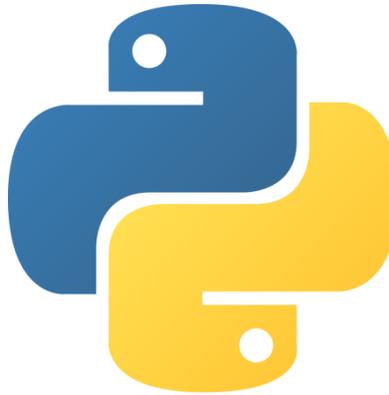


Figura 3.3.4 Python

Características principales de python

- Es un lenguaje muy simple, por lo que es muy fácil iniciarse en este lenguaje. El pseudo-código natural de Python es una de sus grandes fortalezas.
- Se puede crear todo tipo de programas; programas de propósito general y también se pueden desarrollar páginas Web.
- Lenguaje Orientado a Objetos es construido sobre objetos que combinan datos y funcionalidades.
- Se puede insertar lenguaje Python dentro un programa C/C++ y de esta manera ofrecer las facilidades del scripting.
- Contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero.

La tabla 3.3.4 muestra la comparación de algunas características de los tres lenguajes de programación anteriormente mencionadas.



CARACTERISTICAS	Ruby	PHP	Python
Programación	Orientada a objetos	Orientada a objetos	Orientada a objetos
Conectividad a Base de datos	Todas	Todas	Todas
Herramienta de generación de SQL	SI	SI	SI
Gestión de excepciones	Si	Si	Si
Sitios Web en	Twitter Hulu	Facebook Wikipedia	Youtube Google
Aprendizaje del lenguaje de programación	Conocimiento previo en lenguajes orientados a objetos (Avanzado)	Fácil aprendizaje con formación previa en programación en lenguaje C. (Medio)	Sintaxis intuitiva, recomendado para personas que empiezan a programar (Fácil)

Tabla 3.3.2 Tabla comparativa de la base de datos.



Web Server (Servidor Web)

Para la elección del web server se analizaron tres de web server apache, Cherokee y NGINX.

Apache

Es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento. Ver figura 3.3.5



Figura 3.3.5 Servidor Apache

Características principales de Apache

- Instalación/Configuración. Software de código abierto.
- Costo. El servidor web Apache es completamente gratuito.
- Funcional y Soporte. Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyan constantemente con

mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.

- Multi-plataforma. Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- Rendimiento. Capacidad de manejar más de un millón de visitas/día.
- Soporte de seguridad SSL y TLS.

Cherokee

Es un servidor web ligero (esto es, con un bajo consumo de memoria), pero ofreciendo más funcionalidades de serie que alternativas más populares como nginx. Ver figura 3.3.6



Figura 3.3.6. Servidor Cherokee

Características principales de Cherokee

- Interfaz de administración web muy potente e intuitivo, incluso con asistente para configurar de manera óptima diferentes tipos de aplicaciones. Es realmente fácil configurar y optimizar uno o varios sitios virtuales sin tener que pasar horas en

archivos de configuración, aunque obviamente también ofrece esta posibilidad para administradores más experimentados.

- Soporte para scripts CGI
- Excelente documentación
- Varias funciones útiles para su uso como servidor de ficheros multimedia que son muy útiles para instalar galerías de videos. Por ejemplo las barras de progreso de subida como Uber uploader funcionan perfectamente, a diferencia de otros servidores. También tiene funciones avanzadas para el streaming para servir los ficheros de audio y video de manera inteligente ajustándose a la velocidad del cliente, y no enviando el fichero completo si no es necesario, lo que ahorra ancho de banda y recursos en el servidor.

Nginx

Es un servidor web y proxy inverso, multiplataforma, ligero y de alto rendimiento. Nginx es software libre, liberado bajo licencia BSD. Ver figura 3.3.7



Figura 3.3.7 Servidor NGINX

Características principales de NGINX

- Capaz de manejar más de 10.000 conexiones simultáneas con un uso bajo de memoria.



- Es multiplataforma y fácil de instalar: La mayoría de las grandes distribuciones de GNU/Linux, tienen Nginx en sus repositorios. Es compatible con sistemas operativos Windows, IOS y Linux.
- Balanceador de carga: Este servidor web puede funcionar como balanceador de carga, distribuyendo el tráfico entre varios servidores, permitiendo mayor escalabilidad.
- Soporte comunitario y profesional: Nginx, Inc está detrás del desarrollo de Nginx, además de la comunidad en general, permitiendo tener un soporte tanto profesional como comunitario.
- Compatibilidad con las aplicaciones web más populares: Nginx es compatible con una gran cantidad de CMS existentes en el mercado, y hay un sinfín de tutoriales y documentación para instalar bajo Nginx por ejemplo: Wordpress, Joomla, Drupal, phpBB y más! Ver tabla 3.3.3

CARACTERISTICAS	Apache	Cherokee	NGINX
Costo	Gratis	Gratis	Gratis
Balanceador de carga	X	X	SI
Soporte	SI	SI	SI
Sistema operativo	Windows, IOS y	Windows, IOS y	Windows, IOS y



	LINUX	LINUX	LINUX
--	-------	-------	-------

Tabla 3.3.3 Compatibilidad

Se eligió MySQL dado que su diseño está orientado a la velocidad, así como a soportar un gran número de transacciones sin afectar esto al desempeño de la aplicación.

En resumen, los requerimientos mínimos para la implementación con relación al software son:

Requerimientos en el servidor:

- Windows server 2008 Enterprise R2
- Web server Apache
- MySQL
- PHP

Requerimientos en el cliente:

- Mozilla Firefox

Con relación al hardware mínimo en el servidor es:

- Intel Pentium core 2 Duo 2.2 GHz
- 2 GB de memoria RAM
- 160 GB de disco duro
- Monitor

3.4 Aplicación y separación de los módulos según la metodología MVC.

El programa va a funcionar de la siguiente manera:

- El modelo, la vista y el controlador se inicializan
- Se muestra la vista al usuario leyendo datos del modelo
- El usuario interactúa con la vista, lo cual llama a una acción específica del controlador.
- El controlador modifica el modelo.
- La vista se refresca, mostrando el cambio en el modelo. Ver figura 3.4.1

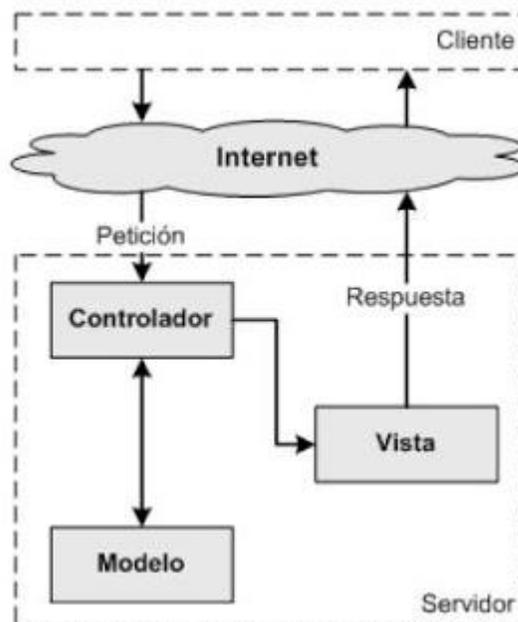
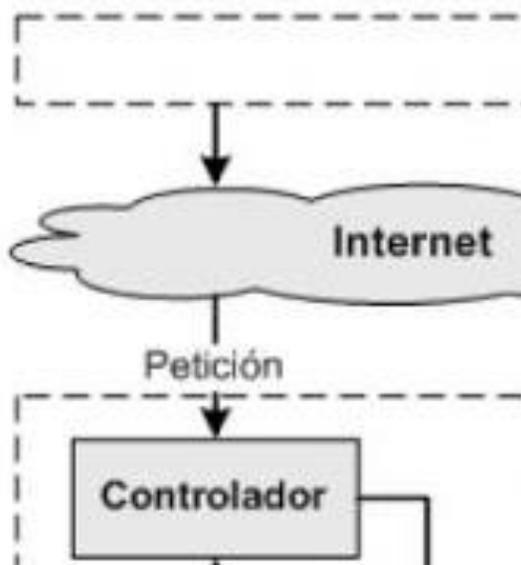
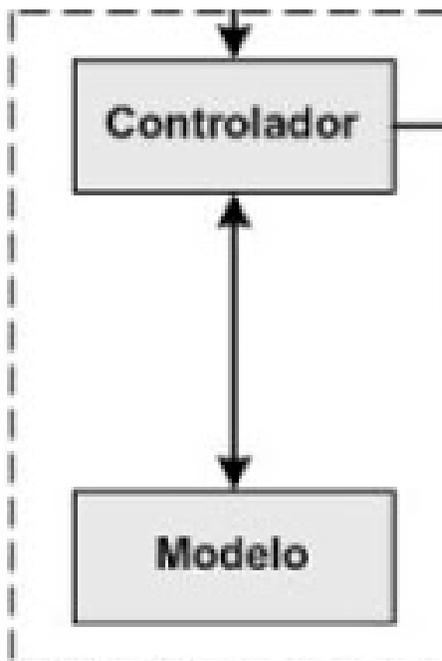


Figura 3.4.1 Modelo MVC



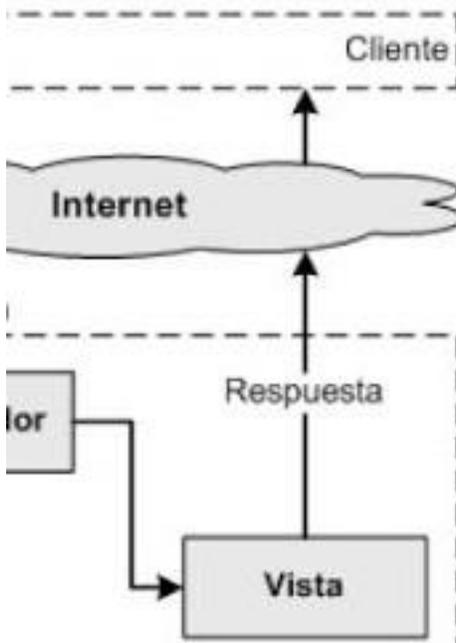
Separando el modelo MVC, el controlador es mucho más fácil de leer. Su única tarea es la de obtener los datos del modelo. El controlador se encarga de procesar las peticiones, las sesiones de los usuarios y la autenticación. Ver figura 3.4.2

Figura 3.4.2 Controlador



La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Ver figura 3.4.3

Figura 3.4.3 Interacción Controlador y Modelo



La capa de la vista también puede aprovechar la separación del código. Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el layout genérico, el pie de página y la navegación global. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. La vista es la respuesta que se muestra al usuario. Ver figura 3.4.4

Figura 3.4.4 Vista

En el modelo MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de alguna forma (el usuario pulsa un botón, enlace).
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega.
- El controlador accede al modelo, actualizándolo, modificándolo solicitado por el usuario.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí



mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se **actualice**. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista, según lo descrito en la segunda variante.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

3.5 Diseño a nivel de bloques de la interfaz del usuario.

En la siguiente figura 3.5.1 se mostrara el bosquejo para iniciar sesión al sistema se presentara un mensaje de bienvenida y dos campos para autenticar el usuario, los cuales serán el nombre del usuario y la contraseña, así como el botón para iniciar sesión.



Figura 3.5.1 Bosquejo de pantalla para iniciar sesión

La siguiente pantalla mostrara en la parte superior del lado izquierdo el Logo, Compañía, debajo se encontrara la hoja de trabajo donde estarán los campos necesarios para la bitácora. En el menú tendrá bitácora de atención a clientes,



Modificación de registros, Reportes, Cerrar Sesión y el número de visitante. Ver figura 3.5.2



Figura 3.5.2 Bosquejo de pantalla principal

En la parte central de esta página web se van a encontrar los campos necesarios para atender oportunamente y con el personal adecuado los datos que el usuario proporcione. Los datos que va proporcionar son: Ingreso de codificaciones con el respectivo campo de DN, ciclo de facturación, encuesta, encuestas disponibles, codificación y el campo se solucionó la petición. En la parte inferior se tendrá la opción de Agendar callback(regreso de llamada) al cliente, con los campos de aplica callback, motivo de callback, fecha de callback, comentarios y Guardar información(Botón de enviar). Ver figura 3.5.3



AGENDAR CLIENTE

Aplica Callback

Motivo del Callback

Fecha de Callback

Comentarios

Guardar información

Figura 3.5.3 Pantalla para agendar callback al cliente

En la pantalla de modificación de caso se mostrara la consulta a los # registros, los cuales contendrán los campos DN, motivo, fecha, estatus, ID y ejecutivo. Ejemplo Ver figura 3.5.4

DN	MOTIVO	FECHA	ESTATUS	ID	EJECUTIVO

Figura 3.5.4 Modificación de registros

En la pantalla de estatus de la base de datos se presentara la fecha de inicio, fecha fin, estatus, plataforma y el botón de Exportar Base.



Estatus de base

Servicio: Callcenter

▶ Exportar Base

▶ Fecha de inicio Fecha Fin Estatus

▶ Plataforma

▶ Exportar Base

Figura 3.5.5 Estatus de base

Capítulo 4 Análisis, diseño y construcción de la aplicación.

4.1 Modelo esencial ambiental del sistema.

Este modelo se conoce como el modelo ambiental, se necesita saber qué información entra al sistema desde el ambiente exterior, y qué información produce como salida al ambiente externo.

Otro aspecto del modelo ambiental consiste en identificar los acontecimientos que ocurren en el ambiente al cual debe responder el sistema (Sólo nos preocupan aquellos que ocurren en el ambiente exterior y requieren una respuesta del sistema). El modelo ambiental consta de tres componentes:

1.- Declaración de propósitos.

Es la declaración textual breve y concisa del propósito del sistema, dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema.

El propósito del sistema de bitácora de atención a clientes de telefonía móvil, es proporcionar un servicio eficiente y eficaz para disminuir tiempos de atención brindando un servicio de calidad que cumpla con las exigencias del usuario, realizando registros de los incidentes de cada llamada guardados en una base de datos para hacer una base de conocimientos y obtener estadísticas de control del call center.

2.- Lista de acontecimientos

Es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder.

- Un cliente realiza una llamada (F)
- Un cliente pide un servicio (F)
- El agente realiza un reporte (T)



- El agente resuelve o canaliza el incidente (C)
- El incidente se resuelve (F)

F,T,C. -flujo, temporal, o de control

El orientado a flujos es el que se asocia con un flujo de datos; es decir, el sistema se da cuenta de que ha ocurrido el acontecimiento cuando llega algún dato(s).

Los acontecimientos temporales arrancan con la llegada de un momento dado en el tiempo

Los acontecimientos de control deben considerarse un caso especial del acontecimiento temporal: un estímulo externo que ocurre en algún momento impredecible.

A diferencia de un acontecimiento temporal normal, el acontecimiento de control no se asocia con el paso regular del tiempo, por lo que el sistema no puede anticiparlo utilizando un reloj interno.

3.- Diagrama de contexto.

Es un caso especial de diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema.

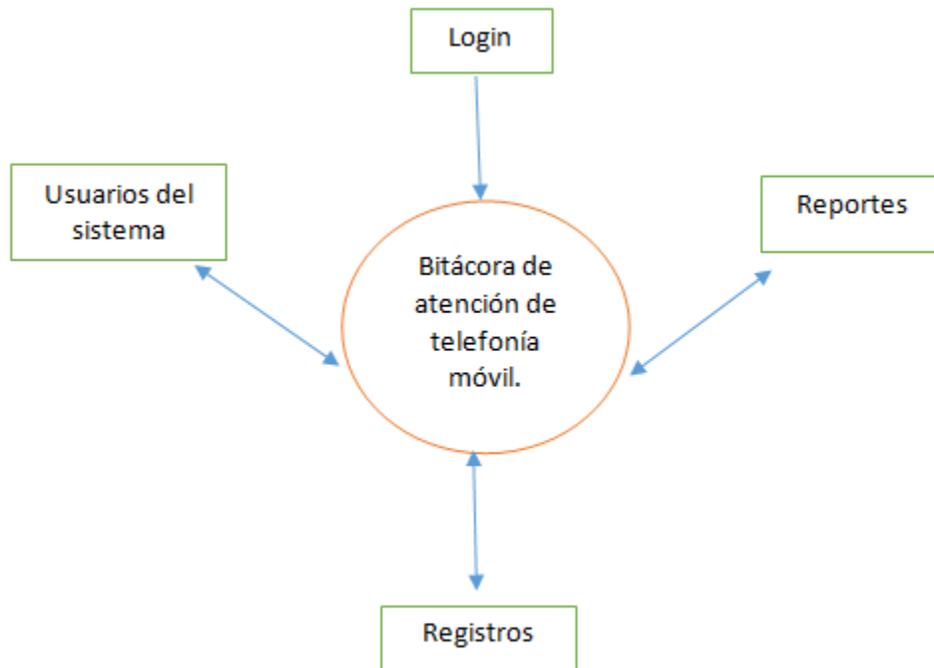


Figura 4.1.1.1 Diagrama de contexto.

En la figura 4.1.1.1 se muestra el diagrama de contexto del sistema de bitácora de atención de telefonía móvil, donde se observa de manera general como el sistema se relaciona con los procesos, a continuación se describe cada uno de ellos.

Login: El sistema solicita un nombre de usuario y contraseña proporcionado por el administrador del sistema para cada usuario para iniciar sesión en el sistema.

Usuarios del sistema: existirán dos tipos de usuarios, administrador y agente, el sistema ingresa a cada perfil según su usuario y contraseña.

Registros: El sistema recibe registros de datos de usuarios que contactan al call center, así como los eventos de cada llamada.



Reportes: Proporcionan información del sistema de manera organizada.

4.1.2 Diagrama de Entidad – Relación del sistema

El modelo entidad relación (E/R) proporciona una herramienta para representar información del mundo real a nivel conceptual.

Elementos del modelo entidad-relación

Entidad.

Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

Estas entidades se representan en un diagrama con rectángulos.

Atributos

Los atributos definen o identifican las características de la entidad (es el contenido de esta entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

Los atributos se representan solo como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos.

Relación.

Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.



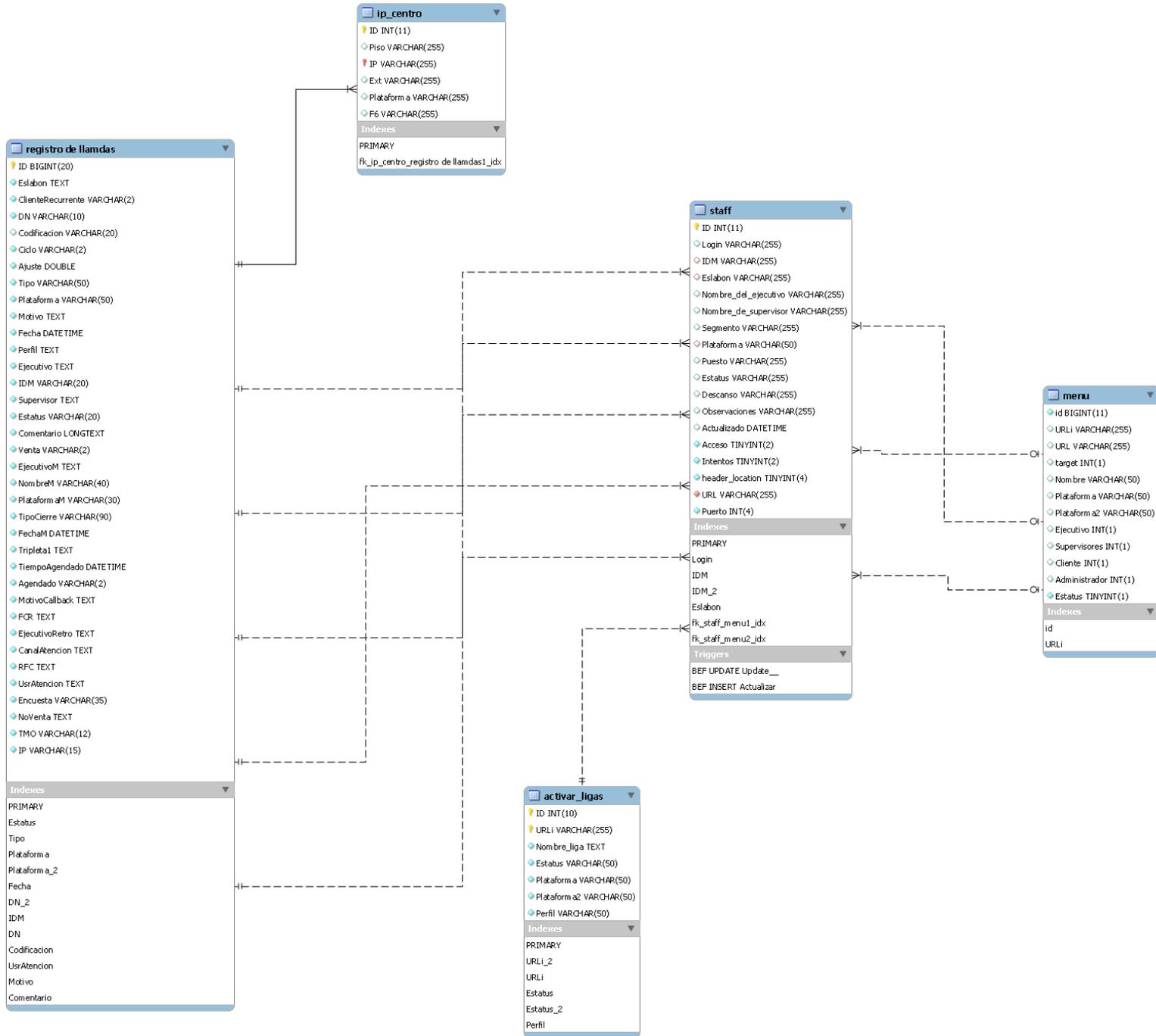
Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.

Restricciones del modelado.

- Los atributos pertenecen a las entidades o a las relaciones.
- Nombres únicos para las relaciones y las entidades dentro del esquema.
- Nombres únicos para los atributos dentro de una entidad o relación, no necesariamente dentro del esquema.
- Las relaciones deben darse entre al menos dos conjuntos de entidades aunque no necesariamente distintos.
- El nombre de un rol, debe ser único y distinto tanto de la entidad como de la relación.
- La llave identifica una entidad dentro del conjunto de entidades.

Pasos para la construcción del Diagrama Entidad Relaciones

- Identificar las entidades
- Determinar las claves primarias
- Describir los atributos de las entidades
- Establecer relaciones entre las entidades
- Dibujar el modelo de datos
- Realizar comprobaciones



En la figura 4.1.2.1 se muestra el diagrama entidad – Relación modelado para el sistema bitácora de atención de telefonía móvil, donde se observa la relación de las tablas las cuales se describirán en el diccionario de datos.

4.1.3 Diagrama de casos de uso, clases y de procesos.

4.1.3.1 Diagrama de casos de uso.

Representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen *qué* es lo que debe hacer el sistema, pero no *cómo*.

- Caso de uso

Un *caso de uso* describe, desde el punto de vista de los actores, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema.



El caso de uso está representado como lo muestra la siguiente Figura 4.1.3.1

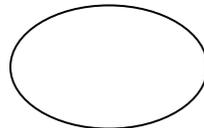


Figura 4.1.3.1 - Caso de uso

Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

- Cada caso de uso está relacionado como mínimo con un actor
- Cada caso de uso es un iniciador (es decir, un actor)
- Cada caso de uso lleva a un resultado relevante (un resultado con «valor intrínseco»)

Los casos de uso pueden tener relaciones con otros casos de uso.

Las relaciones más comunes entre casos de uso son:

- Asociación: Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple representada por Figura 4.1.3.2



Figura 4.1.3.2 Asociación

- Dependencia: Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se



denota con una flecha punteada representada por la Figura 4.1.3.3



Figura 4.1.3.3- Dependencias

- Generalización: Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>). Dicha relación se denota con una flecha simple representada por Figura 4.1.3.4



Figura 4.1.3.4 – Generalización

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

- Actor

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

Los actores no representan a personas físicas o a sistemas, sino su rol. Vea figura 4.1.3.4.5

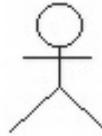


Figura 4.1.3.4.5- Actor

- Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

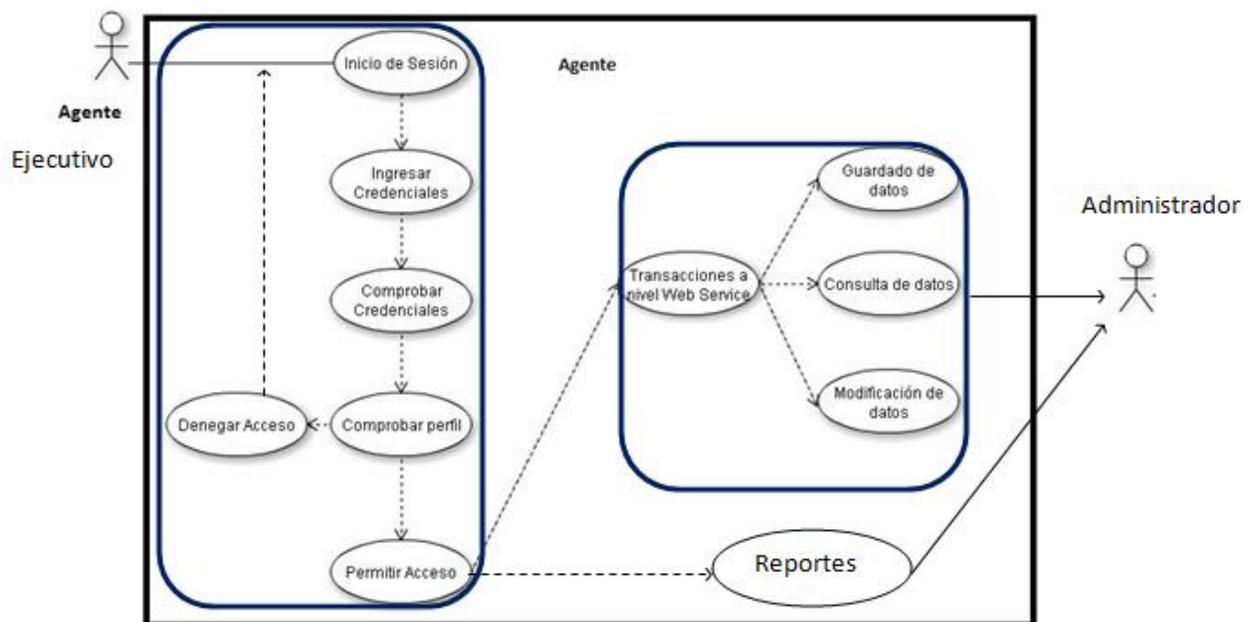


Figura 4.1.3.6 Diagrama de casos de uso



De acuerdo a la figura 4.1.3.6 El agente inicia sesión en el sistema, ingresa sus credenciales usuario y contraseña, el sistema valida, si son correctas se ingresa al aplicativo con el perfil asociado, si son incorrectos los datos se vuelve a solicitar usuario y contraseña, una vez permitido el acceso se está dentro de la bitácora donde podemos realizar las acciones de guardado de datos, consulta de datos (la cual permitirá asesorarse con la información asentada) y la modificación de los datos.

- Inicio de sesión: se ingresa a la aplicación donde en su pantalla principal se muestra la pantalla donde se solicita usuario y contraseña de inicio de sesión.
- Ingresar Credenciales: se ingresan los datos de usuario y contraseña proporcionados por el administrador del sistema.
- Comprobar credenciales: se valida que tanto usuario y contraseña sean correctos.
- Comprobar perfil: se asocia usuario y contraseña al perfil que asigno el administrador.
- Denegar acceso: los datos ingresados de usuario y/o contraseña no son correctos para ingresar al sistema.
- Se reinicia ciclo nuevamente.
- Permitir acceso: Usuario y contraseña corresponden a un perfil asociado al sistema.
- transacciones a nivel web service: se muestra la pantalla principal de la bitácora.
- Guardado de datos: se guardan los datos capturados por el agente durante la llamada.
- Consulta de datos: se muestran los registros existentes en el sistema.
- Modificación de datos: se modifican datos existentes para mantener la información actualizada.



4.1.3.2 Diagrama de clases.

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas estáticos porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas.

Cada clase se representa en un rectángulo con tres compartimientos como se muestra en la figura 4.1.3.7:

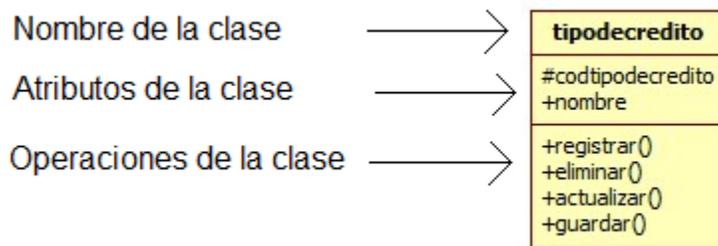


Figura 4.1.3.7 Diagrama de clase

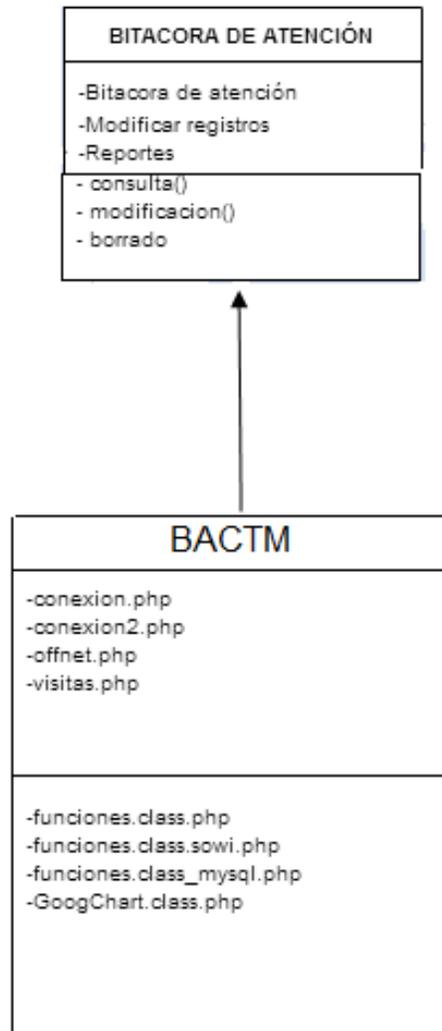


Figura 4.1.3.8 Bitácora

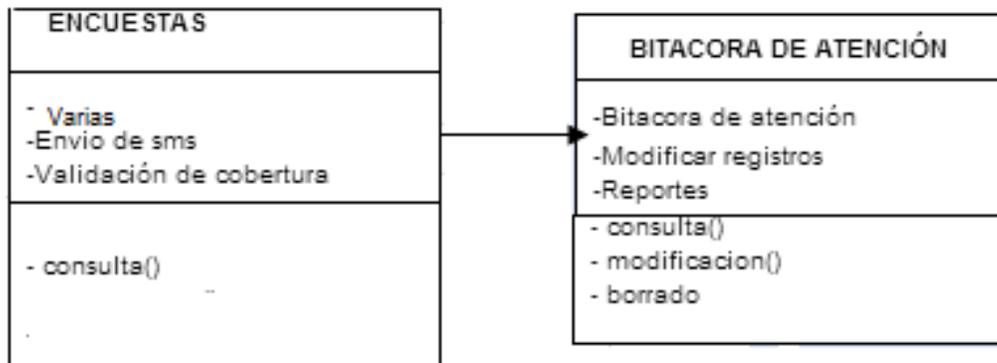


Figura 4.1.3.9 Encuestas

4.1.3.3 Diagrama de procesos

Estos diagramas representan la funcionalidad que se espera sea solucionada por el sistema.

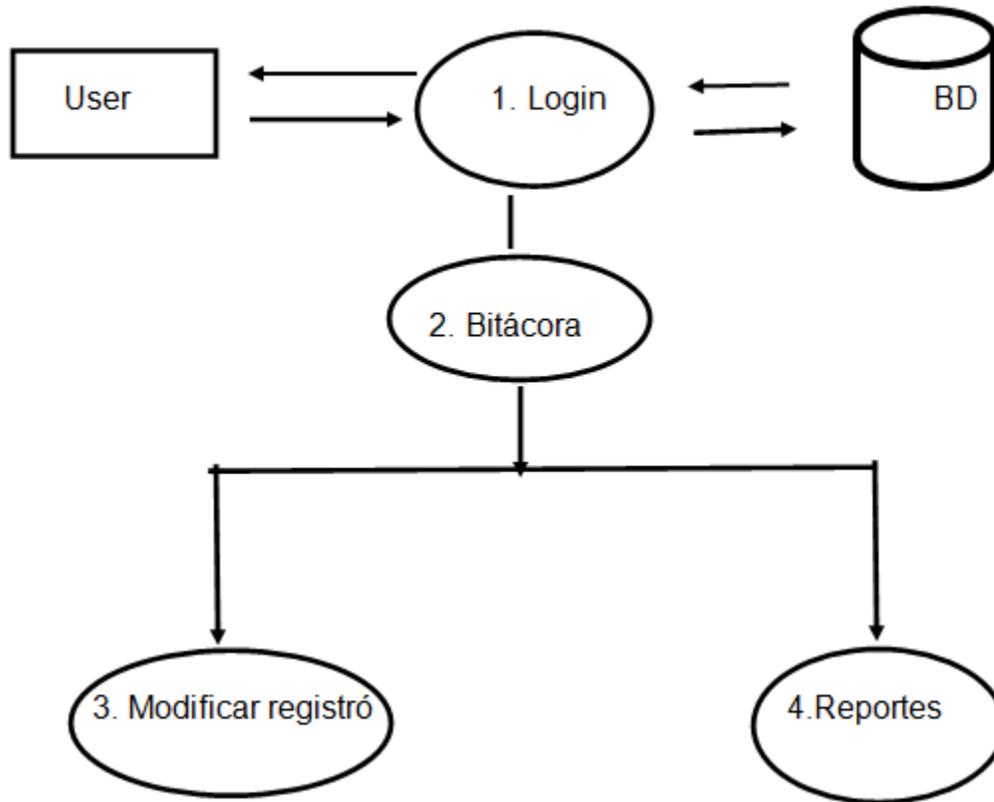


Figura 4.1.3.9 Diagrama de procesos general

En este diagrama se muestra de manera general los procesos del sistema de bitácora de atención a clientes de telefonía móvil el cual cuenta con 4 procesos, login, bitácora, modificar registro y reportes como se muestra en la figura 4.1.3.9 los cuales se describen a continuación.

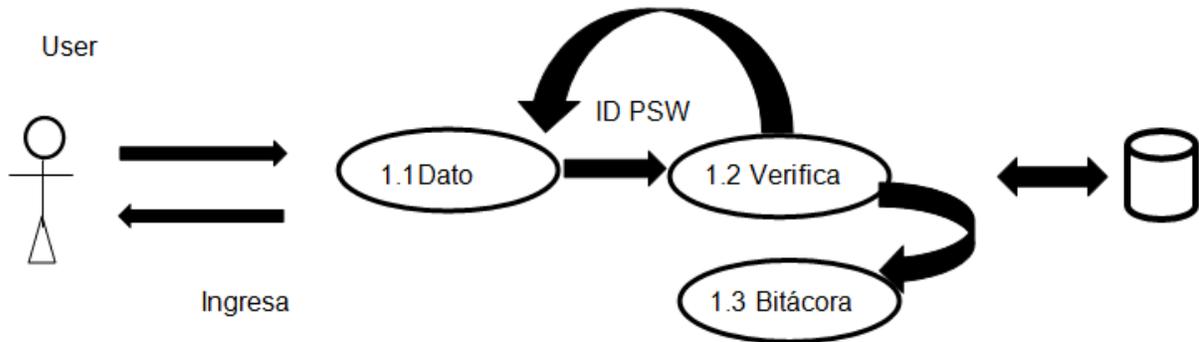


Figura 4.1.3.10 diagrama de proceso Login

En este proceso de acceso al sistema se realizan dos actividades **validación** (se determina si los datos de usuario y contraseña coinciden con lo establecido por el administrador) y **acceso** (para usuarios autorizados se obtiene acceso de acuerdo al perfil y asignado por el administrador) como se muestra en la figura 4.1.3.10

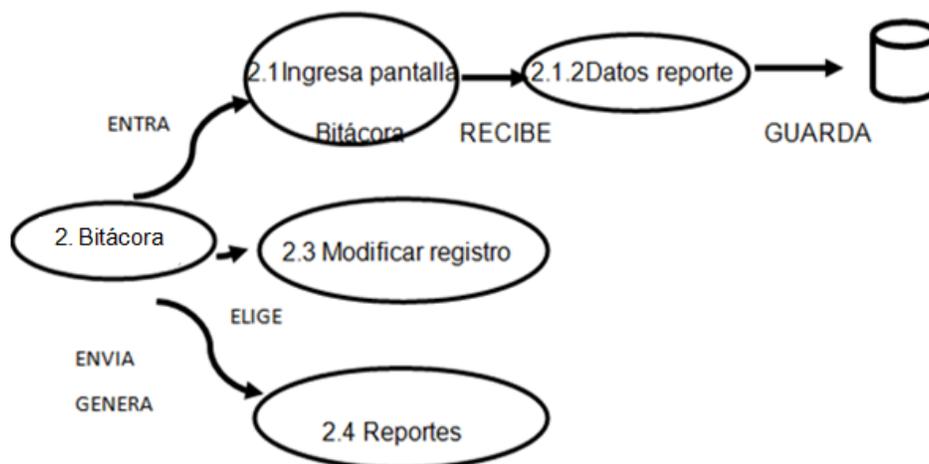


Figura 4.1.3.11 Diagrama de proceso bitácora

La figura 4.1.3.11 nos representa el proceso bitácora. Este proceso inicia dentro del sistema y nos permite realizar diversas tareas como es el llenado del formulario de bitácora y el acceso a los menús modificar registros y reportes.

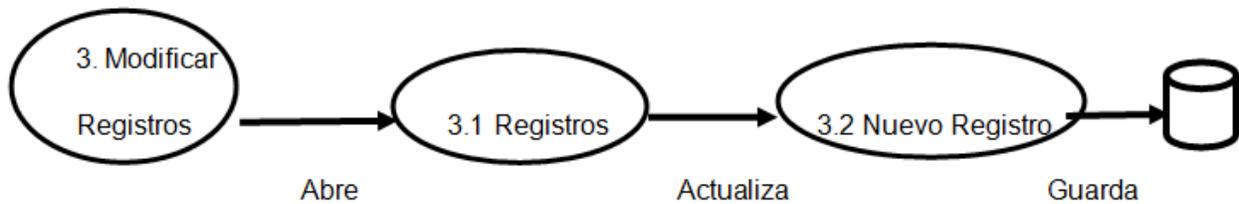


Figura 4.1.3.12 Diagrama de proceso modificar registro.

En el proceso modificar reportes que se muestra en la figura 4.1.3.12 se modifican reportes generados ya sea para actualizar datos o hacer algún cambio en el status del reporte.

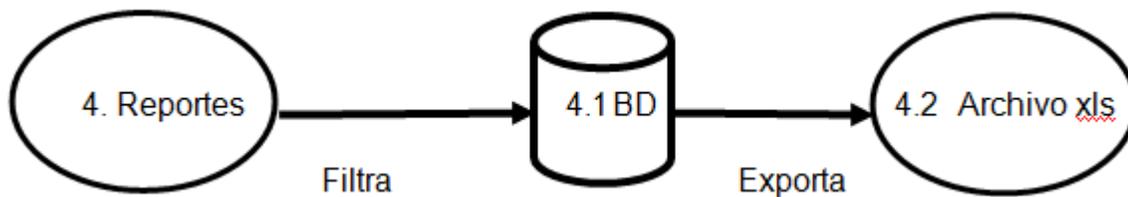


Figura 4.1.3.13 diagrama de proceso reportes

En la sección Reportes del diagrama que se muestra en la figura 4.1.3.13 la aplicación nos permite obtener información organizada para obtener estadísticas de las actividades del call center.



4.1.4 Diccionario de datos.

El diccionario de datos es un listado organizado de todos los datos que pertenecen a un sistema. Su objetivo es dar precisión sobre la información que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades.

A continuación se muestra el diccionario de datos conformado por las tablas correspondientes vea tablas 4.1.4.1 hasta 4.1.4.5

Tabla staf

Nombre Original	Acrónimo	Tipo	Longitud	Nulos	Key P/F	Tabla con que se relaciona	Descripción
ID	ID_STAF	Int	11	NO	P	Registro de llamadas	Identificador de la tabla staf
Login	Log_STAF	Varchar	255	YES			Usuario y contraseña de acceso
IDM	IDM_STAF	Varchar	255	YES	F	Registro de llamadas	Usuario que entra a la aplicacion
Eslabon	Eslabon	Varchar	255	YES	F	Registro de llamadas	Numero de empleado
Nombre_del_ejecutivo	Nom_ejec	Varchar	255	YES			Nombre del ejecutivo
Nombre_de_supervisor	Nom_sup	Varchar	255	YES			Nombre del supervisor
Segmento	segmento	varchar	255	YES			Segmento al que pertenece
Plataforma	plataforma	Varchar	50	YES			Plataforma a la que pertenece
Puesto	puesto	Varchar	255	YES			Puesto del empleado
Estatus	Estatus	Varchar	255	YES			Estatus del reporte
Descanso	Descanso	Varchar	255	YES			Descanso del empleado



Observaciones	Observaciones	Varchar	255	YES	Observaciones del incidente
Actualizado	Actualizado	datetime		YES	Tiempo de actualización
Acceso	Acceso	Tinyint	2	NO	Ingreso al sistema
Intentos	Intentos	Tinyint	2	NO	Numero de intentos
header_location	header_loc	Tinyint	4	NO	
URL	url_staf	Varchar	255	NO	Liga de staf
Puerto	puerto	Int	4	NO	Puerto

Tabla 4.1.4.1

Tabla Ip centro

Nombre Original	Acrónimo	Tipo	Longitud	Nulos	Key y P/F	Tabla con que se relaciona	Descripción
ID	Id_centro	int	11		P	Registro de llamadas	Identificador de la tabla ip centro
Piso	Piso_centro	varchar	255				Piso del edificio
IP	Ip_centro	varchar	255		P	Registro de llamadas	Ip de del cliente
Ext	Ext_centro	varchar	255				Extensión marcada
Plataforma	plataforma	varchar	255				plataforma
F6	F6_centro	varchar	255				Actualiza

Tabla 4.1.4.2



Tabla activar ligas.

Nombre Original	Acrónimo	Tipo	Longitud	Nulos	Key P/F	Tabla con que se relaciona	Descripción
ID	Id_ligas	int	10	NO	P	staf	Identificador de la tabla activar liga
URLi	Ur_ligas	varchar	255	NO	P	staf	Ruta de la URL
Nombre_liga	Nom_ligas	text		NO			Nombre de la liga
Estatus	estatus	varchar	50	NO	F	staf	Estatus de la liga
Plataforma	plataforma	varchar	50	NO			Nombre de la plataforma
Plataforma2	Plataforma2	varchar	50	NO			Nombre de la plataforma
Perfil	Perfil_ligas	varchar	50	NO			Perfil de inicio

Tabla 4.1.4.3

Tabla registro de llamada.

Nombre Original	Acrónimo	Tipo	Longitud	Nulos	Key P/F	Tabla con que se relaciona	Descripción
ID	Id_llamada	Bigint	20	NO	P	lp centro	auto_increment
ClienteRecurrente	Client_rec	Varchar	2	NO			Si el cliente es recurrente
DN	Dn_llamada	Varchar	10	NO	F	staf	Dn cliente
Codificacion	codificacion	Varchar	20	YES	F	staf	
Ciclo	Ciclo_llamada	Varchar	2	NO			Ciclo de facturación
Ajuste	Ajuste_llamada	Double		NO			Ajuste de pago
Tipo	Tipo_llamada	varchar(50)	50	NO	F	staf	Tipo de llamada
Plataforma	plataforma	varchar(50)	50	NO	F	staf	Nombre de la plataforma
Motivo	Motivo_llamada	Text		NO	F	staf	Motivo de llamada
Fecha	Fecha_llamada	datetime		NO	F	staf	Fecha de llamada



Perfil	Perfil_llamada	Text		NO			Perfil de usuario
Ejecutivo	ejecutivo	Text		NO			Perfil de ejecutivo
IDM	IDM_llamada	Varchar	20	NO	F	staf	Usuario
Supervisor	Supervisor	Text		NO			Perfil de supervisor
Estatus	Estatus_llamada	Varchar	20	NO	F	staf	Estatus de llamada
Comentario	Comentario	longtext		NO	F	staf	Comentario de la llamada
Venta	Venta_llamada	varchar	2	NO			Venta del agente
EjecutivoM	EjecutivoM	text		NO			Perfil de ejecutivo
NombreM	NombreM	varchar	40	NO			Nombre del agente
PlataformaM	PlataformaM	Varchar	30	NO			Plataforma
TipoCierre	TipoCierre	Varchar	90	NO			Tipo de cierre de la llamada
FechaM	FechaM	datetime		NO			Fecha de llamada
Tripleta1	Tripleta1	Text		NO			Tripleta
TiempoAgendado	TiempoAgendado	datetime		NO			Regresar llamada
Agendado	Agendado	Varchar	2	NO			Agendar llamada
MotivoCallback	MotivoCallback	Text		NO			Motivo de Callback
FCR	FCR_llamada	Text		NO			
EjecutivoRetro	EjecutivoRetro	Text		NO			
CanalAtencion	CanalAtencion	Text		NO			Canal de atención
RFC	RFC_llamada	Text		NO			RFC cliente
UsrAtencion	UsrAtencion	Text		NO	F	staf	Comentario de la llamada
Encuesta	Encuesta	Varchar	35	NO			Acepta encuesta
NoVenta	NoVenta	Text		NO			No se realiza la venta
TMO	TMO_llamada	Varchar	12	NO			Tiempo de llamada
IP	IP_llamada	Varchar	15	NO			Ip de conexión a la plataforma

Tabla 4.1.4.4



Tabla menú.

Nombre Original	Acrónimo	Tipo	Longitud	Nulos	Key P/F	Tabla con que se relaciona	Descripción
id	Id_menu	bigint	11	NO	p	Staf	Identificador de la tabla menú
URLi	Urli_men u	text		YES			Ruta de la URL
URL	Url_menu	text		YES			Liga de staf
target	Target	int(1)	1	YES			Indica como abrir el hipervinculo
Plataforma		Varchar	50	YES			Nombre de la plataforma
Plataforma2		Varchar	50	YES			Nombre de la plataforma
Ejecutivo		Int	1	YES			Selección de perfil
Supervisores		Int	1	YES			Selección de perfil
Cliente		Int	1	YES			Selección de perfil
Administrador		int	1	YES			Selección de perfil
Estatus		tinyint	1	NO			Estatus del reporte

Tabla 4.1.4.5

4.1.5 Normalización de la base de datos.

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas



diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Los datos redundantes desperdician el espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar de la misma forma exactamente en todas sus ubicaciones. Un cambio en la dirección de un cliente es mucho más fácil de implementar si los datos sólo se almacenan en la tabla Clientes y no en algún otro lugar de la base de datos.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

Formas Normales.

Las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos.

Primera Forma Normal (1FN)

Sea un conjunto de atributo perteneciente (\in) a la relación R, en donde R está en la Primera Forma Normal si todos los atributos $\alpha[n]$ son atómicos, es decir no pueden seguir dividiéndose.

- Elimine los grupos repetidos de las tablas individuales.
- Cree una tabla independiente para cada conjunto de datos relacionados.
- Identifique cada conjunto de datos relacionados con una clave principal.
-



Segunda Forma Normal (2FN)

Dependencia completa. Está en 2FN si esta en 1FN y si sus atributos no principales dependen de forma completa de la clave principal.

- Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
- Relacione estas tablas con una clave externa.

Tercera Forma Normal (3FN)

Está en segunda forma normal y todo atributo no primo es implicado por la clave primaria en una secuencia no transitiva. Se eliminan las dependencias transitivas.

- Elimine los campos que no dependan de la clave.

A continuación se realizara el análisis de normalización de la base de datos tomando como muestra algunos datos obtenidos. Ver Tabla 4.1.5.1

ID	LOGIN	IDM	ESLABON	NOMBRE DE EJECUTIVO	NOMBE DE SUPERVISOR	SEGMENTO	PLATAFORMA
1	1acfc3a3de9 64a0975ebb 71169b295d0	crojas	4210537	ROJAS PONCE CHRISTIAN	OROZCO MORALES JORGE SAID	ADMINISTRADOR	ADMINISTRADOR
2	106a797daa 0ab6e0f436c aac3d2d3b3e	cmez a	*****	MEZA MELENDEZ CARLOS ALBERTO	CESAR EDUARDO ALARCON CASTILLO	Contrato	Miscelaneas
3	74dc6f70d84 1e4f24cb ed4 1eb4baa72e	jsaid	4212107	OROZCO MORALES JORGE SAID	CARRASCO IRINEO JOSE CRISTIAN	Contrato	APV
4	a99d5a65e8 3dae2c9	jvalenzuela	4213326	VALENZUELA MORAN JOSE YADIR	OROZCO MORALES JORGE SAID	Contrato	Contrato Básico



	c3b7 5bf08136 99c						
5	49f3e5dc 9b9 92263a4f 567 a98a79f1 7a	esjuar ez	4205831	JUAREZ SAUCILLO DIANA ESMERALDA	OROZCO MORALES JORGE SAID	Contrato	Miscelaneas
6	5236a75 610 668f3712 9ba a471c6fc e25	MRT0 0415	MRT00415	MARTIN ENRIQUE MONTIEL GONZALEZ	OSCAR ALBERTO AGUILERA FERNANDEZ	Contrato	ADMINISTRAD OR
7	f1766134 bc6 2224636 890d 68928b7 7ab	ltorres	4212534	TORRES DOMINGUES LUIS ANGEL	VALENZUELA MORAN JOSE YADIR	Contrato	Contrato Basico
8	acdfa2cd 606 e5f3f15d	nfranc o	56664	FRANCO GONZALEZ NANCY	OROZCO MORALES JORGE SAID	Contrato	Contrato Basico



	b8e 86ab0cbf a5			SELENE			
9	ecac2cef 10d 118db20 be4e d1e835af 6a	mlara	4209829	MANILALLI ANGELES LARA HERNANDEZ	JORGE ALBERTO FLORES	Contrato	Contrato Basico
10	904db5c ad8 b520013 981 66be28e 5bb9d	cmira nda	4210527	CARLOS MIRANDA GONZALEZ	JUAN ANTONIO MORALES	Contrato	Contrato Basico
11	91f98816 0d0 ab3ebc3 0666 33ce1bb 132	jcarr sco	jcarrasco	CARRASCO IRINEO JOSE CRISTIAN	CARRASCO IRINEO JOSE CRISTIAN	Contrato	ADMINISTRAD OR
12	418f50bc 1d0 809cf9b6	Prueb a	Prueba	PRUEBA	ROJAS PONCE CHRISTIAN	Contrato	FO Prepago



	84b ada6f29f 02						
13	7145b2a c17 6d3f9d19 ec1 1c5b66c 7e83	Earriga	orgasmatron	ERIC NOE ARRIAGA CARRILLO	OROZCO MORALES JORGE SAID	Contrato	ADMINISTRADOR
14	0caa3cf4 db5 d680018 416b 3a37be6f 94	AXR2 9002	197707	FELIPE RAMIREZ GUADALUPE JAQUELINE	AMBROSIO GARCIA EDGARDO	Contrato	Contrato Basico
15	375cb63 ec82 4df3f514 6cb9 5129038 44	AXR1 5640	4211803	FUENTES HERNANDEZ ROCIO CARMINA	EUGENIO GUTIERREZ TAVERA	Contrato	Contrato Basico

Tabla 4.1.5.1 Datos Muestra

De la primera forma Normal obtenemos la Tabla 4.1.5.2

ID	LOGIN	IDM	ESLABON	NOMBRE DE EJECUTIVO
1	1acfc3a3de964a0975ebb7 1169b295d0	crojas	4210537	ROJAS PONCE CHRISTIAN
2	106a797daa0ab6e0f436ca ac3d2d3b3e	cmeza	*****	MEZA MELENDEZ CARLOS ALBERTO
3	74dc6f70d841e4f24cbcd41 eb4baa72e	jsaid	4212107	OROZCO MORALES JORGE SAID
4	a99d5a65e83dae2c9c3b7 5bf0813699c	jvalenzuela	4213326	VALENZUELA MORAN JOSE YADIR
5	49f3e5dc9b992263a4f567 a98a79f17a	esjuarez	4205831	JUAREZ SAUCILLO DIANA ESMERALDA
6	5236a75610668f37129baa 471c6fce25	MRT00415	MRT00415	MARTIN ENRIQUE MONTIEL GONZALEZ
7	f1766134bc62224636890d 68928b77ab	ltorres	4212534	TORRES DOMINGUES LUIS ANGEL
8	acdfa2cd606e5f3f15db8e8 6ab0cbfa5	nfranco	56664	FRANCO GONZALEZ NANCY SELENE
9	ecac2cef10d118db20be4e d1e835af6a	mlara	4209829	MANILALLI ANGELES LARA HERNANDEZ
10	904db5cad8b5200139816 6be28e5bb9d	cmiranda	4210527	CARLOS MIRANDA GONZALEZ
11	91f988160d0ab3ebc30666 33ce1bb132	jcarrasco	jcarrasco	CARRASCO IRINEO JOSE CRISTIAN
12	418f50bc1d0809cf9b684ba da6f29f02	Prueba	Prueba	PRUEBA
13	7145b2ac176d3f9d19ec11 c5b66c7e83	Earriaga	orgasmatron	ERIC NOE ARRIAGA CARRILLO
14	0caa3cf4db5d680018416b 3a37be6f94	AXR29002	197707	FELIPE RAMIREZ GUADALUPE JAQUELINE
15	375cb63ec824df3f5146cb9 512903844	AXR15640	4211803	FUENTES HERNANDEZ



ROCIO CARMINA

Tabla 4.1.5.2 primera forma normal

Aplicamos la segunda forma normal para quitar los datos duplicados de las columnas nombre de supervisor, segmento y plataforma y obtenemos las tablas.

Staf_id_sup, ver tabla 4.1.5.3

Staf_id_segmento, ver tabla 4.1.5.4

Staf_id_plataforma, ver tabla 4.1.5.5

ID_SUP	SUP
AXR00010	AMBROSIO GARCIA EDGARDO
AXR00003	CARRASCO IRINEO JOSE CRISTIAN
AXR00002	CESAR EDUARDO ALARCON CASTILLO
AXR00011	EUGENIO GUTIERREZ TAVERA
AXR00006	JORGE ALBERTO FLORES
AXR00007	JUAN ANTONIO MORALES
AXR00001	OROZCO MORALES JORGE SAID
AXR00004	OSCAR ALBERTO AGUILERA FERNANDEZ
AXR00009	ROJAS PONCE CHRISTIAN
AXR00005	VALENZUELA MORAN JOSE YADIR

Tabla 4.1.5.3 Staf_id_sup



ID_SEGMENTO	SEGMENTO
1	ADMINISTRADOR
2	Contrato

Tabla 4.1.5.4 Staf_id_segmento Staf_id_segmento.

ID PLATAFORMA	ID PLATAFORMA
1	ADMINISTRADOR
27	Miscelaneas
2	APV
17	Contrato Basico
22	FO Prepago

Tabla 4.1.5.5 Staf_id_plataforma

En este punto hemos terminado con nuestra normalización ya que no hay datos repetidos y cada columna en estas tablas se identifica de manera única, mostrando los datos en un esquema limpio y fácil de trabajar, por tanto ya nos necesario aplicar la tercera forma normal.

4.2 Creación de la base de datos

En esta sección se especifican los pasos a seguir para la descarga e instalación del manejador de base de datos MySQL.



Para la instalación:

XAMPP es un servidor independiente de la plataforma, de software libre, que consiste principalmente en la base de datos MySQL, un servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. También incluye otros módulos como OpenSSL y phpMyAdmin.

Para descargar el software XAMPP, es necesario ingresar al portal oficial de la aplicación. <https://www.apachefriends.org/es/> . Ver Figura 4.2.1

¿Qué es XAMPP?
XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MySQL, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Descargar
Pulsa aquí para otras versiones

XAMPP para Windows
v5.6.8 (PHP 5.6.8)

XAMPP para Linux
v5.6.8 (PHP 5.6.8)

XAMPP para OS X
v5.6.8 (PHP 5.6.8)

Figura 4.2.1 Portal de XAMPP.

Una vez descargado el paquete de instalación, el siguiente paso es hacer clic sobre el mismo para comenzar la instalación. Ver Figura 4.2.2

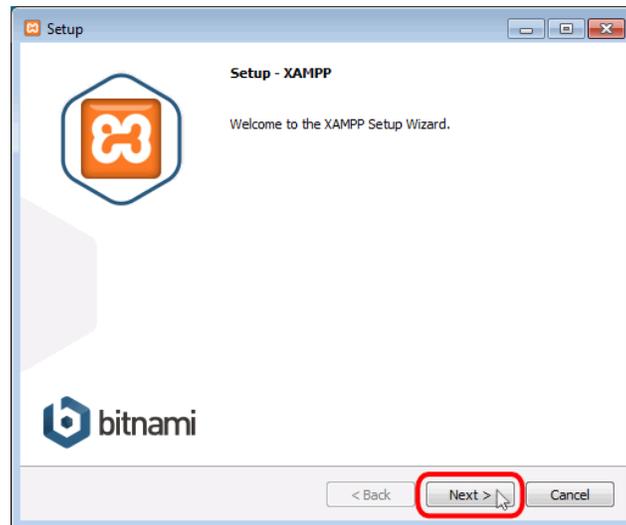


Figura 4.2.2 Instalación de XAMPP.

Una vez terminada la instalación, se mostrará la pantalla de confirmación, la cual dará paso a la configuración del servidor de base de datos.

El panel de control de XAMPP se divide en tres zonas:

- La zona de módulos, que indica para cada uno de los módulos de XAMPP: si está instalado como servicio, su nombre, el identificador de proceso, el puerto utilizado e incluye unos botones para iniciar y detener los procesos, administrarlos, editar los archivos de configuración y abrir los archivos de registro de actividad.
- La zona de notificación, en la que XAMPP informa del éxito o fracaso de las acciones realizadas
- La zona de utilidades, para acceder rápidamente. Ver Figura 4.2.3

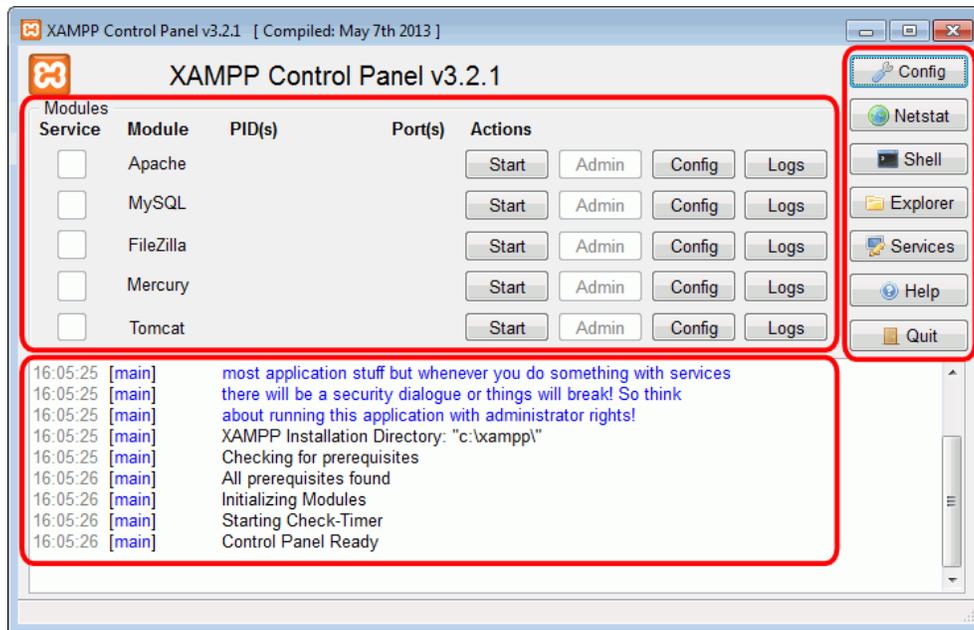


Figura 4.2.3 Levantar servicios

Para comprobar que todo funciona correctamente, hay que escribir en el navegador la dirección <http://localhost> Al abrir la página por primera vez, XAMPP pedirá seleccionar el idioma.

Una vez elegido el idioma, se mostrará el panel de administración web de XAMPP. Ver Figura 4.2.4

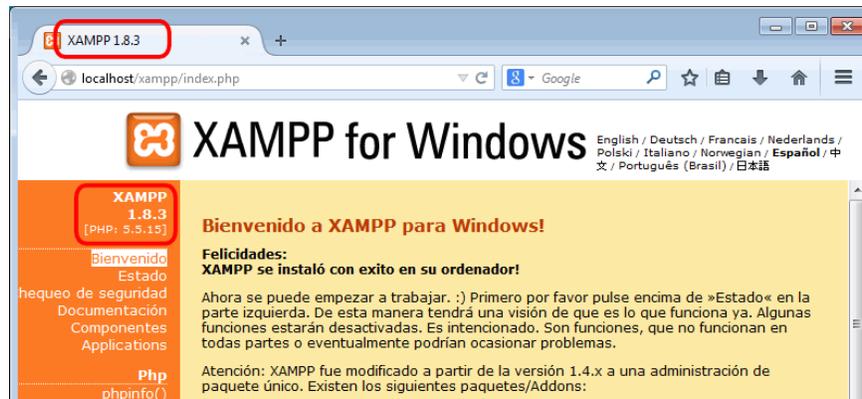


Figura 4.2.4 Menú de XAMPP

Creación de la una base de datos en phpMyAdmin. El nombre de la base de datos para este ejemplo es: BaseSQL. Ver Figura 4.2.5.



Figura 4.2.5 Creación de la base de datos

Creación de una tabla: Para la creación de una tabla en phpMyAdmin, seleccionar la base de datos. Seguidamente seleccionar el botón Crear tabla, para este ejemplo el nombre de la tabla Usuarios y el Número de columnas cinco. Ver Figura 4.2.6.

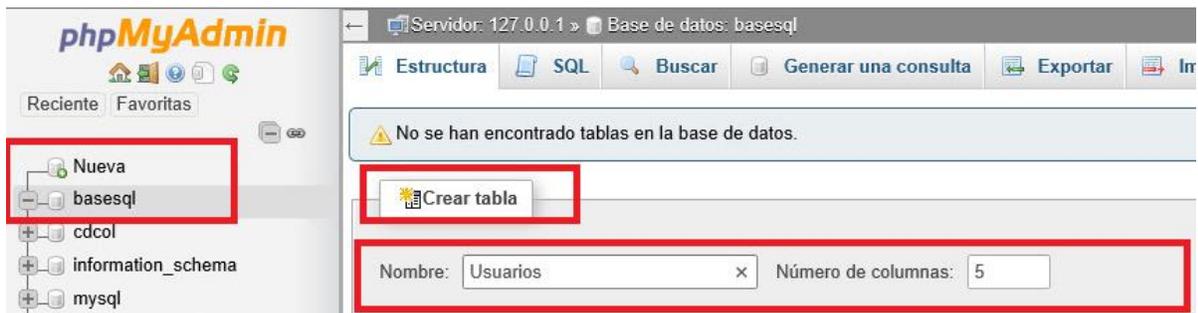


Figura 4.2.6 Creación de la tabla usuarios

Se muestran todas las tablas de la base de datos. La selección de tablas se muestra a continuación:

SELECT auditoria_navegacion_url, audotoria_navegacion, estadistico, menu, registrodellamdas, staff, tripleta, u_onlines. Ver Figura 4.2.7

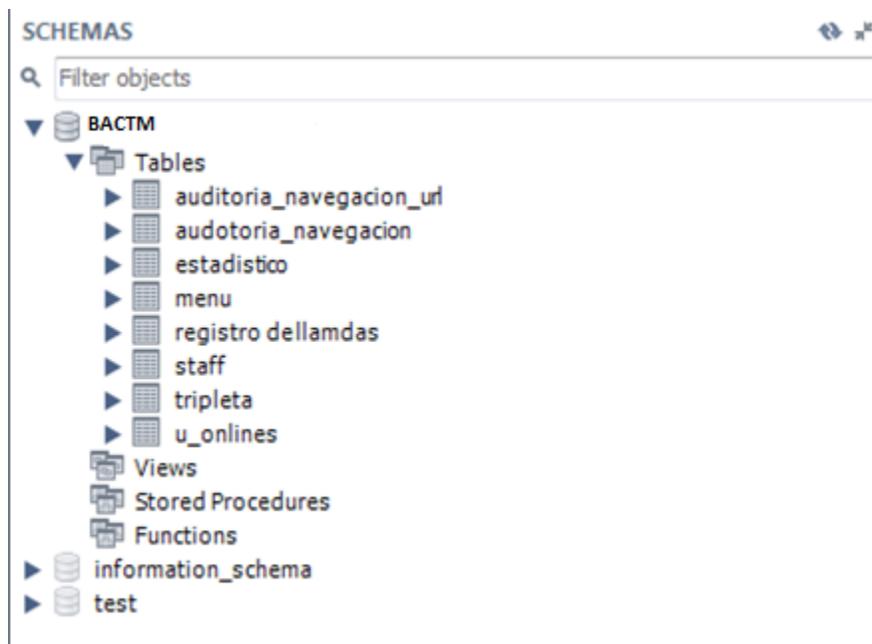


Figura 4.2.7 La tablas



Creación de un trigger(disparador): Para la creación de un trigger en phpMyAdmin, seleccionar el menú Más > Disparadores > Nuevo. Ver Figura 4.2.8.

Sintaxis básica de un trigger en SQL:

```
CREATE trigger Nombredeltrigger
```

```
on NombreTabla
```

```
for update
```



Figura 4.2.8 Creación de un trigger

Se muestran los triggers que utiliza la base de datos. Ver Figura 4.2.9

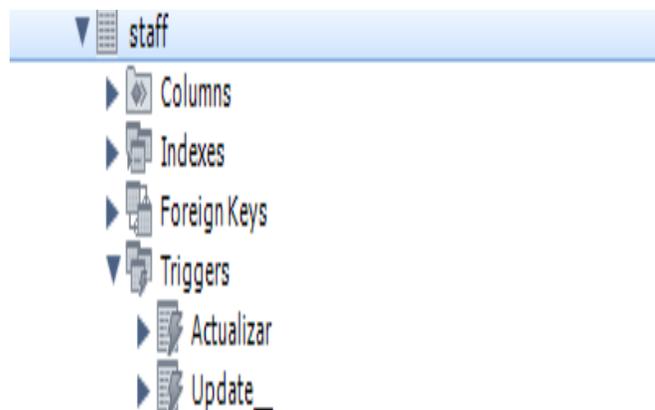


Figura 4.2.9 Los triggers



Crear un procedimiento almacenado: Para la creación de un procedimiento almacenado en phpMyAdmin, seleccionar el menú SQL > insertar el procedimiento almacenado Ver Figura 4.2.10

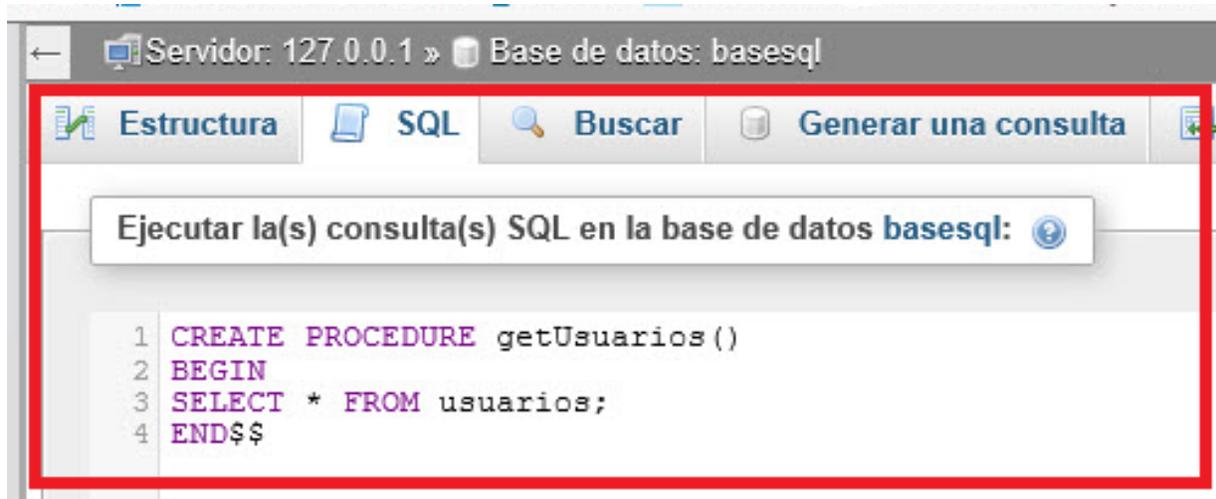


Figura 4.2.10 Creación del procedimiento almacenado “getUsuarios”

Creación de una consulta: Para la creación de una consulta en el menú SQL, se hará la consulta automática con el procedimiento almacenado. Para este ejemplo muestra los datos que contiene la tabla “Usuarios”. Ver Figura 4.2.11

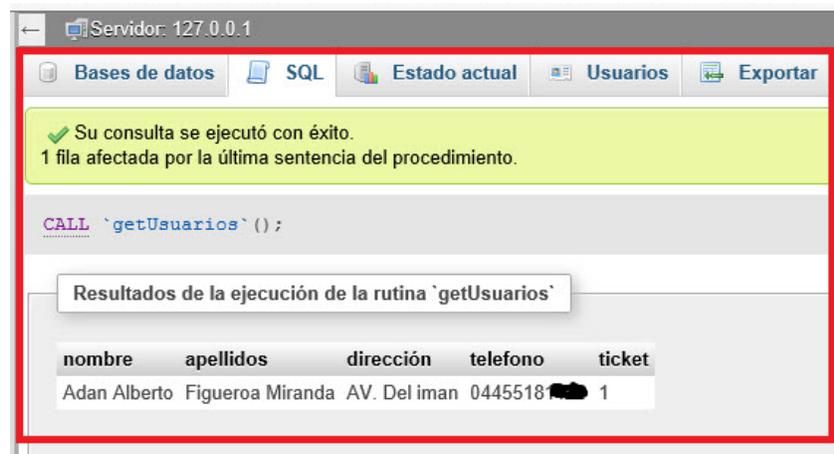
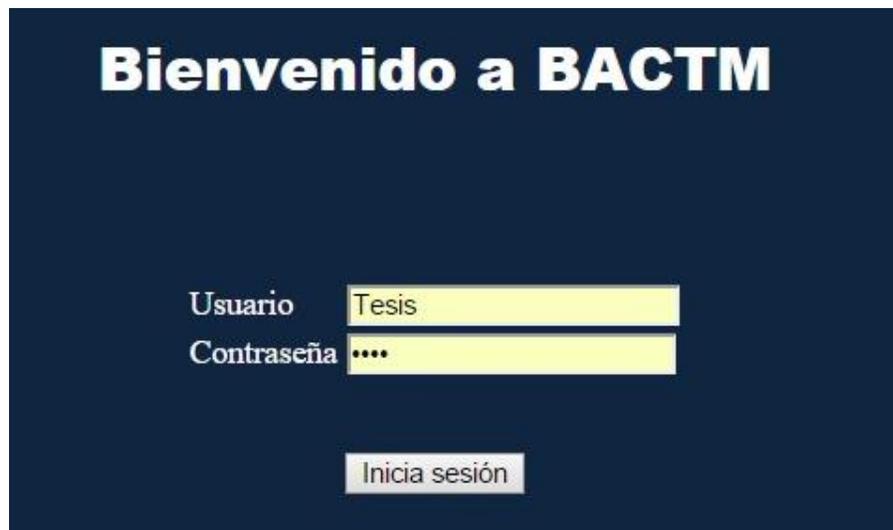


Figura 4.2.11 Consulta de la base de datos

4.3 Desarrollo del Front-end

Para el sistema desarrollado en esta tesis, el front end consiste en lo siguiente: Al inicio del sistema se presentan unos cuadros de texto donde se deben ingresar el usuario y la contraseña



The image shows a login interface with a dark blue background. At the top, the text "Bienvenido a BACTM" is displayed in white. Below this, there are two input fields. The first is labeled "Usuario" and contains the text "Tesis". The second is labeled "Contraseña" and contains four dots "****". Below the input fields is a button labeled "Inicia sesión".

Figura 4.3.1 Pantalla de inicio

Una vez ingresados los datos se muestra la siguiente pantalla donde se podrán ingresar los datos de la llamada que se está atendiendo



Figura 4.3.2 Pantalla inicial bitácora

En esta pantalla, del lado izquierdo se aprecia el menú de la aplicación que consiste en 4 opciones y una estadística ordenados de la siguiente manera:

- Bitácora de atención
- Modificar registros
- Reportes
- Regresar
- Cerrar sesión
- Visitas online



En la parte de bitácora nos mostrara nuestra pantalla de inicio que consiste en varios cuadros de texto que se llenaran con los datos del cliente, menú desplegable de encuesta de servicio, codificación que tipo de problema se presentó, si se solucionó o no el problema y por ultimo si es necesario regresar la llamada al cliente

Figura 4.3.2 Pantalla inicial bitácora parte 2

Al final encontramos el botón de enviar para guardar los datos en la base.



Bitacora de atención a cliente x Página principal - Dropbox x

https://127.0.0.1/Seguimientos/Modificar_DN.php

Bitacora
Regresar

2 visitantes online.
Ingresa todas tus Codificaciones en esta liga

Consulta de registro
Consulta aqui

Consultar

Modificación de caso

Acceso a Chat

Se generó una consulta con 10 registros

DN	Motivo	Fecha	Estatus	ID	Ejecutivo	Supervisor	Plataforma
5533958176	ConsultaMovimientos de cuenta-clienteBloqueo IMEInvo autenticado	2014-10-17 22:51:28	Seguimiento	4213064 4213064	PALACIOS VAZQUEZ AIDEE ALEJANDRA	OJEDA RODRIGUEZ MARIO ALEJANDRO	Migrados Contrato
2461399640	SolicitudMovimientos de CuentaClienteSuspension del servicio	2014-10-17 22:47:18	Seguimiento	4215586 4215586	LEON MUNGUJA JULIO IVAN	JUDITH CANALES REBOLLO	Migrados Contrato
4423011177	SolicitudMovimientos de CuentaClienteCambio de Plan AgendadoControlCiclo de Facturación 28	2014-10-17 22:42:34	Seguimiento	4214925 4214925	BAZAN BARRERA CRESCENCIO JAVIER	ESQUIVEL CATARINO KARLA ALEJANDRA	Migrados Contrato
3521220258	SolucionAtencionSoporte SOWIAjuste	2014-10-17 22:38:41	Seguimiento	4216633 4216633	GONZALEZ PIEDRA RAYMUNDO MISAEAL	TORRES DOMINGUES LUIS ANGEL	Migrados Contrato
3521220258	Ajustes	2014-10-17 22:38:41	Seguimiento	4216634 4216634	GONZALEZ PIEDRA RAYMUNDO MISAEAL	TORRES DOMINGUES LUIS ANGEL	Migrados Contrato
7226049073	TransferenciaLlamada no relacionadaTransferencia Rechazada	2014-10-17 22:36:35	Seguimiento	4216635 4216635	PALACIOS VAZQUEZ AIDEE ALEJANDRA	OJEDA RODRIGUEZ MARIO ALEJANDRO	Migrados Contrato
5527968781	SolicitudServicio TécnicoTarjeta SIMAsignación PIN Genérico	2014-10-17 22:36:18	Seguimiento	4216636 4216636	MANZANO ALARCON MARIA DEL CARMEN	ESQUIVEL CATARINO KARLA ALEJANDRA	Migrados Contrato
4421759286	SolicitudMovimientos de CuentaClienteNumeros FrecuentesAlta pendiente	2014-10-17 22:35:45	Seguimiento	4216637 4216637	LEON MUNGUJA JULIO IVAN	JUDITH CANALES REBOLLO	Migrados Contrato
6461402769	ConsultaProductos y ServiciosInformacion de Planes	2014-10-17 22:32:56	Seguimiento	4216638 4216638	ROSALES JUAREZ JORGE GERARDO	ESQUIVEL CATARINO KARLA ALEJANDRA	Migrados Contrato
	Productos y ServiciosSolicitud de baja	2014-10-17	Seguimiento	4216639	PALACIOS VAZQUEZ AIDEE	OJEDA RODRIGUEZ	Migrados

https://127.0.0.1/Seguimientos/index.php

Figura 4.3.3 Modificar Registró



Dentro de este menú encontramos todos los casos no resueltos que se identificaron por medio de la bitácora para que puedan ser revisados de manera más puntual.



Figura 4.3.4 Reportes

En este menú tenemos la opción de acotar la información de la base de datos mediante fechas y exportarlo a un Excel



Figura 4.3.5 Visitantes Online



Aquí se muestra el contador que se colocó dentro de la aplicación para saber cuántos ejecutivos tenemos conectados al sistema



Figura 4.3.6 Cierre de sesión

Esta opción del menú finaliza la sesión y nos regresa a la pantalla inicial para ingresar las credenciales.

4.4 Generación de pruebas y mantenimiento.

Fase de Pruebas.

Uno de los objetivos de la fase de pruebas del sistema es verificar que el comportamiento externo del sistema de software satisface los requisitos establecidos por los clientes y futuros usuarios del mismo. A medida que aumenta la complejidad de los sistemas de software y aumenta la demanda de calidad, se hacen necesarios procesos y métodos que permitan obtener buenos conjuntos de pruebas del sistema.

Dentro de esta fase pueden desarrollarse varios tipos distintos de pruebas en función de los objetivos de las mismas a continuación se describen algunos tipos de prueba.

Caja Negra: Son aquellas que se enfocan directamente en la función del módulo, sin importar el código. Este tipo de prueba se le aplica a los módulos para verificar que realizan la función para la cual fueron programados ver figura 4.3.1.a

Caja Blanca: Al contrario de las pruebas de caja Negra, estas se basan en el conocimiento de la lógica interna del código. Las pruebas contemplan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc., ver figura 4.3.1.b

Pruebas de Caja Negra y Caja Blanca

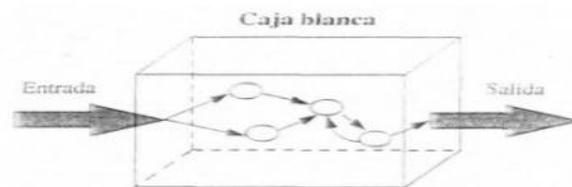


Figura 4.3.1.a

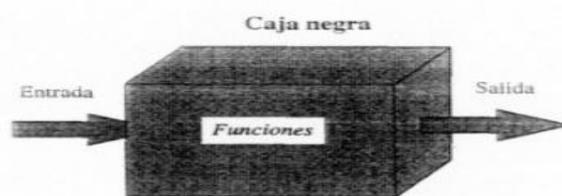


Figura 4.3.1.b

Integración: las pruebas de integración buscan probar la combinación de las distintas partes de la aplicación para determinar si funciona correctamente en conjunto.



La prueba de integración detecta errores de interacción se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y corregir.

Modulo (unitarias): El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y lo hará, generalmente, la persona que lo creó. Un módulo se entiende como un componente de software que cumple las siguientes características:

- Debe ser un bloque básico de construcción de programas.
- Debe implementar una función independiente y simple.
- Podrá ser probado al cien por cien por separado.
- No deberá tener más de 500 líneas de código.

Stress: Este tipo de prueba se realiza para determinar la solidez de la aplicación en momentos de carga extrema y ayuda a los administradores a determinar si la aplicación rendirá lo suficiente en caso de sobrecarga, se utilizan con el objetivo de romper la aplicación, o determinar el número esperado de usuarios en concurrencia para hacer una simulación lo más cercana posible a la realidad, también se prueban los servicios de aplicación y el rendimiento de cada componente que interactúa para verificar si hay degradación o comportamiento de alto consumo de recursos.

Estas pruebas permiten tomar decisiones sobre configuraciones de hardware, ajustes de software y selección de arquitecturas.

Alfa: se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando



los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.

Beta: se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.

Validación: Dentro de esta prueba debemos entender por validación a la verificación del funcionamiento del software de acuerdo a los requerimientos del cliente.

El resultado de esta prueba es crucial dado que si el software no cumple con las expectativas del cliente, todo el esfuerzo y trabajo no tendrán ningún significado y aunque pudiera verse como una prueba arbitraria no es así, ya que solamente si no hemos implementado los mecanismos de calidad necesarios en el proceso de desarrollo esta prueba no tendría que modificar aspectos estructurales del software sino solamente aspectos superficiales y de interfaces.

Regresión: Cada vez que se añade un módulo nuevo como parte de las pruebas de integración, el software cambia: se establecen nuevos caminos en el flujo de datos, pueden existir nuevas I/O, y se invoca a una nueva lógica de control, lo cual puede ocasionar problemas con funciones que ya trabajaban correctamente. Las pruebas de regresión consisten en volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente, para asegurarse que los cambios no han ocasionado efectos colaterales indeseados.

Por otra parte todos los sistemas sufren una evolución a lo largo de la vida activa.



En cada nueva versión se corrigen defectos o se añaden nuevas funciones, o ambas cosas. En cualquier caso, una nueva versión exige una nueva verificación. Si éstas se han sistematizado en una fase anterior, ahora pueden volver a realizarse automáticamente, simplemente para comprobar que las modificaciones no provocaron errores donde antes no los había. Por consiguiente, ésta prueba detectará fallas en algún módulo debido a modificaciones realizadas con anterioridad para comprobar que las modificaciones no provocaron errores donde antes no los había. Por consiguiente, ésta prueba detectará fallas en algún módulo debido a modificaciones realizadas con anterioridad.

Pruebas realizadas al sistema Bitácora de atención a clientes de telefonía móvil.

Se verifica la funcionalidad de cada módulo:

- Bitácora de atención.



Bitacora de Atencion

Modificar registros

Reportes

Regresar

Cerrar Sesión

2 visitantes online.
Ingresa todas tus codificaciones en esta liga

Ingreso de codificaciones

DN

No Codificación ¿Es llamada cortada? No ▾

Ciclo de facturación NC ▾

Encuesta FALLA DURANTE LA TRANSFERENCIA ▾

Encuestas Disponibles Selecciona... ▾

Selecciona una tripleta

Codificación

¿Se soluciono la petición? Sí No

Agendar Callback al cliente

Aplica Callback No ▾

Motivo de Callback Soporte a Movitalk ▾

Fecha del Callback 2015-06-24 Hora 03 ▾ : 06 ▾

Figura 4.3.2 pantalla inicial de la Bitácora.

Se inicia con el llenado de la bitácora, como se observa en la figura 4.3.4 se dejó vacío el campo No Codificación el cual es obligatorio.



Atencion
Modificar registros
Reportes
Regresar
Cerrar Sesión

2 visitantes online.
Ingresa todas tus codificaciones en esta liga

DN 5541766164

No Codificación
¿Es llamada cortada? No ▾

Ciclo de facturación 1 ▾

Encuesta ACEPTA TRANSFERENCIA ▾

Encuestas Disponibles Encuesta FO Empresas Medio ▾

Selecciona una tripleta

Codificación Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO' ▾

¿Se soluciono la petición? Sí No

Agendar Callback al cliente

Aplica Callback Si ▾
Motivo de Callback Seguimientos (Casos Especiales) ▾
Fecha del Callback 2015-06-24 **Hora** 20 ▾ : 40 ▾

Comentarios

Guardar información

Figura 4.3.4 Llenado de bitácora.

La aplicación envía mensaje de validación del llenado como el de la figura 4.3.5.

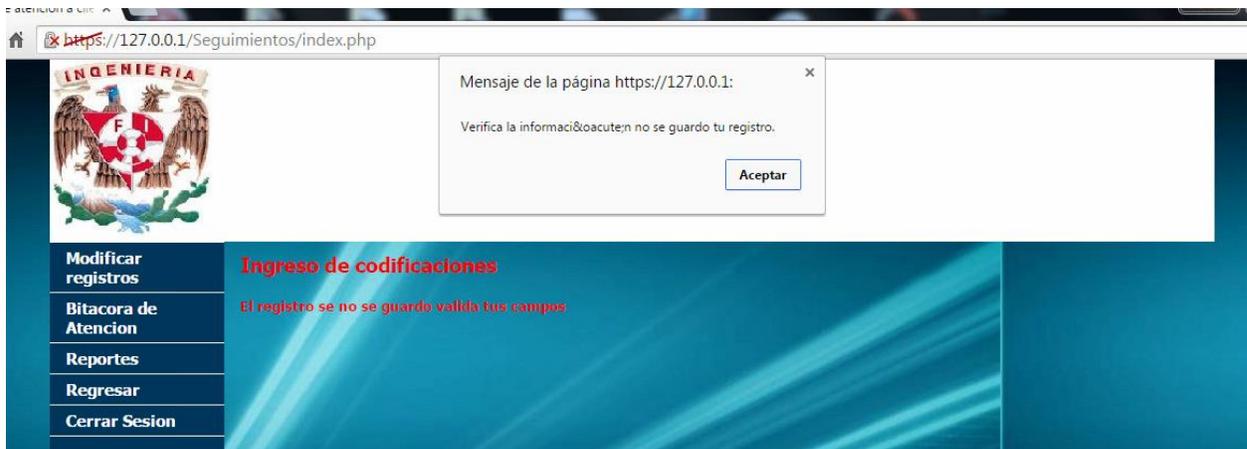


Figura 4.3.5 mensaje de verificación de llenado de bitâcora.

Al seleccionar la opción de aceptar se regresa a la pantalla anterior con las leyendas que se señalan en la figura 4.3.6

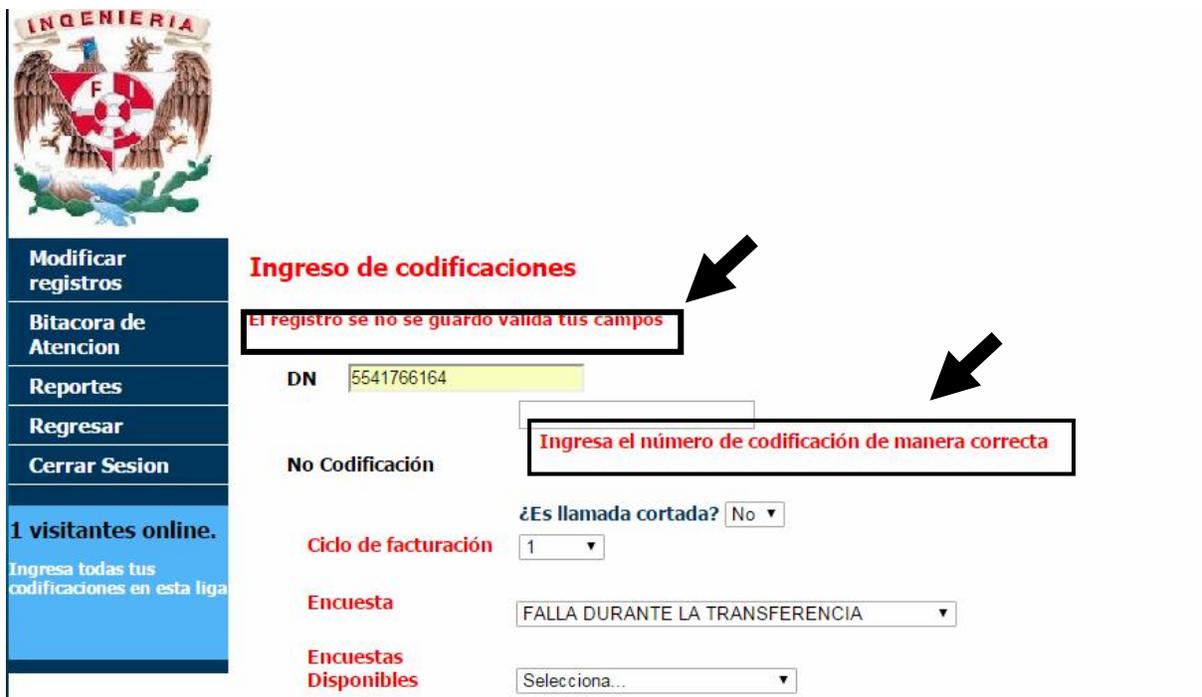


Figura 4.3.6 leyendas de verificación de bitâcora.



https://127.0.0.1/Seguimientos/index.php

registros

Bitacora de Atencion

Reportes

Regresar

Cerrar Sesion

1 visitantes online.

Ingresa todas tus codificaciones en esta liga

El registro se no se guardo valida tus campos

DN

Ingresa el número de codificación de manera correcta

No Codificación

¿Es llamada cortada?

Ciclo de facturación

Encuesta

Encuestas Disponibles

Selecciona una tripleta

Codificación

¿Se soluciono la petición? Sí No

Selecciona una opción

Agendar Callback al cliente

Aplica Callback

Motivo de Callback

Fecha del Callback Hora :

Figura 4.3.7 Llenado correcto de bitácora.

Se realiza el correcto llenado del formulario figura 4.3.7 y se guarda.

Para validar que se guardó el registro se verifica en el módulo registros como se muestra en la Figura 4.3.8

Bitacora

Regresar

1 visitantes online.

Ingresa todas tus Codificaciones en esta liga

Consulta de registro

Consulta aqui

Consultar

Modificación de caso

Acceso a Chat

Se generó una consulta con 12 registros

DN	Motivo	Fecha	Estatus	ID	Ejecutivo	Supervisor	Plataforma
5541766164	Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'	2015-06-24 03:31:14	Seguimiento	4216706 4216706	PRUEBA	UNAM	Callcenter
Prueba	ConsultaProductos y ServiciosInformacion de Promociones	2015-06-10 06:03:47	Seguimiento	4216705 4216705	PRUEBA	UNAM	Callcenter

Figura 4.3.8 vista modificar registro.



- Modificar registro

Se valida el funcionamiento del módulo modificar registro, donde es posible editar un reporte para mantenerlo actualizado y darle seguimiento a cada caso registrado en el sistema.

Bitacora	Modificación de registro
Modificar registro	
Reportes Liga	
CheckList Datos	
Regresar	
1 visitantes online.	
Ingresar todas tus codificaciones en esta liga	
Consulta de DN	
Consulta aqui	
Consultar	
Guardar	

Modificación de registro

DN 5541766164

Motivo Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'

Fecha 2015-06-24 03:31:14

Estatus Seguimiento ▼

Codificación 3

Comentario PRUEBA Seguimiento Callcenter 2015-06-24 03:31:14: No se refleja el bono; anterior

Agendar Callback al cliente

Nuevo Comentario

Evaluación del proceso Selecciona ▼

Figura 4.3.9 Modulo modificar registro

Se posiciona el puntero en nuevo comentario y es posible actualizar el estatus y seleccionar la evaluación del proceso como se muestra en la Figura 4.3.9



<ul style="list-style-type: none"> Bitacora Modificar registro Reportes Liga CheckList Datos Regresar <p>1 visitantes online.</p> <p>Ingresar todas tus codificaciones en esta liga</p> <p>Consulta de DN</p> <p>Consulta aqui</p> <p>Consultar</p>	<h3>Modificación de registro</h3> <p>DN 5541766164</p> <p>Motivo Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'</p> <p>Fecha 2015-06-24 03:31:14</p> <p>Estatus Seguimiento</p> <p>Codificación 3</p> <p>Comentario PRUEBA Seguimiento Callcenter 2015-06-24 03:31:14: No se refleja el bono;</p> <p>anterior</p> <p>Agendar Callback al cliente</p> <p>Nuevo</p> <p>Comentario</p> <p>Evaluación del proceso Proceso Correcto</p> <p>Guardar</p>
---	---

Figura 4.3.10 Nuevo comentario de bitácora

Se realiza la actualización de la información ver figura 4.3.10.

<ul style="list-style-type: none"> Bitacora Modificar registro Reportes Liga CheckList Datos Regresar <p>1 visitantes online.</p> <p>Ingresar todas tus codificaciones en esta liga</p> <p>Consulta de DN</p> <p>Consulta aqui</p> <p>Consultar</p>	<h3>Modificación de registro</h3> <p>DN 5541766164</p> <p>Motivo Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'</p> <p>Fecha 2015-06-24 03:31:14</p> <p>Estatus Escalado</p> <p>Codificación 3</p> <p>Comentario '\PRUEBA Seguimiento Callcenter 2015-06-24 03:31:14: No se refleja el bono;\' - Prueba PRUEBA Seguimiento Callcenter 2015-06-24 03:35:31: **Proceso Correcto** el usuario pregunta por su estatus;\' - Prueba PRUEBA Escalado Callcenter 2015-06-24 03:36:28: **Proceso Correcto** Se escala reporte por problemas con el servidor;</p> <p>anterior</p> <p>Agendar Callback al cliente</p> <p>Nuevo</p> <p>Comentario</p> <p>Evaluación del proceso Selecciona</p> <p>Guardar</p>
---	--

Figura 4.3.11 Historial de cometarios.

En el campo comentario se muestra el historial de comentarios ver figura 4.3.11.



- **Reportes.**

Se valida el módulo de reportes donde el resultado debe ser un archivo de Excel de la consulta.



Figura 4.3.12 módulos de reportes

Se selecciona el filtro requerido y se da click en exportar base figura 4.3.12, en la figura 4.3.13 se muestra el archivo del reporte en Excel.

Reportes BACTM

Reporte generado con datos del día 2015-06-24 al día 2015-06-24

ID	DN	Ciclo	Plataforma	Motivo	Fecha	Ejecutivo	Estatus	Comentario	Tripleta1	MotivoCallback	Encuesta
4216706	5541766164	1	Callcenter	Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'	24/06/2015 3:31	PRUEBA	Escalado	06-24 03:31:14: No se refleja el bono;\n - Prueba PRUEBA Seguimiento Callcenter 2015-06-24 03:35:31: **Proceso Correcto** el usuario pregunta por su estatus;\n - Prueba PRUEBA Escalado Callcenter 2015-06-24 03:36:28: **Proceso Correcto** Se escala reporte	Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'	Soporte a Movitaik	ACEPTA TRANSFERENCIA

La consulta tardó 0.06 segundos.-

Figura 4.3.13 Reporte Excel

Se verifica que los módulos de la aplicación interactúen como un todo.

Última modificación del registro y su vista en Excel, vea figura 4.3.14.



Bitacora

Modificar registro

Reportes Liga

CheckList Datos

Regresar

1 visitantes online.

Ingresar todas tus modificaciones en esta liga

Consulta de DN

Consulta aqui

Consultar

Modificación de registro

DN 5541766164

Motivo Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'

Fecha 2015-06-24 03:31:14

Estatus Escalado

Codificación 3

Comentario '\PRUEBA Seguimiento Callcenter 2015-06-24 03:31:14: No se refleja el bono'\ - Prueba PRUEBA Seguimiento Callcenter 2015-06-24 03:35:31: **Proceso Correcto** el usuario pregunta por su estatus;' - Prueba PRUEBA Escalado Callcenter 2015-06-24 03:36:28: **Proceso Correcto** Se escala reporte por problemas con el servidor;

anterior

Agendar Callback al cliente

Nuevo

Comentario

Evaluación del proceso Selecciona

Guardar

Reportes BACTM

Reporte generado con datos del día 2015-06-24 al día 2015-06-24

ID	DN	Ciclo	Plataforma	Motivo	Fecha	Ejecutivo	Estatus	Comentario	Tripleta1	MotivoCallback	Encuesta
4216706	5541766164	1	Callcenter	Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'	24/06/2015 3:31	PRUEBA	Escalado	06-24 03:31:14: No se refleja el bono'\ - Prueba PRUEBA Seguimiento Callcenter 2015-06-24 03:35:31: **Proceso Correcto** el usuario pregunta por su estatus;' - Prueba PRUEBA Escalado Callcenter 2015-06-24 03:36:28: **Proceso Correcto** Se escala reporte	Bono de 1000 minutos APLICADO Y NO SE RESPETA BENEFICIO'	Soporte a MovItalk	ACEPTA TRANSFERENCIA

La consulta tardó 0.06 segundos.

Figura 4.3.14 Reporte origen/destino

Fase de Mantenimiento:

A medida que pasa el tiempo, las aplicaciones de software deben ser sometidas a procesos de modificación que extiendan su vida útil o mejoren sus características. Corrección de bugs, adaptación a nuevos entornos tecnológicos o agregado de funcionalidad son algunas de las tareas que incluye el mantenimiento del software, una actividad que se repite periódicamente desde que empieza a utilizarse hasta su abandono definitivo.



A lo largo de su vida útil, la aplicación puede necesitar modificaciones por distintas razones, que determinan diferentes tipos de mantenimiento:

Preventivo:

Consiste en la modificación del software para mejorar sus propiedades (por ejemplo, aumentando su calidad y/o su mantenimiento) sin alterar sus especificaciones funcionales. Por ejemplo, se pueden incluir sentencias que comprueben la validez de los datos de entrada, re-estructurar los programas para mejorar su legibilidad, o incluir nuevos comentarios que faciliten la posterior comprensión del programa. Este tipo de mantenimiento es el que más partido saca de las técnicas de ingeniería inversa y reingeniería.

Correctivo: tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de causar un fallo. Un fallo ocurre cuando el comportamiento de un sistema es diferente del establecido en la especificación.

Entre otros, los fallos en el software pueden ser de:

- Procesamiento, por ejemplo, salidas incorrectas de un programa.
- Rendimiento, por ejemplo, tiempo de respuesta demasiado alto en una búsqueda de información.
- Programación, por ejemplo, inconsistencias en el diseño de un programa.
- Documentación, por ejemplo, inconsistencias entre la funcionalidad de un programa y el manual de usuario.

Perfectivo: Un mantenimiento perfectivo es aquel que requiere cambios en la especificación, normalmente debidos a cambios en los requisitos de un producto software.



Desde algo tan simple como cambiar el formato de impresión de un informe, hasta la incorporación de un nuevo módulo aplicativo.

Podemos definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Algunos autores dividen este tipo de mantenimiento en dos:

- **Mantenimiento de Ampliación:** orientado a la incorporación de nuevas funcionalidades.
- **Mantenimiento de Eficiencia:** que busca la mejora de la eficiencia de ejecución.

Este tipo de mantenimiento aumenta cuando un producto de software tiene éxito comercial y es utilizado por muchos usuarios, ya que cuanto más se utiliza un software, más peticiones de los usuarios se reciben demandando nuevas funcionalidades o mejoras en las existentes.

Aumentativo: Este tipo de mantenimiento se da cuando se incluyen nuevas funciones que no se contemplan al inicio del desarrollo del sistema y surgen como una necesidad del usuario.

Horizontal: El mantenimiento horizontal es aquel que se da debido a una actualización de hardware.

Vertical: Este tipo de mantenimiento es el que se lleva a cabo cuando hay una actualización de software.



4.5 Generación de reportes

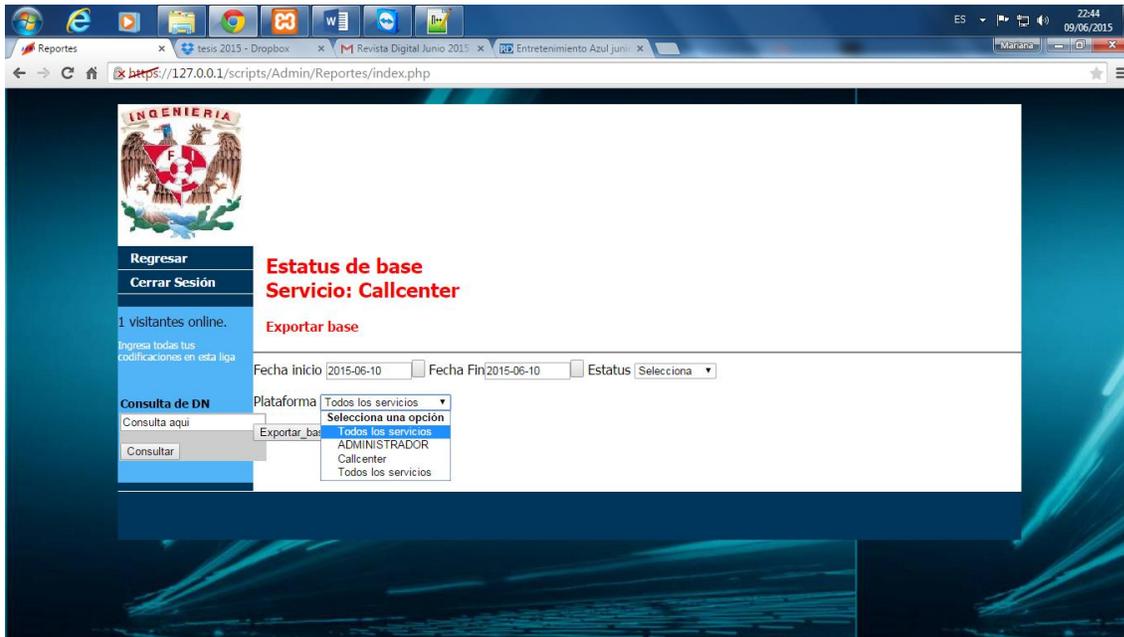


Figura 4.5.1 Generar reportes

Esta sección nos permitirá tener un registro detallado de todos los movimientos que se realicen dentro de la aplicación, también se puede observar que nuestra base tiene muchos más campos de los que se muestran en la bitácora. Esto se debe a que en la base se guardan todos los movimientos realizados por las otras áreas que tienen que utilizarla, no solo la de la bitácora

En las siguientes imágenes se muestra el reporte completo que se genera

Conclusiones

- En esta tesis se cumplieron los objetivos planteados, de tal manera que la bitácora de atención a usuarios cumple perfectamente con los requerimientos para satisfacer las necesidades de los usuarios y canalizarlos con las áreas correspondientes para una oportuna respuesta por parte de la compañía de telefonía móvil.
- En cualquier empresa y muy especialmente en aquellas cuyo objeto de su actividad es la venta de productos y servicios, la productividad se mide en términos de satisfacción del cliente, y el grado de esta satisfacción va más allá de la calidad del servicio, pues en su valoración entra en juego un factor dominante: La atención al cliente. Actualmente la mayoría de productos y servicios existentes en el mercado, poseen características muy similares. Esta homogeneidad dificulta enormemente los esfuerzos de las empresas por diferenciar sus productos o servicios respecto a los competidores.
- Esta bitácora de atención a clientes minimiza tiempos de procesamiento de datos, así como tiempos de respuesta a los problemas de los usuarios.
- Se logró hacer una aplicación que agiliza el tiempo de captura, medir el tiempo de solución al usuario, mejorar el seguimiento del reporte (callback al cliente) y así mejorar la imagen de la compañía ante el cliente.
- La atención al cliente es el nexo de unión de tres conceptos: Servicio al cliente, satisfacción del cliente y calidad en el servicio. En esta bitácora de servicio se llega a este objetivo.



- En este proyecto tuvimos la oportunidad de hacer una herramienta confiable que satisface, los posibles escenarios por parte de los clientes al requerir servicios a una compañía de telefonía móvil.
- En esquemas tradicionales de venta donde la entrega del producto al cliente en el proceso de ventas era el punto final de todo un proceso productivo, surge nuevas directrices que permiten mejorar los servicios y productos mediante un seguimiento a las inquietudes o problemas del cliente con servicios de telefonía móvil.
- La solución ofrecida para una empresa de telefonía móvil, para el área de call center para el manejo de solicitudes por parte de sus clientes y ventas le permite mantener un contacto más eficiente con las áreas involucradas con los requerimientos por parte de los clientes de dicha compañía.
- En esta tesis se logró detectar analizar e implementar una bitácora de atención a clientes de telefonía móvil, donde el ejecutivo será capaz de gestionar correctamente los diversos problemas que se presenten en el servicio para ser canalizados al área correspondiente en un menor tiempo.
- Se logró mitigar que la información se encuentre dispersa, facilitando el acceso y evitando invertir varias horas en consolidar los datos.
- El front-end del aplicativo se logró que el ejecutivo del call center puede interactuar de una manera intuitiva con el mismo y puede evitar errores humanos haciendo el uso de catálogos pre-capturados.



-
- En conclusión se debe evitar tensión en el cliente, mediante la actitud del profesional. Éste vigilará sus movimientos para que el cliente perciba soltura y profesionalidad a través de ellos, con herramientas adecuadas para resolver los problemas que demande el cliente en cuanto a servicios de Telefonía.



Bibliografía

- Pressman, Roger, “Ingeniería del Software, Un enfoque Práctico”, Mc Graw-Hill, 7ma edición, 2010.
- Kruglinski, David. “Sistemas de Administración de Bases de Datos”, Mc Graw-Hill, 1984.
- Vandekopple, Julius. “Lenguajes de Programación”, Mc-Graw-Hill México, 2da edición, 2000.
- Quigley Ellie, PHP y MySQL Práctico: Para Programadores y Diseñadores WEB. 1ra edición. Editorial: Anaya Multimedia 2007 452pp.

Última visita 16 mayo de 2015

- <http://www.contactcentersonline.com/novedad.php?IdNovedad=14248>
- <http://www.relacioncliente.es/un-paso-mas-hacia-la-iso-18295-norma-internacional-para-los-contact-centers-20606>
- https://www.nacionsservicios.com.ar/novedades_ampliada_04.html
- <https://prezi.com/gut1vubldtxj/normas-internacionales-delos-contact-center/>
- <http://www.stps.gob.mx/bp/secciones/dgsst/normatividad/normas/Nom-017.pdf>
- <http://www.stps.gob.mx/bp/secciones/dgsst/normatividad/normas/Nom-011.pdf>

Última visita 23 mayo de 2015

- <http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/MySQL.pdf>
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- <http://es.kioskea.net/contents/148-entorno-cliente-servidor>
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- http://es.wikipedia.org/wiki/Bloqueo_mutuo



- <http://es.kioskea.net/contents/144-redes-cliente-liviano>
- <http://es.wikipedia.org/wiki/Cliente-servidor>
- <https://msdn.microsoft.com/es-es/library/bb972232.aspx>
- http://programacion.net/articulo/programacion_orientada_a_objetos_279

Última visita 28 mayo de 2015

- <http://www.firebirdmanual.com/firebird/es/firebird-manual/2/-que-es-firebird-/8>
- <http://es.wikipedia.org/wiki/Firebird>
- <http://es.wikipedia.org/wiki/PostgreSQL>
- <http://postgresql-dbms.blogspot.mx/p/limitaciones-puntos-de-recuperacion.html>
- http://www.postgresql.org.es/sobre_postgresql
- <http://postgresql-dbms.blogspot.mx/p/limitaciones-puntos-de-recuperacion.html>
- <http://www.aalpha.net/modern-web-programming-language-wars/>
- <http://es.axiacore.com/blog/2008/04/comparativa-ruby-php-python/>
- <http://www.taringa.net/posts/info/15423619/Comparacion-PHP-Python-y-Ruby.html>
- <http://aplicaciones-web-lenguajes-programaci.blogspot.mx/2011/12/ruby.html>
- <https://www.ruby-lang.org/es/about/>
- <http://es.wikipedia.org/wiki/Python>
- <http://www.scribd.com/doc/50288837/Caracteristicas-de-PHP#scribd>
- <http://aprendizajeweb.com/CursoJava2012/LibrosVarios/Programacion%20Orientada%20A%20Objetos%20-%20Luis%20Joyanes%20Aguilar.pdf>



Última visita 7 junio de 2015

- http://www.mclibre.org/consultar/php/otros/in_php_instalacion.html
- <http://www.deseoaprender.com/Joomla-1-5-basico/tema-03/joomla-en-local-tema-03-leccion-3-mysql-en-local.html>
- <https://cursoprogramador.wordpress.com/tag/xampp/>
- <http://txtplano.blogspot.mx/2014/04/ejemplo-de-creacion-triggers-mysql.html>
- <http://rightnowweb.blogspot.mx/2010/06/procedimiento-almacenado-en-phpmyadmin.html>
- <http://www.hermosaprogramacion.com/2014/06/mysql-procedure-show-create/>
- <http://www.kmoransv.me/2014/05/procedimientos-almacenados-en-mysql.html>
- <https://romaneduardo.wordpress.com/2010/08/30/consultar-procedimientos-mysql-almacenados-con-phpmyadmin/>
- http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=611:crear-tablas-mysql-con-phpmyadmin-tipos-de-datos-basicos-varchar-int-float-insertar-filas-cu00840b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193
- <http://fcqi.tij.uabc.mx/usuarios/luisgmo/data/8.3%20prb-cal-mant.pdf>
- <https://rfolivares.wordpress.com/2010/09/08/pruebas-alfa-y-beta/>
- <https://www.galatea-it.com/servicios/calidad-de-software/prueba-estres>
- http://www.ecured.cu/index.php/Estrategia_de_Pruebas
- <https://swcb37.files.wordpress.com/2013/08/mantenimiento-de-software.pdf>
- <http://www.4rsoluciones.com/tipos-de-mantenimiento-de-software/>