



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE INVESTIGACIÓN DE SISTEMAS DE INFORMACIÓN

DESARROLLO DE UNA SOLUCIÓN PARA EL PROCESO DE ASIGNACIÓN DE CITA TÉCNICA PROGRAMADA DE UN PROVEEDOR DE SERVICIO DE INTERNET

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por:

Br. Kevin Marin

C.I. V-21.195.316

Br. Misael Polidor

C.I. V-20.878.802

Tutor: Prof. Dr. Franklin Sandoval

Caracas, mayo 2019

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "DESARROLLO DE UNA SOLUCIÓN PARA EL PROCESO DE ASIGNACIÓN DE CITA TÉCNICA PROGRAMADA DE UN PROVEEDOR DE SERVICIO DE INTERNET" y presentado por el Br. Kevin Marín C.I. 21.195.316, y Br. Misael Polidor C.I. 20.878.802 a los fines de optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Dicho trabajo, leído por cada uno de los miembros del jurado, se fijó el día 21 de mayo del año 2019, a las 9:00 a.m., para que sus autores lo defendieran en forma pública en la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual se respondieron las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 20 puntos.

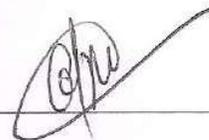
En fe de lo cual se levanta la presente Acta, en Caracas a los 21 días del mes de mayo del año 2019.



Prof. Franklin Sandoval
Tutor



Prof. Mercy Ospina
Jurado Principal



Prof. Carlos Acosta
Jurado Principal

DEDICATORIA

A mis padres por brindar desde sus posibilidades y sacrificios, todas las oportunidades para seguir con mi educación y poder cumplir con mis sueños.

A mis familiares por siempre creer en mí y darme el apoyo necesario en todo momento para seguir con mis metas.

Kevin Marin Saavedra

A mis familiares, sobre todo a mis abuelos por su apoyo incondicional desde mi niñez y siempre darme la educación necesaria con los mejores valores.

A mis amigos, compañeros de trabajo y a todas aquellas personas que hicieron esto posible.

Misael Polidor La salvia

AGRADECIMIENTOS

Principalmente agradecemos a nuestros padres y abuelos por su gran apoyo incondicional a lo largo de todo este proceso, siempre motivándonos a ser mejores persona y que todo se puede con perseverancia y dedicación.

A nuestros hermanos por estar siempre ahí para nosotros, creer siempre en nosotros y ayudarnos a desconectarnos un poco de todo.

A todos nuestros amigos, por los buenos momentos, aprendizajes, y tanta ayuda brindada durante todos estos años. A nuestros hermanos “Los varonsitos” por todo su apoyo y estar con nosotros en los peores momentos. A las nuevas amistades y relaciones que hemos conseguido, nos han demostrado tanto apoyo y darnos impulso para afrontar todos los retos en los que nos hemos enfrentado. Y a nuestras parejas por su apoyo incondicional y ser esa luz que inspira a seguir en este camino lleno de obstáculos, las amamos.

A nuestro tutor Franklin Sandoval por toda su paciencia. Gracias por enseñarnos a ser mejores profesionales y siempre exigir lo mejor de nosotros.

Al Centro de Investigación de Computación Grafica, por permitirnos ser parte de él durante tantos años de nuestra carrera, por darnos la oportunidad de convertirnos en preparador y descubrir lo mucho que nos gusta enseñar.

A la Universidad Central de Venezuela, la casa que vence la sombra, por abrirnos sus puertas y permitir nuestra formación académica y profesional, ser testigo de nuestro crecimiento personal y darnos la oportunidad de formarnos como Licenciado de Computación.

¡Gracias a todos!

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

DESARROLLO DE UNA SOLUCIÓN PARA EL PROCESO DE ASIGNACIÓN DE CITA TÉCNICA PROGRAMADA DE UN PROVEEDOR DE SERVICIO DE INTERNET

Autores: Kevin Jesus Marin Saavedra, C.I. V-21.195.316

Misael Josue Polidor La Salvia, C.I. V-20.878.802

Tutor: Prof. Franklin Sandoval

RESUMEN

El objetivo del presente Trabajo Especial de Grado se basa en la creación de un sistema haciendo uso de tecnologías web para construir un sistema que apoye en el proceso de automatización del control de citas, dado un reporte de servicio técnico en una compañía proveedora de servicio internet, de esta manera poder realizar seguimiento de los reportes, en busca de mejorar los tiempos de respuestas y corregir el problema de manera rápida. Asimismo, este sistema funcionara como base para futuros desarrollos tecnológicos que pueden estar orientados al análisis de datos, para tener proyecciones de cuáles son las fallas más comunes, evaluar el rendimiento servicio, tiempo de respuesta de un técnico, así como muchos indicadores que puedan impulsar a mejorar este tipo de compañías. Para el desarrollo del sistema, se aplicó una adaptación de la metodología de desarrollo ágil SCRUM. Se usaron herramientas web como; el framework SailsJS, usando AngularJS para la parte visual de la página web, Express para proveer los servicios web que proporcionar los datos necesarios y MongoDB como base de datos central donde se almacenan todos los datos. Estas tecnologías con un estilo de arquitectura MVC. Cabe destacar, que el proyecto se dividió en cuatro fases; teniendo así la fase de análisis donde se realizó el levantamiento y análisis de requerimientos, seguidamente la fase de desarrollo, donde se implementó la solución, luego la fase de pruebas funcionales para comprobar el comportamiento y funcionamiento y por último una fase de pruebas de aceptación para usuarios finales con el fin de tener retroalimentación sobre aspectos básicos de la aplicación.

PALABRAS CLAVE: Aplicación web, SailsJS, AngularJS, ExpressJS, NodeJS, MongoDB, MEAN, Reporte de servicio técnico, Compañía proveedora de internet, Automatización del control de citas.

ÍNDICE GENERAL

ÍNDICE GENERAL	1
ÍNDICE DE ILUSTRACIONES	4
ÍNDICE DE TABLAS	6
INTRODUCCIÓN	7
CAPÍTULO 1: PROBLEMA DE INVESTIGACIÓN	8
1.1. Título	8
1.2. Planteamiento del problema	8
1.3. Justificación	11
1.4. Objetivo General	12
1.5. Objetivos Especificos	12
1.6. Solución propuesta.....	12
1.6.1. Arquitectura conceptual a utilizar	13
1.6.2. Arquitectura tecnológica a utilizar.....	14
1.6.3. Metodología de desarrollo a utilizar.....	15
1.6.4. Descripción del flujo asociado a la solución.....	15
1.7. Alcance	19
CAPÍTULO 2: MARCO CONCEPTUAL	20
2.1. Sistema de Información	20
2.1.1. Definición.....	20
2.1.2. Tipos de Sistemas de Información	20
2.1.2.4. Sistema de apoyo a la toma de decisiones (DSS).....	21
2.2. Tecnología de desarrollo web	22
2.2.1. Internet	22
2.2.2. World Wide Web	22
2.2.3. HTTP	23
2.2.4. HTML.....	23
2.2.5. CSS	23
2.2.6. Less	24
2.2.7. Bootstrap.....	24
2.2.8. JavaScript.....	24
2.2.9. AJAX	24
2.2.10. JWT	25
2.2.11. JSON	25
2.2.12. AngularJS	25

2.2.13.	Servicio Web	25
2.2.14.	Protocolo SOAP	26
2.2.15.	XML	26
2.2.16.	Protocolo REST	26
2.2.17.	NodeJS	27
2.2.18.	ExpressJS	27
2.2.19.	SailsJS	27
2.2.20.	Waterline	28
2.2.21.	Web Socket	28
2.2.22.	CRONjob	28
2.2.23.	MEAN	28
2.3.	Base de Datos	29
2.3.1.	Sistema manejador de Base de Datos	29
2.3.2.	Base de Datos SQL	29
2.3.3.	Base de Datos NoSQL	30
2.3.4.	Cuadro Comparativo Bases de datos SQL Y NoSQL	31
CAPÍTULO 3: MARCO METODOLÓGICO		32
3.1.	Metodología tradicional Base de Datos	32
3.2.	Metodología ágil	33
3.3.	Comparación entre las metodologías de desarrollo	34
3.4.	SCRUM	34
3.4.1.	Teoría de Scrum	35
3.4.2.	El equipo Scrum	35
3.4.3.	Eventos de Scrum	36
3.4.4.	Artefactos de Scrum	38
3.5.	Comparación entre metodologías ágiles	39
CAPÍTULO 4: MARCO APLICATIVO		40
4.1	Descripción general de la solución	40
4.2	Aplicación de la metodología SCRUM	42
4.2.1	Lista de Objetivos:	42
4.2.2	Lista de tareas de la iteración	43
4.3	Requerimientos del sistema	45
4.3.1	Requerimientos funcionales	45
4.3.2	Requerimientos no funcionales	45
4.4	Perfiles de usuarios	45
4.5	Descripción del flujo asociado a la solución	47

4.6	Análisis del modelo de datos	50
4.6.1	Listado de tablas de la aplicación web	50
4.6.2	Modelo de datos.....	51
4.7	Descripción de los módulos del sistema e interfaces.....	52
4.7.1	Descripción de los módulos del sistema.	52
4.7.2	Logo diseñado y Paleta de colores	53
4.7.3	Descripción de las interfaces del sistema.	54
4.8	Fase de pruebas	65
4.8.1	Pruebas funcionales	65
4.8.2	Pruebas de aceptación y usabilidad.....	72
	CONCLUSIONES.....	79
	RECOMENDACIONES	80
	REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES.....	81

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Flujograma del proceso actual de Reporte.....	9
Ilustración 2: Flujograma del proceso actual de revisión de reportes.	10
Ilustración 3: Flujograma del proceso actual de consulta de reportes.	11
Ilustración 4 : Arquitectura conceptual de la solución.....	13
Ilustración 5: Arquitectura tecnologica de la solución.	15
Ilustración 6: Flujograma del proceso propuesto de reporte	16
Ilustración 7: Flujograma del proceso propuesto de revisión de citas.....	17
Ilustración 8: Flujograma del proceso propuesto de consulta de reportes.....	18
Ilustración 9: Clasificación de los sistemas de información.....	20
Ilustración 10: Desarrollo de software por etapas a través de Sprints.	38
Ilustración 11: Arquitectura de la solución tecnologica.....	41
Ilustración 12: Flujograma del proceso propuesto de reporte	47
Ilustración 13: Flujograma del proceso propuesto de revisión de citas.....	48
Ilustración 14: Flujograma del proceso propuesto de consulta de reportes.....	49
Ilustración 15: Modelo de datos.....	51
Ilustración 16: Modelo de datos.....	52
Ilustración 17: Logo diseñado para el sistema.....	53
Ilustración 18: Inicio de sesión de la aplicación.	54
Ilustración 19: Recuperar contraseña.	54
Ilustración 20: Consulta de reporte para el cliente.....	55
Ilustración 21: Detalle consulta de reporte para el cliente.....	55
Ilustración 22: Consulta de cliente para el empleado.	56
Ilustración 23: Detalle de cliente.....	56
Ilustración 24: Lista de reportes.	57
Ilustración 25: Crear nuevo reporte.....	57
Ilustración 26: Calendario de citas.	58
Ilustración 27: Detalle de cita.....	58
Ilustración 28: Lista de clientes.	59
Ilustración 29: Formulario de nuevo cliente.....	59
Ilustración 30: Lista de técnicos.	60
Ilustración 31: Formulario de nuevo técnico paso 1.....	60
Ilustración 32: Formulario de nuevo técnico paso 2.....	61
Ilustración 33: Lista de empleados.....	61
Ilustración 34: Formulario de nuevo empleado.	62
Ilustración 35: Detalle de técnico.....	62
Ilustración 36: Detalle de empleado.....	63
Ilustración 37: Gerencia calidad.	63
Ilustración 38: Gerencia técnica.	64
Ilustración 39: Perfil del técnico.....	64
Ilustración 40: Resultados de pregunta 1 del instrumento de evaluación.....	73
Ilustración 41: Resultados de pregunta 2 del instrumento de evaluación.....	73
Ilustración 42: Resultados de pregunta 3 del instrumento de evaluación.....	74
Ilustración 43: Resultados de pregunta 4 del instrumento de evaluación.....	74
Ilustración 44: Resultados de pregunta 5 del instrumento de evaluación.....	75
Ilustración 45: Resultados de pregunta 6 del instrumento de evaluación.....	75

Ilustración 46: Resultados de pregunta 7 del instrumento de evaluación	76
Ilustración 47: Resultados de pregunta 8 del instrumento de evaluación	76
Ilustración 48: Resultados de pregunta 9 del instrumento de evaluación	77
Ilustración 49: Resultados de pregunta 10 del instrumento de evaluación	77
Ilustración 50: Resultados de pregunta 11 del instrumento de evaluación	78
Ilustración 51: Resultados de pregunta 12 del instrumento de evaluación	78

ÍNDICE DE TABLAS

Tabla 1: Cuadro Comparativo Bases de datos SQL y NoSQL.....	31
Tabla 2: Metodologías tradicionales y Metodologías ágiles.	34
Tabla 3: Metodologías ágiles SCRUM y XP.	39
Tabla 4: Lista de Objetivos y tareas realizadas	42
Tabla 5: Pruebas funcionales por módulo.....	65
Tabla 6: Matriz de ciclos de prueba.....	66
Tabla 7: Instrumento de evaluación para pruebas de aceptación.....	72

INTRODUCCIÓN

Las compañías proveedoras de servicio internet brindan conexión a internet de banda ancha con cobertura nacional a sus clientes. Para poder ofrecer un servicio de calidad estas compañías requieren realizar citas técnicas asistidas a sus clientes por temas de instalación, mantenimiento o bien cuando sean reportadas fallas de rendimiento del servicio. Por un lado, la asistencia técnica requiere de una previa cita que debe ser programada por la compañía proveedora debido a un reporte que es generado mediante una llamada telefónica, un correo electrónico o en una agencia donde el cliente detalle su caso. Dicho lo anterior, el reporte generado requiere de un seguimiento adecuado para que el personal técnico pueda resolver el incidente de forma eficiente. Por lo tanto, llevar un control preciso de cada reporte puede volverse un proceso tedioso.

Aunado a la situación, el caso de estudio de este trabajo especial de grado es: el proceso de asignación de cita técnica programada de un proveedor de servicio internet. La gestión actual de este proceso presenta varios inconvenientes; uno de ellos es no tener un tiempo definido para dar respuesta o solución a un reporte, no se tiene un registro del técnico que toma el reporte, los problemas presentados no quedan registrados en un histórico, no existe una escala de prioridad para los reportes, de este modo dado la gravedad para el cliente o en cómo puede afectar la compañía unos sean atendidos con más premura que otros, y por último que el cliente no está al tanto del estado de su reporte. Estos problemas mencionados buscan ser solventados por la compañía. Para ello, se propone el uso las aplicaciones con tecnología web con la intención de mejorar y agilizar el desempeño de este proceso.

Este documento se estructura en cuatro capítulos, el primer capítulo considerado como el Problema de Investigación, se presenta el flujo del proceso actual y se expone el caso de estudio, además se plantean los objetivos y alcances del estudio. En el segundo capítulo considerado como el Marco Teórico consta de los conceptos, definiciones de los temas y tecnologías consideradas para el desarrollo de este estudio. En el tercer capítulo considerado como Marco Metodológico se definen las metodologías ágiles y tradicionales del desarrollo de un producto, exponiendo ventajas, desventajas de cada una y mediante un cuadro comparativo se determina el uso de una de ellas como propuesta de este estudio. En el cuarto capítulo, denominado como el Marco Aplicativo, en el cual se explican y describen las fases que se aplicaron para la elaboración del sistema, según una adaptación de la metodología de desarrollo ágil SCRUM.

Por último, las conclusiones y recomendaciones sobre el desarrollo de una solución para el proceso de asignación de cita técnica programada de un proveedor de servicio de internet.

CAPÍTULO 1: PROBLEMA DE INVESTIGACIÓN

1.1. Título

Desarrollo de una solución para el proceso de asignación de cita técnica programada de un proveedor de servicio de internet.

1.2. Planteamiento del problema

Actualmente en el área de las compañías proveedoras de servicio de internet se experimenta una alta demanda de clientes, estos exigen un servicio de calidad y un servicio técnico que solucione sus problemas de manera eficiente, por consiguiente, es necesario adaptarse a las nuevas tecnologías y desarrollar un proceso óptimo para el control de citas a consecuencia de un reporte de servicio técnico que permita mejorar los tiempos de respuestas y solucionar el problema de manera rápida.

El proceso actual para generar un reporte de servicio técnico es a través de una llamada telefónica que realiza el cliente, esta llamada es atendida por un empleado encargado de tomar nota del problema y crear el reporte correspondiente. Entonces, el reporte generado pasa a una lista de reportes pendientes, estos reportes pueden ser tomados por algunos de los técnicos disponibles en cualquier momento sin tener un tiempo de respuesta para dar solución al problema presentado. Una vez tomado un reporte el técnico realiza las acciones pertinentes para solventar el caso, esperando solucionar el problema y cerrar el caso o agregar un comentario al reporte. Los comentarios se agregan al reporte sin especificar quién lo hizo ya que es un archivo de texto plano sin ningún formato en específico. El seguimiento a estos reportes se hace complicado, ya que no se tiene un control de que técnicos están atendiendo el problema, además que los clientes están constantemente llamando a las líneas telefónicas de atención al cliente de la compañía, preguntando sobre su reporte, ya que no tienen una forma de llevar un seguimiento del mismo. Debido a este factor se buscan soluciones y/o alternativas que permitan cumplir estos objetivos de una manera automatizada, y así optimizar el proceso con nuevas tecnologías.

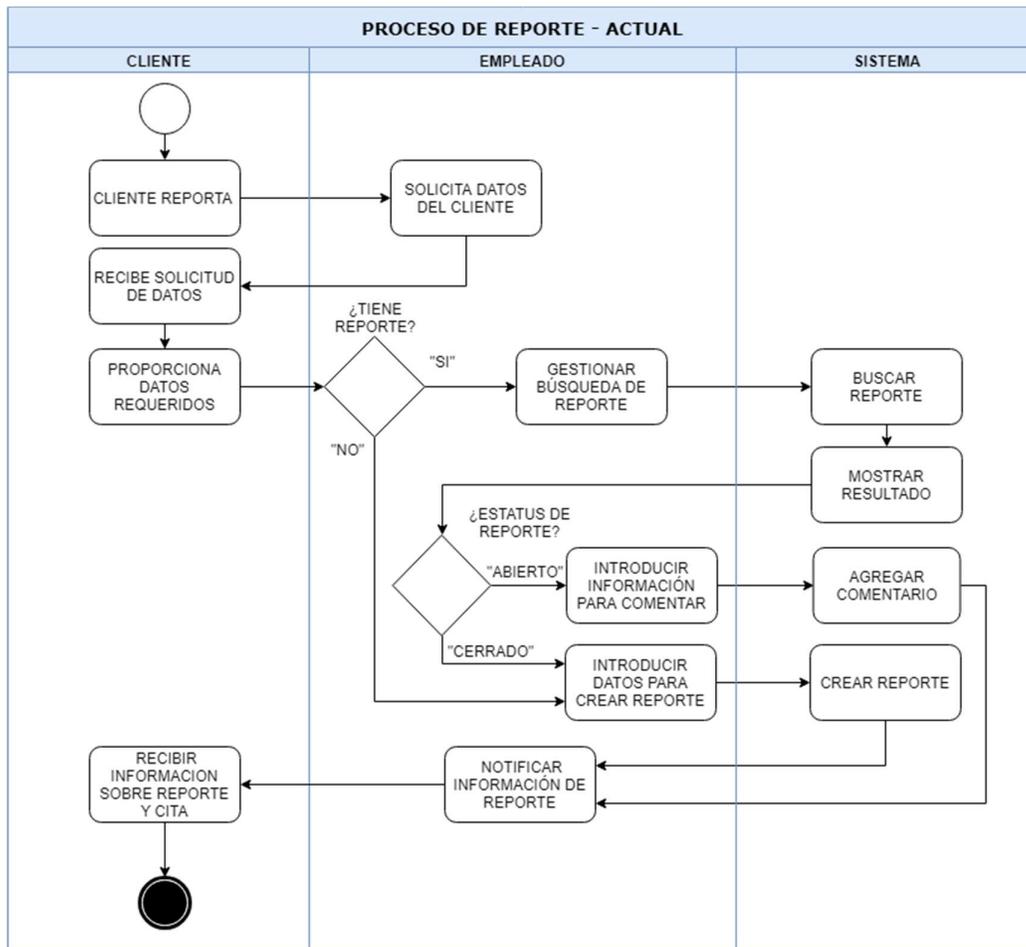


Ilustración 1: Flujoograma del proceso actual de Reporte.

En la ilustración 1, se presenta un flujoograma del proceso de reporte actual en el cual intervienen el cliente, el empleado de atención al cliente y el sistema. Este proceso inicia con el cliente que manifiesta un inconveniente o solicitud referente al servicio, el empleado solicita el número de contrato del cliente y número de reporte en caso de tenerlo. Si el cliente no posee reporte, el empleado crea un nuevo reporte en el sistema y le informa al cliente su número de reporte generado y el tiempo de respuesta estimado para su problema. Si el cliente posee reporte, el empleado hace una búsqueda en el sistema con el número de reporte para detallar la información correspondiente y verifica el estatus del reporte, por el contrario, si este es de estatus cerrado, entonces procede a crear un nuevo reporte y le indica al cliente el número de reporte generado en el sistema y en qué lapso de tiempo será contactado. Si el estatus del reporte es abierto, el empleado agrega en el reporte un comentario a través del sistema indicando que el cliente se ha comunicado nuevamente y, por último, el empleado le notifica al cliente información sobre el reporte y tiempo aproximado de respuesta.

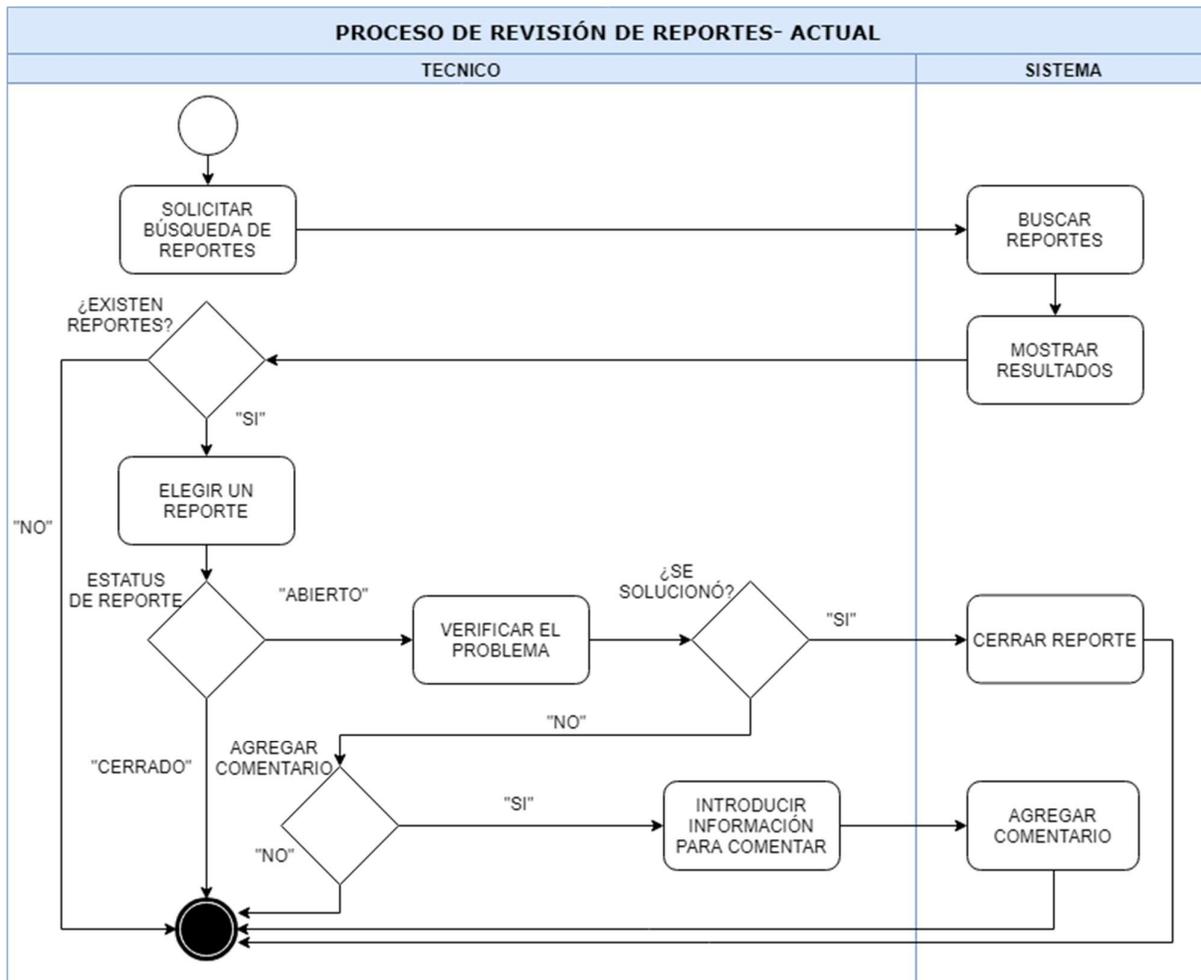


Ilustración 2: Flujograma del proceso actual de revisión de reportes.

En la ilustración 2, se presenta un flujograma del proceso de revisión de reportes actual en el cual intervienen el técnico y el sistema, este proceso inicia con el técnico buscando los reportes en el sistema, ya teniendo los resultados de esta búsqueda si no existen reportes se termina el proceso, pero si existen reportes se elige un reporte; si el estatus del reporte es cerrado, no se realiza nada al reporte y se finaliza el proceso. Si el estatus del reporte es abierto, el técnico verifica el problema, si se solucionó se cierra el reporte y termina el proceso; si no se solucionó el técnico decide si agregar un comentario y se termina el proceso.

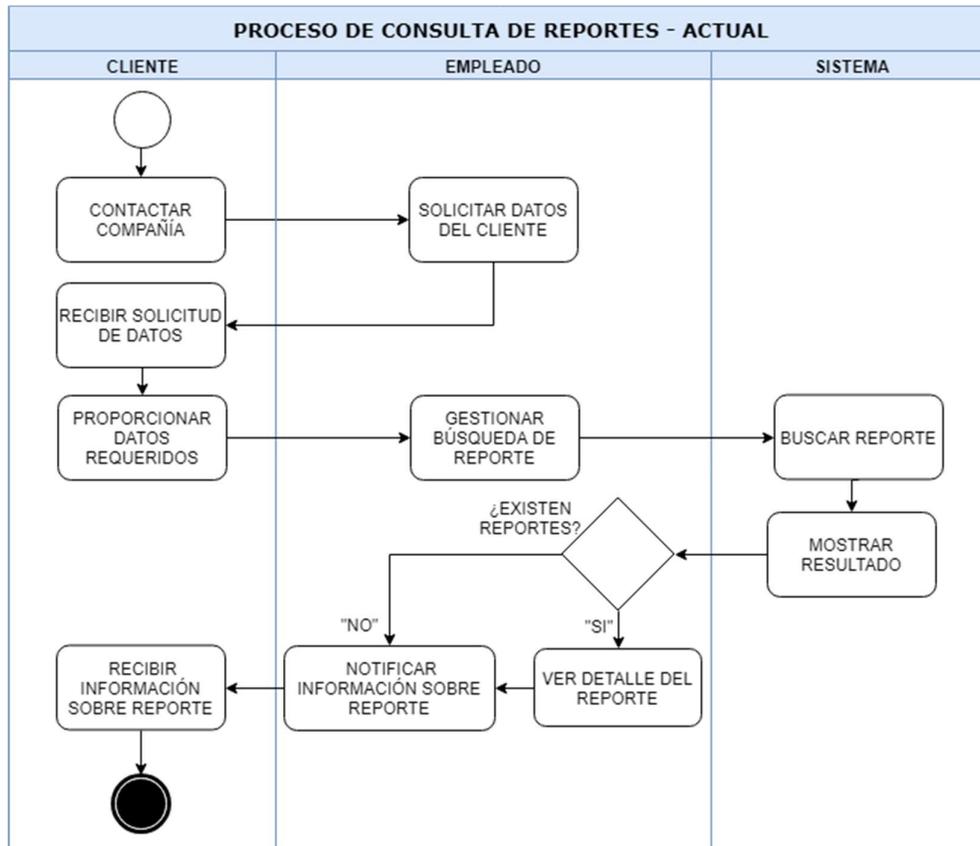


Ilustración 3: Flujograma del proceso actual de consulta de reportes.

En la ilustración 3, se presenta un flujograma del proceso de consulta de reporte actual en el cual intervienen el cliente, el empleado de atención al cliente y el sistema, este proceso inicia con el cliente que contacta a la compañía, el empleado solicita el número de contrato del cliente y número de reporte en caso de tenerlo. Seguidamente, el empleado hace una búsqueda en el sistema con el número de contrato y número de reporte para detallar la información correspondiente y verifica si existe el reporte, si existe el empleado detalla el reporte y da al cliente información sobre reporte terminando así el proceso, en caso que el reporte no exista se le notifica al cliente la información y se termina el proceso.

1.3. Justificación

Un sistema con tecnologías web, es la mejor solución para automatizar la programación de citas dado un reporte de servicio técnico, para que de esta forma la compañía proveedora de servicio internet pueda llevar un registro y control de estas citas entre un cliente y un técnico, en una fecha designada.

En efecto, la implementación de este sistema representa una disminución en los tiempos de respuesta de los reportes, y garantiza que los técnicos de servicio como recursos de la compañía, estén siendo aprovechados oportunamente, agilizando que la gestión sea más eficiente y se asegure el cumplimiento de la cita.

1.4. Objetivo General

Desarrollar una solución con tecnología web que permita la automatización y control de citas dado un reporte de servicio técnico de una compañía proveedora de servicio internet.

1.5. Objetivos Específicos

- Analizar requerimientos del proceso del control de citas dado un reporte de servicio técnico.
- Diseñar cada uno de los componentes y módulos de la arquitectura de la solución.
- Desarrollar cada uno de los componentes y módulos de la arquitectura de la solución.
- Aplicar un conjunto de pruebas funcionales, no funcionales y de calidad de datos a cada uno de los componentes y módulos de la arquitectura de la solución.

1.6. Solución propuesta

Se propone desarrollar una aplicación web que podrá controlar el proceso de programación y seguimiento de citas dado un reporte de servicio técnico de una compañía proveedora de servicio internet. Es así como mediante la aplicación web podrán autenticarse el empleado y el técnico.

Por un lado, el empleado será capaz de obtener una lista de clientes, una lista de reportes, crear reportes describiendo el problema presentado, definir citas para un reporte en una fecha específica, asignar un técnico a una cita, hacer seguimiento de un reporte con un número de identificación. A su vez el técnico podrá obtener una lista de sus citas asignadas, hacer seguimiento, dejar comentario, subir documento y cambiar estatus de sus citas.

De este modo, el cliente que reporta su inconveniente con el servicio podrá consultar en la aplicación web información sobre el estado de su reporte por medio de su número de contrato y el número de reporte asignado.

La aplicación propuesta tendrá como ventaja definir fechas específicas de las citas dado un reporte para así dar respuesta eficiente a los clientes, asignando a cada cita un técnico responsable, quién podrá dejar comentarios debidamente identificados, además de subir archivos referentes al caso. Asimismo, el seguimiento de los reportes por parte de los empleados se agiliza ya que se

tendrá información detallada de quién está atendiendo el reporte y la fecha pautada para su solución.

De esta manera, al existir fechas fijadas para las citas de los reportes, se evita que el cliente realice llamadas constantemente a las líneas telefónicas de la compañía consultando sobre su reporte, adicionalmente podrá consultar a través de la aplicación web con su número de contrato y número de identificación de su reporte la información detallada correspondiente.

1.6.1. Arquitectura conceptual a utilizar

Se plantea una aplicación web que administre los usuarios, permitiendo la creación de reportes y control de citas técnicas con los clientes de la compañía. El intercambio de información se realiza a través de los servicios web que permiten el envío y recepción de datos asociados a los usuarios del sistema, los reportes y las citas técnicas. Estos servicios permiten la conexión entre los distintos componentes de la arquitectura de la solución propuesta. Los datos se almacenan en la base de datos central de la compañía que mediante una estructura orientada a documentos permite guardar en detalle la información asociada a los usuarios del sistema, los reportes generados por los empleados, almacenar los datos pertinentes al control de citas de servicio técnico, además de los documentos necesarios para el buen seguimiento y registro del reporte de servicio técnico.

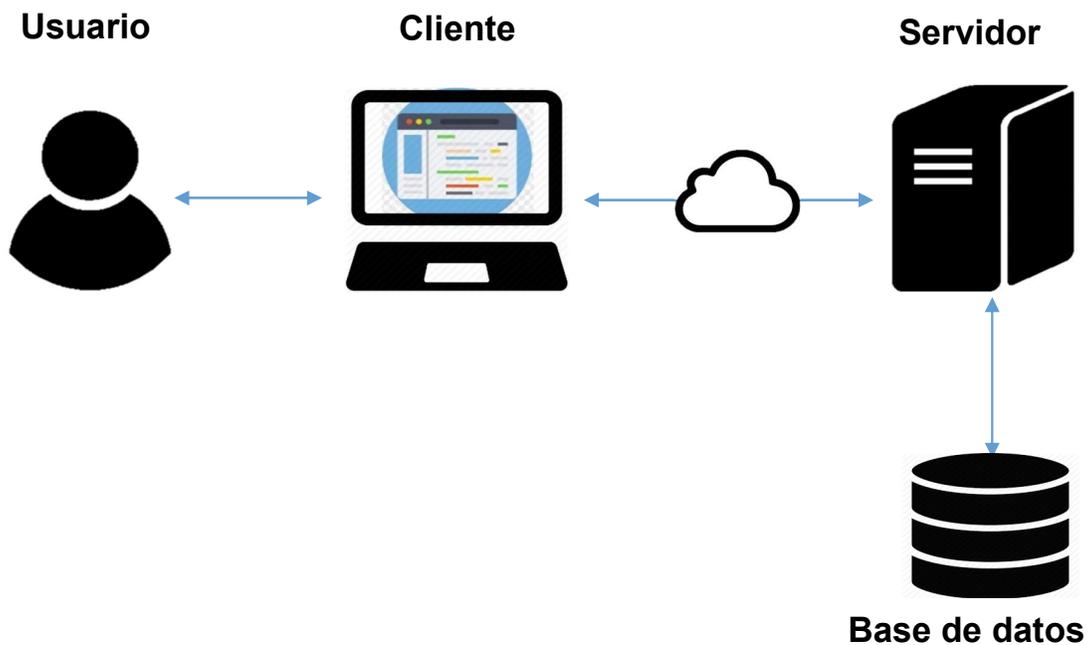


Ilustración 4 : Arquitectura conceptual de la solución.

1.6.2. Arquitectura tecnológica a utilizar

La aplicación será desarrollada sobre el framework Sails, el cual permite realizar aplicaciones web de nivel empresarial siendo el marco de aplicación MVC más popular para NodeJS. Sails permite desarrollar el front-end con cualquier tecnología, lo que nos permite apoyarnos en AngularJS, que cuenta con un conjunto de herramientas para crear un marco de desarrollo adecuado, además que es totalmente extensible, funciona bien con otras bibliotecas y es basado en JavaScript.

Ahora bien, los servicios serán de tipo REST, permitiendo mantenerse y desarrollar nuevas funcionalidades si así lo requiere el sistema a futuro; como formato de intercambio de datos se propone el uso de JSON por su simpleza y gran desempeño para el desarrollo de servicios web, además de valorar la compatibilidad absoluta que se tiene con Angular. Sails está basado en ExpressJS que a su vez está fundamentado en NodeJS. ExpressJS con miles de métodos de utilidad HTTP y middleware a su disposición, permite la creación de una API sólida, rápida y sencilla. Sails también agrupa un potente ORM llamado Waterline que proporciona una capa de acceso simple a los datos, que funciona sin importar que base de datos estemos usando. Apoyados en esta premisa usaremos MongoDB como base de datos que guarda estructuras de datos Bson, que es una especificación similar a JSON.

Todas estas tecnologías antes mencionadas forman un conjunto de subsistemas de software para el desarrollo de aplicaciones basadas en JavaScript, integrándose exitosamente de manera autosuficiente. Además, que podemos seguir un flujo total de datos en formatos JSON desde Front-end hasta el almacenamiento en base de datos.

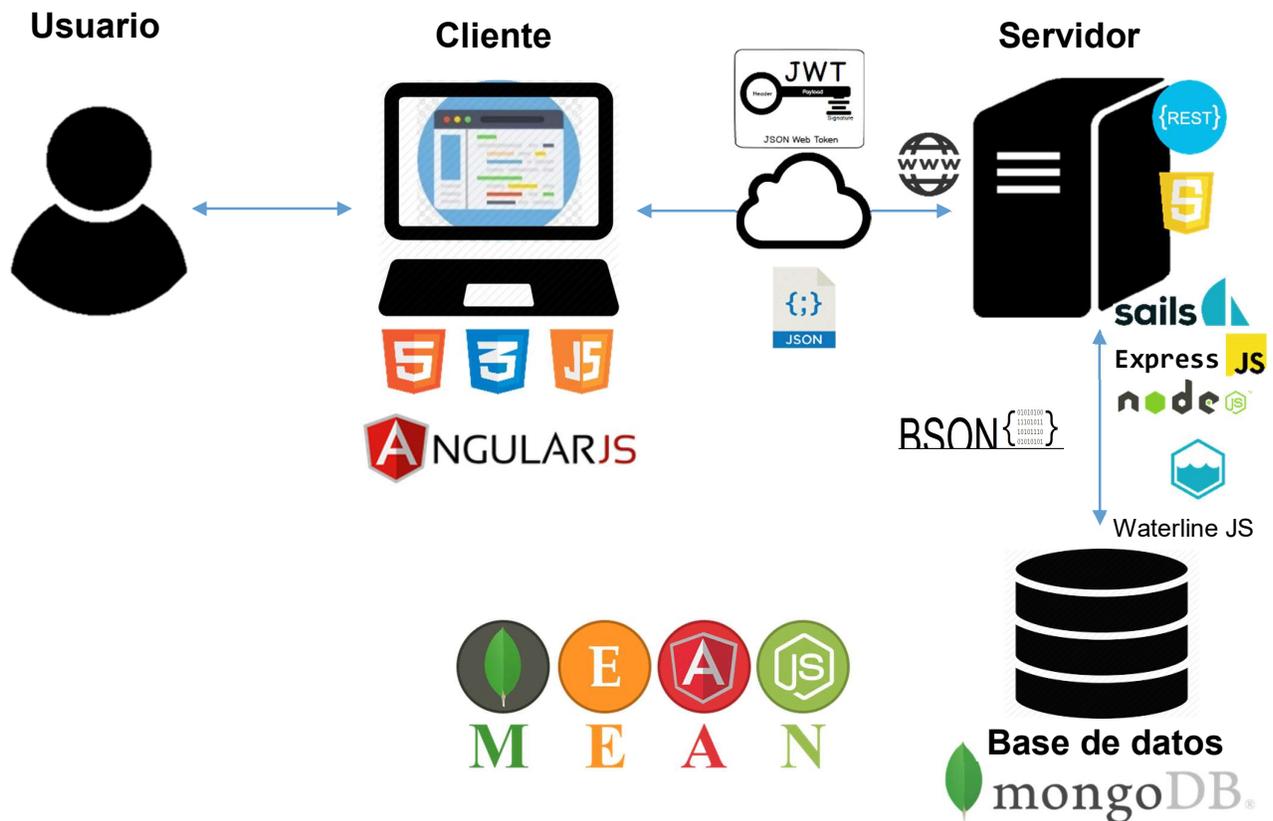


Ilustración 5: Arquitectura tecnológica de la solución.

1.6.3. Metodología de desarrollo a utilizar

Para guiar el desarrollo de este proyecto se propone el uso de la metodología de desarrollo ágil SCRUM, por su facilidad de adaptación diseñada para ofrecer un valor significativo de forma rápida en todo proyecto. La misma permite presentar soluciones parciales mediante el uso de iteraciones, lo que hará que existan pruebas que contribuyan con el progreso del desarrollo de la aplicación al poder obtener retroalimentación de parte del cliente.

1.6.4. Descripción del flujo asociado a la solución

Las soluciones propuestas representados en flujogramas muestran cómo se mejora el proceso de la problemática que queremos atacar y fortalecer para brindar un mejor servicio a los clientes.

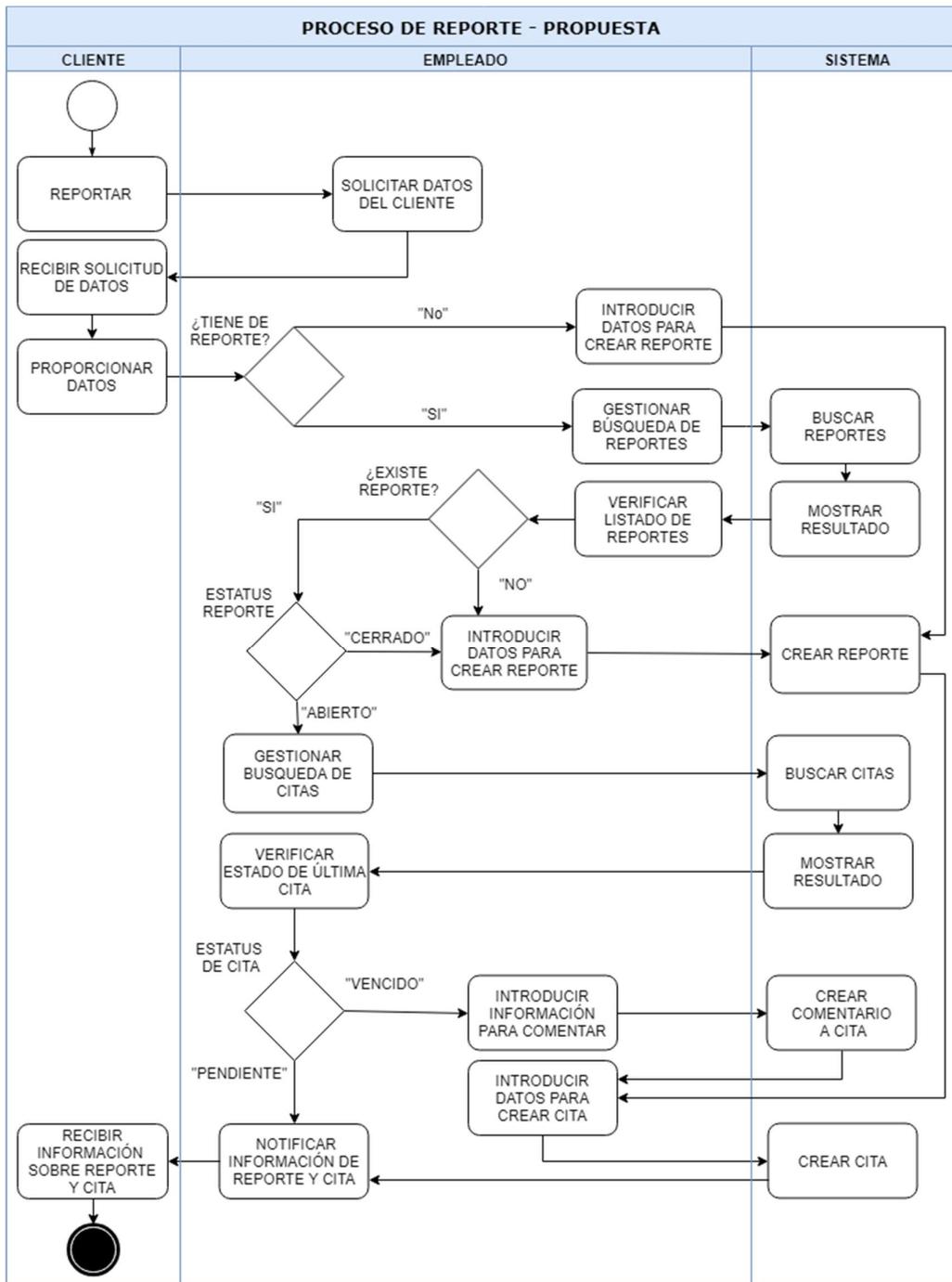


Ilustración 6: Flujograma del proceso propuesto de reporte

En la ilustración 6, se detalla en el proceso de reporte por parte de un cliente: Comienza cuando un cliente reporta un inconveniente con su servicio contactando a la compañía proveedora de servicio internet, un empleado de la compañía es quien atiende al cliente y solicita información como el número de contrato del cliente y número de reporte en caso de tenerlo. Si el cliente posee reporte, el empleado hace una búsqueda en el sistema con el número de contrato y el número de

reporte para detallar la información correspondiente, el empleado verifica si el reporte existe, luego se comprueba si el reporte está en estatus abierto, de ser así se gestiona la búsqueda de las citas que posee el reporte, se verifica el estatus de la última cita, si está en estatus pendiente, el empleado le notifica al cliente la información de reporte y cita.

El empleado crea un nuevo reporte si: el cliente no posee reporte; el reporte no existe; el reporte está en estatus cerrado o la última cita del reporte se encuentra en estatus vencido o cerrado. Posteriormente se crea una cita para ese reporte, se le informa al cliente su número de reporte generado y la fecha en la que se fijó la cita con el técnico de la compañía para ir a resolver su problema.

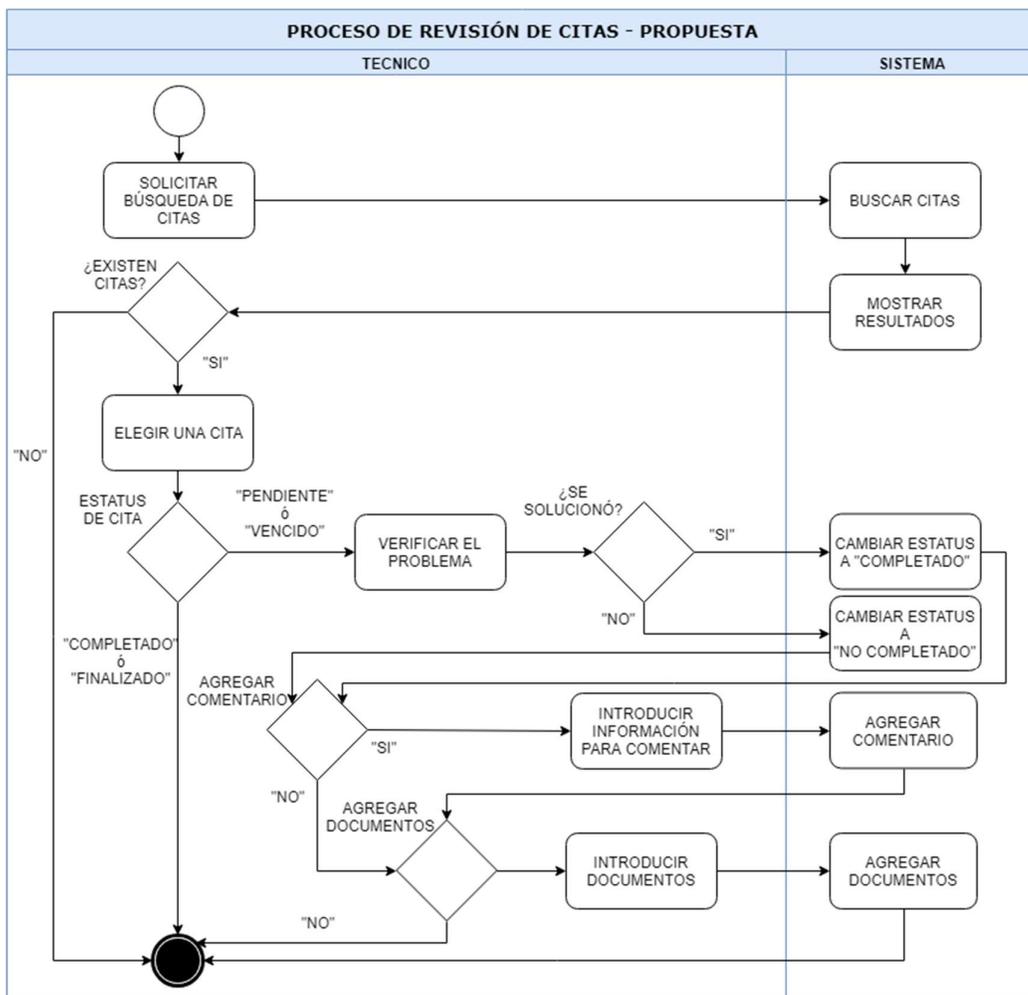


Ilustración 7: Flujograma del proceso propuesto de revisión de citas

En la ilustración 7, se especifica el proceso de revisión de citas por parte de un técnico perteneciente a la compañía proveedora de servicio internet: Comienza con el técnico realizando una búsqueda en el sistema de sus citas, si no existen citas se termina el proceso, si existen citas se elige una, posteriormente se verifica el estatus de la cita, si está en estatus completado o finalizado, y el técnico termina el proceso.

Cuando el estatus de la cita es pendiente o vencido el técnico verifica si el problema está solucionado, en caso de estar solucionado el problema, el técnico mediante el sistema pasa la cita a estatus completado, si no se solucionó el problema se pasa a estatus no completado. Así mismo el técnico luego de cambiar el estatus de la cita decide si agregar un comentario, de esta misma forma si tiene documentos que sumen información importante a la cita los podrá adjuntar, terminando el proceso de revisión de citas por parte del técnico.

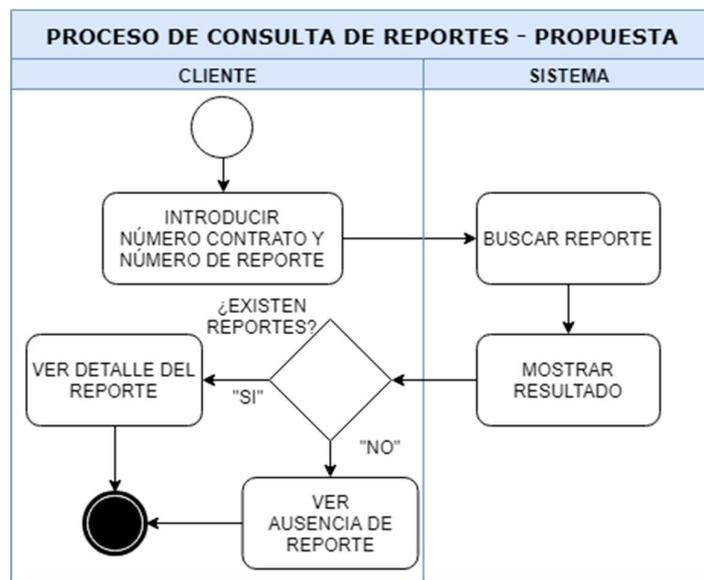


Ilustración 8: Flujograma del proceso propuesto de consulta de reportes.

Por último, en la ilustración 8, se detalla el proceso de revisión de citas por parte de un cliente: Inicia con el cliente navegando a la página web de la compañía donde en la sección de reportes podrá introducir su número de contrato que lo identifica como cliente de la compañía y el número de reporte. El sistema hace una búsqueda con los datos proporcionados y muestra la información sobre el reporte terminando el proceso, si el reporte no existe se le muestra un mensaje en la pantalla al cliente indicando lo que ocurrió.

1.7. Alcance

- Crear la base de datos que soportará todo el modelo de datos del proceso de negocio.
- Crear servicios de datos que permitan crear, modificar, eliminar y actualizar los registros de la base de datos.
- Contemplar las permisologías y roles para la autenticación de usuarios.
- Administrar usuarios en el sistema como: administrador, empleado, técnico y cliente.
- Registrar reporte por parte del empleado.
- Asignar cita entre un cliente y un técnico para un reporte en una fecha específica de manera automática.
- Consultar historial de citas y reportes de un cliente.
- Detallar la información correspondiente a una cita.
- Consultar citas asignadas para un técnico.
- Comentar cita por parte del técnico o empleado, de manera que se proporcione información útil para la cita.
- Gestión de documentos relacionados con una cita.
- Cambiar estado de una cita por parte del empleado y del técnico.
- Cambiar estado del reporte de manera automática.
- Consulta de estado del Reporte por parte del cliente por medio de un número de identificación de reporte.
- Envío de correo automático al registrar un nuevo usuario, crear un reporte, crear una cita o cambiar estado de la cita.
- Notificar al técnico cuando se sea asignado una cita.

CAPÍTULO 2: MARCO CONCEPTUAL

En este capítulo se describen teorías, conceptos, procesos y plataformas tecnológicas relacionadas con el problema a resolver.

2.1. Sistema de Información

2.1.1. Definición

De acuerdo con (Laudon & Laudon, 2012) Podemos plantear la definición técnica de un sistema de información como “un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización”.

Respaldados en este concepto podemos puntualizar que los sistemas de información son una necesidad, ya que las empresas manejan grandes cantidades de datos que pueden ser analizados, de manera que se pueda encontrar la información relevante para tomar diferentes decisiones que permitan tomar ventajas competitivas y así posicionarse de manera estratégica en la competencia.

2.1.2. Tipos de Sistemas de Información

Existen varios tipos de Sistemas de Información en función a las diferentes actividades, áreas y necesidades dentro de una organización. Desde el punto de vista organizacional se dividen en niveles estratégicos, administración, conocimiento y operativo.



Ilustración 9: Clasificación de los sistemas de información.

Fuente: <http://www.rafaelmellado.cl/material/inf3242/complemet/01.pdf>

2.1.2.1. Sistemas de procesamiento de transacciones (TPS)

Logran la automatización de los procesos operativos dentro de una organización, su función primordial consiste en procesar transacciones rutinarias, eliminando el trabajo tedioso de las transacciones operacionales y como resultado reducir el tiempo empleado en ejecutarlas. Por lo general requieren un volumen considerable de E/S de información, pero sus cálculos, procesos y métodos suelen ser poco sofisticados.

Es el primer tipo de sistema de información que se implanta en las organizaciones que van evolucionando hacia las tecnologías de la información, por ende, se encuentran en el nivel operativo que es el más bajo de la pirámide, debido a que son la base de los siguientes niveles. Sin un sistema operacional, los otros sistemas pierden su sentido.

2.1.2.2. Sistemas de gestión del conocimiento (KWS)

Son sistemas de información que Auxilian a los trabajadores en la creación e integración de nuevo conocimiento en el área operativa. Por lo general este nivel está conformado por personal profesional (ingenieros, científicos, programadores) que generan o diseminan información creando documentación, productos y sistemas. Operan en el nivel de conocimiento con la finalidad de ayudar a la organización a integrar nuevos conocimientos para que la organización controle el flujo de la documentación.

2.1.2.3. Sistemas de información de Gestión (MIS)

Son un tipo de sistemas de información que recopilan y procesan información de diferentes fuentes para ayudar en la toma de decisiones en lo referente a la gestión de la organización.

Los sistemas de información de gestión utilizan los datos recogidos por el TPS para proporcionar a los supervisores los informes de gestión y estadísticas de control necesarios para utilizarlos como apoyo a las actividades de gestión y la toma de decisiones.

Este nivel contiene los sistemas informáticos que están destinados a ayudar a los gerentes de nivel medio a la gestión operativa en la supervisión y control de las actividades de procesamiento de transacciones que se producen a nivel administrativo.

2.1.2.4. Sistema de apoyo a la toma de decisiones (DSS)

Es un sistema basado en ordenadores destinado a ser utilizado por usuarios de rango, un gerente particular o por un grupo de gerentes a nivel organizacional administrativo para tomar una decisión en el proceso de resolver una problemática semiestructurada.

Los sistemas de apoyo a la toma de decisiones ayudan al gerente en la toma de decisiones cuando necesita modelar, formular, calcular, comparar, seleccionar la mejor opción o predecir los escenarios. favorece al equipo directivo a tomar decisiones en situaciones en las que existe incertidumbre sobre los posibles resultados o consecuencias.

2.1.2.5. Sistema de información ejecutiva (EIS)

Proporcionan un acceso rápido a la información interna y externa, presentada a menudo en formato gráfico, pero con la capacidad de presentar datos básicos más detallados si es necesario. Los sistemas información ejecutiva proporcionan información crítica de una amplia variedad de fuentes internas y externas en formatos fáciles de usar para ejecutivos y gerentes.

Un sistema de información ejecutiva proporciona a los altos directivos un sistema para ayudar a tomar decisiones estratégicas. Está diseñado para generar información que sea lo suficientemente abstracta como para presentar toda la operación de la empresa en una versión simplificada para satisfacer a la alta dirección.

2.2. Tecnología de desarrollo web

2.2.1. Internet

Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, lo cual garantiza que las redes físicas heterogéneas que la componen, formen una red lógica única de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California (Estados Unidos).

Uno de los servicios que más éxito ha tenido en internet ha sido la World Wide Web (WWW o la Web), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Esta fue un desarrollo posterior (1990) y utiliza internet como medio de transmisión.

2.2.2. World Wide Web

En informática, la World Wide Web (WWW) o red informática mundial¹ es un sistema de distribución de documentos de hipertexto o hipermedia interconectados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener textos, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces. (Fundeu BBVA Fundación del español urgente, 2012)

2.2.3. HTTP

Es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de sesión, y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado. (W3C, 1999)

2.2.4. HTML

HTML, por sus siglas en inglés HyperText Markup Language, que significa Lenguaje de Marcas de Hipertexto se hace relación con la elaboración de páginas web. Define una estructura jerárquica de árbol, donde sus nodos son etiquetas que consisten en breves instrucciones de comienzo y final que el navegador interpreta, esta determina la forma en la que debe aparecer en la pantalla del ordenador el texto, imágenes, videos y los demás elementos que pueden conformar una página web. Los archivos con este lenguaje de etiquetas tienen extensión “.html”. (Gutiérrez E. G.)

HTML, es un estándar el cual está actualmente siendo gestionado por la W3C (World Wide Web Consortium), una comunidad internacional que se dedica al desarrollo de estándares que aseguran el crecimiento de la Web. Cabe destacar, su versión más reciente y la que más uso tiene es HTML5.

2.2.5. CSS

CSS (siglas en inglés de Cascading Style Sheets), en español "Hojas de estilo en cascada", es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas. Los archivos con este lenguaje de estilos tienen extensión “.css”. (Mozilla, s.f.)

2.2.6. Less

Less, es un dinámico lenguaje de hojas de estilo que puede ser compilado en hojas de estilo en cascada (CSS) y ejecutarse en el lado del cliente o en el lado del servidor. Diseñado por Alexis Sellier. LESS, es de código abierto. La sintaxis indentada de Less es un metalenguaje anidado, lo que es válido en CSS es válido en SCSS con la misma semántica. LESS proporciona los siguientes mecanismos: variables, anidamiento, operadores, mixins y funciones. Los archivos con este lenguaje de etiquetas tienen extensión “. less”.

2.2.7. Bootstrap

Bootstrap, es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end. (Twitter, s.f.)

2.2.8. JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargando junto con el código HTML. (Domínguez-Dorado, 2008)

2.2.9. AJAX

Acronimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el

servidor en segundo plano. De esta forma, es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

2.2.10. JWT

Es un estándar abierto basado en JSON para crear tokens de acceso que afirman un cierto número de peticiones. Por ejemplo, un servidor podría generar un token que tiene el reclamo "registrado como administrador" y proporcionárselo a un cliente. El cliente podría usar ese token para probar que está conectado como administrador. Los tokens están firmados por la clave privada de una parte (generalmente la del servidor), de modo que ambas partes (la otra ya está, por algún medio adecuado y confiable, en posesión de la clave pública correspondiente) son capaces de verificar que el token es legítimo. Los tokens están diseñados para ser compactos, URL -seguro, y utilizables especialmente en un contexto de inicio de sesión único (SSO) del navegador web. Las reclamaciones de JWT se pueden usar normalmente para pasar la identidad de los usuarios autenticados entre un proveedor de identidades y un proveedor de servicios, o cualquier otro tipo de reclamaciones según lo requieran los procesos comerciales.

2.2.11. JSON

JSON (acrónimo de JavaScript Object Notation, notación de objeto de JavaScript) es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript, aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente. (Kapic Edin, 2016)

2.2.12. AngularJS

AngularJS, es un framework estructural para páginas web dinámicas. Permite usar HTML como lenguaje base para luego extender la sintaxis de HTML y darles más funcionalidad y dinamismo a los componentes de una aplicación web. Inicialmente basado en el patrón MVC (Modelo – Vista – Controlador), usa JavaScript por lo que es ejecutado en el lado del cliente o navegador. (Google, 2018).

2.2.13. Servicio Web

Un servicio web (web service, en inglés) es un estándar de interoperabilidad entre distintas plataformas, utiliza un conjunto de protocolo y estándares que trabajan conjuntamente para intercambiar datos entre aplicaciones, garantizando una comunicación segura. Cualquier aplicación de software desarrollada en un lenguaje de programación particular y ejecutada sobre cualquier

plataforma, puede utilizar servicio web para intercambiar datos en una red, como por ejemplo internet. Existen dos protocolos que utilizan los servicios web para la comunicación y que se explica a continuación.

2.2.14. Protocolo SOAP

SOAP (originalmente las siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por Dave Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

Básicamente SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más complejos y completos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir. Este protocolo está basado en XML y se conforma de tres partes: Sobre (envelope): el cual define qué hay en el mensaje y cómo procesarlo; conjunto de reglas de codificación para expresar instancias de tipos de datos; la convención para representar llamadas a procedimientos y respuestas.

2.2.15. XML

XML, es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de recomendación W3C como lenguaje de marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que debes definir tus propias etiquetas. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet. Hay muchos lenguajes basados en XML; Algunos ejemplos son XHTML, MathML, SVG, XUL, XBL, RSS, y RDF. También puedes crear uno propio. (Mozilla, s.f.)

2.2.16. Protocolo REST

Por sus siglas en inglés, Representational State Transfer, que significa Transferencia de Estado Representacional, plantea una arquitectura cliente-servidor, en el cual un servicio es visto como un recurso que se identifica a través de una dirección URL, mediante la cual puede ser accedido o consumido. Para acceder a estos servicios web se hace uso de mensajes en formato simple, los cuales se intercambian entre el cliente y el servidor.

El protocolo REST define a partir de HTTP, cuatro métodos; GET, PUT, DELETE y POST. Este protocolo se basa en el envío de mensajes a través de diversos formatos y lenguajes; XML, HTML, JSON, debido a la sencillez de los mensajes JSON este es el tipo de mensaje más utilizado en el protocolo.

Cada mensaje intercambiado contiene la información necesaria para cada funcionamiento de los servicios, de tal forma que para cada servicio el cliente y el servidor conocen el formato o protocolo interno de los mensajes.

2.2.17. NodeJS

NodeJS, es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

NodeJS, es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y Java EE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. NodeJS implementa algunas especificaciones de CommonJS. NodeJS incluye un entorno REPL para depuración interactiva.

2.2.18. ExpressJS

ExpressJs, o simplemente Express, es un marco de aplicación web para NodeJS, lanzado como software libre y de código abierto bajo la Licencia MIT. Está diseñado para construir aplicaciones web y API's. Se ha llamado el marco de servidor estándar para NodeJS. (Eneko, s.f.)

El autor original, TJ Holowaychuk, lo describió como un servidor inspirado en Sinatra, lo que significa que es relativamente mínimo con muchas funciones disponibles como complementos. Express es la parte de fondo de la pila MEAN, junto con la base de datos MongoDB y el marco de front-end de AngularJS.

2.2.19. SailsJS

SailsJS, es un framework MVC para NodeJS pensado para crear aplicaciones modernas y escalables, destaca a la hora de crear aplicaciones en tiempo real ya que incorpora websockets, aunque puede ser usado para crear cualquier tipo de aplicación.

Otra de las grandes virtudes de SailsJS es su capacidad de gestionar los modelos de forma independiente, es decir, podemos crear un modelo que ataque a una tabla posts en MySQL y otro modelo que ataque una tabla users en Postgres o un documento en Mongo.

Para el manejo de las templates SailsJS utiliza por defecto ejs, aunque no es obligatorio, ejs es sencillo y permite utilizar HTML.

2.2.20. Waterline

Es un nuevo tipo de motor de almacenamiento y recuperación. Proporciona una API uniforme para acceder a elementos de diferentes tipos de bases de datos, protocolos y API de terceros. Eso significa que escribes el mismo código para obtener y almacenar cosas como usuarios, ya sea que vivan en Redis, MySQL, LDAP, MongoDB o Postgres.

Waterline se esfuerza por heredar las mejores partes de los ORM, como ActiveRecord, Hibernate y Mongoose, pero con una perspectiva fresca y énfasis en el modularidad, la capacidad de prueba y la consistencia entre los adaptadores.

2.2.21. Web Socket

Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

2.2.22. CRONjob

Los CRONJobs, o tareas programadas, permiten ejecutar procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Si un proceso se debe ejecutar automáticamente con cierta frecuencia se puede establecer la periodicidad configurando un CRONJob.

2.2.23. MEAN

MEAN Stack (acrónimo para: MongoDB, ExpressJS, AngularJS, NodeJS), es un framework o conjunto de subsistemas de software para el desarrollo de aplicaciones, y páginas web dinámicas, que están basadas, cada una de estas en el popular lenguaje de programación conocido como

JavaScript. Gracias a esta característica el conjunto se integra exitosamente en una plataforma autosuficiente, es decir que las aplicaciones pueden ser escritas en JavaScript tanto para el lado del servidor y del lado del cliente entornos de ejecución.

2.3. Base de Datos

Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. Una base de datos (sea cual sea) es un soporte digital que tiene como fin el almacenamiento masivo de información en formato texto plano. No es capaz de almacenar imágenes como se cree, sino que almacena las rutas (path) de dichas fotos; ni almacena otro tipo de datos; sino que almacena sus rutas de acceso de ser necesario.

Las bases de datos, son utilizadas en sistemas que requieren una interacción fluida con la aplicación; estas se encargan muchas veces de administrar, editar, y dar de alta. Usualmente la base de datos, está ligada a la programación directa del sitio, causando que una edición en ella cause una modificación directa en lo que ve el usuario.

2.3.1. Sistema manejador de Base de Datos

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

2.3.2. Base de Datos SQL

Es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo. Los datos deben de estar relacionados y estructurados de acuerdo con un modelo capaz de recoger el contenido semántico de los datos almacenados. Dada la importancia que tienen en el mundo real las relaciones entre los datos, es imprescindible que la base de datos sea capaz de almacenar estas interrelaciones. Además, las bases de datos modernas también almacenan las

restricciones que están presentes en los datos y a las que se les está concediendo una importancia creciente.

2.3.3. Base de Datos NoSQL

En informática, NoSQL (a veces llamado "no sólo SQL") es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad), y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no sólo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL. (ADAM LITH, 2010)

2.3.3.1. Redis

Base de datos creada por Salvatore San Filippo y Pieter Noordhuis y está apoyado por VMware. Se trata de una base de datos del tipo clave-valor. Se puede imaginar como un array gigante en memoria para almacenar datos, datos que pueden ser cadenas, hashes, conjunto de datos o listas.

Tienen la ventaja de que sus operaciones son atómicas y persistentes. Por ponerle una pega, Redis no permite realizar consultas, solo se puede insertar y obtener datos, además de las operaciones comunes sobre conjuntos. Creado en ANSI C, por lo tanto, es compatible y funciona sin problemas en sistemas Unix, Linux

2.3.3.2. MongoDB

Base de datos creada por 10gen del tipo orientada a documentos, de esquema libre, es decir que cada entrada puede tener un esquema de datos diferente que nada tenga que ver con el resto de registro almacenados. Es bastante rápido a la hora de ejecutar sus operaciones ya que estos datos binarios. En poco tiempo, MongoDB se ha convertido en una de las bases de datos NoSQL favoritas por los desarrolladores escrito en lenguaje C++. Para el almacenamiento de la información, utiliza un sistema propio de documento conocido como JSON, pero con la peculiaridad de que puede almacenar. El código binario está disponible para los sistemas operativos Windows, Linux, OS X y Solaris.

2.3.4. Cuadro Comparativo Bases de datos SQL Y NoSQL

Base de Datos SQL	Base de Datos NoSQL
No existe redundancia en los datos.	Sin importar cuanto optimicemos una base de datos NoSQL, siempre existirá redundancia en los datos.
Atomicidad de las operaciones en la base de datos. Esto es, que en estas bases de datos o se hace la operación entera o no se hace utilizando la famosa técnica del rollback.	No todas las bases de datos NoSQL contemplan la atomicidad de las instrucciones y la integridad de los datos. Soportan lo que se llama consistencia eventual.
La atomicidad de las operaciones juega un papel crucial en el rendimiento de las bases de datos.	Gana más rendimiento del sistema mediante la eliminación de una gran cantidad de comprobaciones de integridad.
La escalabilidad en estas bases de datos resulta más complicada y en muchos casos requiere grandes máquinas.	Escalabilidad horizontal: son capaces de crecer en número de máquinas, la escalabilidad y su carácter descentralizado soportan estructuras distribuidas.
Los datos deben cumplir requisitos de integridad tanto en tipo de dato como en compatibilidad.	Suelen ser bases de datos mucho más abiertos y flexibles. Permiten adaptarse a necesidades de proyectos fácilmente.
Las bases de datos relacionales siguen estándares.	Las bases de datos NoSQL tienen pocas normas en común. Cada base de datos NoSQL tiene su propia API, las interfaces de consultas son únicas y tienen peculiaridades.
Debido al largo tiempo que llevan en el mercado, estas herramientas tienen un mayor soporte, mejores suites de productos y características adicionales para gestionar estas bases de datos. Además de una gran comunidad de desarrolladores.	La falta de experiencia. - La novedad de NoSQL significa que no hay una gran cantidad de desarrolladores y administradores que conocen la tecnología lo que hace difícil a las empresas encontrar personas con los conocimientos técnicos apropiados.

Tabla 1: Cuadro Comparativo Bases de datos SQL y NoSQL.

CAPÍTULO 3: MARCO METODOLÓGICO

En este capítulo se presentará todo lo relacionado a la investigación metodológica necesaria para cumplir con la ejecución exitosa del presente trabajo asociado a la creación de una aplicación web que sirva para el manejo y programación de citas técnicas de un proveedor de servicio de internet, teniendo de este modo un desarrollo de software controlado, pautando así el alcance que este debería llevar.

Un proceso de desarrollo de software consiste en la realización de un conjunto de actividades y resultados asociados, a partir de los cuales se obtiene un producto de software.

3.1. Metodología tradicional Base de Datos

Teniendo en cuenta la filosofía de desarrollo de las metodologías, aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de Metodologías Tradicionales.

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. A continuación, se mencionan algunas metodologías tradicionales:

- **Metodología RUP (Rational Unified Process):** Se basa en un marco de procesos de trabajo que pueden ser adaptados por parte de las organizaciones que desarrollen el producto. Esta metodología se compone de tres módulos principales: Inicio para establecer el objetivo del sistema e identificar los requerimientos. Elaboración, se genera una estructura arquitectónica que permite diseñar, implementar y probar el sistema. Fase de construcción, a partir de la arquitectura establecida anteriormente se realiza la implementación y el manejo de versiones con respecto a las funcionalidades. Transición, se comprueba que el sistema cumpla con las necesidades.

- **Metodología en Cascada:** El ciclo de vida en cascada se caracteriza porque todas las fases se realizan de forma secuencial, pero una etapa no puede comenzar sin que la etapa anterior esté finalizada. La ventaja de esta metodología es que posee una planificación sencilla, la calidad del producto obtenido es alto y no requiere de un personal altamente calificado. Sin embargo, para el desarrollo es necesario tener todos los requerimientos iniciales.

3.2. Metodología ágil

El software es parte de casi todas las operaciones de negocio, por lo que es fundamental que el software nuevo se desarrolle rápidamente para aprovechar nuevas oportunidades y responder a la presión competitiva. Actualmente el desarrollo y entrega de manera rápida son los requerimientos más críticos de los sistemas. Los procesos de desarrollo del software basados en una completa especificación de los requerimientos, diseño, construcción y pruebas del sistema no se ajustan al desarrollo rápido de aplicaciones. Cuando los requerimientos cambian o se descubren problemas con ellos, el diseño o implementación del sistema se tiene que volver a realizar o probar. Como consecuencia, normalmente se prolonga en el tiempo un proceso en cascada convencional. Las metodologías ágiles surgen como una alternativa a las metodologías tradicionales. Las metodologías ágiles combinan una filosofía y un conjunto de directrices de desarrollo. La filosofía busca la satisfacción del cliente y la entrega temprana de software incremental, equipos pequeños con alta motivación, métodos informales y una simplicidad general del desarrollo. De hecho, muchas organizaciones están dispuestas a obtener una pérdida en la calidad del software y en el compromiso sobre los requerimientos en favor de una entrega rápida del software.

Los procesos de desarrollo rápido de software están diseñados para producir software útil de forma rápida. Generalmente, son procesos interactivos en los que se entrelazan la especificación, el diseño, el desarrollo y las pruebas. A continuación, se mencionan algunas metodologías ágiles:

- **SCRUM:** Esta metodología tiene como base la creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones, pero en Scrum se llaman "Sprints". Los Sprints son la base de la metodología Scrum, este componente se considera un contenedor de eventos que permite obtener una versión estable del producto. Las ventajas de esta metodología es la flexibilidad a cambios entre cada uno de los Sprints, la necesidad de obtener una versión funcional en cada iteración permite obtener un software de alta calidad, la realización de funcionalidades de mayor prioridad permite la reducción de riesgos de manera anticipada.
- **XP- eXtreme Programming:** Es la primera metodología ágil, está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo, promoviendo el trabajo en equipo. XP se basa en la retroalimentación continua entre el cliente y el equipo de

desarrollo y la comunicación fluida entre los participantes, sin embargo, la particularidad de esta metodología es el alto nivel de disciplina de las personas que participan en el proyecto. XP es una metodología muy popular, además es considerada una metodología posmoderna debido a que las capacidades se generan a través de procesos emergentes.

3.3. Comparación entre las metodologías de desarrollo

De acuerdo con (Amaro Calderón & Valverde Rebaza, 2007), en la Tabla 2 se establece un cuadro comparativo entre las metodologías tradicionales y las metodologías ágiles, las diferencias entre estas metodologías afectan no sólo al proceso de desarrollo, sino también al contexto del equipo como una organización.

Metodologías tradicionales	Metodologías ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.	Basadas en heurísticas provenientes de prácticas de producción de código.
Cierta resistencia a los cambios.	Esencialmente preparadas para cambios durante el proyecto.
Proceso mucho más controlado, con numerosas políticas/normas.	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos.	Pocos artefactos.
Muchos roles.	Pocos roles.
La arquitectura del software es esencial y se expresa mediante modelos.	Menos énfasis en la arquitectura del software.

Tabla 2: Metodologías tradicionales y Metodologías ágiles.

Fuente: Metodologías Ágiles. Amaro, Sarah & Valverde, Jorge (2007).

3.4. SCRUM

Es un marco de trabajo de procesos usado para gestionar el desarrollo de productos complejos, Scrum no es un proceso o una técnica para construir productos sino un marco de trabajo en el cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las

prácticas de gestión de producto y las prácticas de desarrollo de modo que se pueda mejorar, según (Schwaber & Sutherland, 2016). Scrum es liviano, fácil de entender, pero difícil de llegar a dominar.

3.4.1. Teoría de Scrum

De acuerdo con (Schwaber & Sutherland, 2016) Scrum se basa en la teoría de control de procesos empírica o empirismo, donde el empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce, mediante un enfoque iterativo o incremental para optimizar la predictibilidad y el control del riesgo.

3.4.2. El equipo Scrum

Los equipos Scrum son auto-organizados y multifuncionales, el modelo de equipo está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

3.4.2.1. Dueño de producto (Product Owner)

Es el responsable de maximizar el valor del producto y el trabajo del equipo de desarrollo, es la única persona responsable de gestionar la lista del producto (Product Backlog). Se encarga de expresar claramente los elementos de la lista del producto, ordenar los elementos de la lista del producto, optimizar el valor del trabajo que el equipo de desarrollo realiza, asegurar que la lista del producto sea visible transparente y clara para todos y asegurar que el equipo de desarrollo entiende los elementos de la lista del producto.

3.4.2.2. Equipo de desarrollo (Development Team)

Consiste en los profesionales que realizan el trabajo de entregar un incremento de producto “terminado” que potencialmente se pueda poner en producción al final de cada Sprint, solo los miembros del equipo de trabajo participan en la creación del incremento.

Los equipos de desarrollo son auto-organizados, multifuncionales dado que cuentan con todas las habilidades necesarias para crear un incremento del producto, todos son desarrolladores independientemente del trabajo de cada persona, no hay sub-equipos en estos equipos ni importar los dominios particulares que se requieran a tener en cuenta y la responsabilidad recae en el equipo de desarrollo como un todo.

3.4.2.3. Scrum Master

Es el responsable de asegurar que Scrum se entienda y se adopte, asegurándose que el equipo de Scrum ajuste la teoría, prácticas y reglas de Scrum en su proceso. Es un líder que está

al servicio del equipo Scrum ayudando a las personas externas al equipo Scrum a entender qué interacciones con el equipo Scrum pueden ser útiles y cuáles no.

El Scrum Master da servicios al dueño de producto, al equipo de desarrollo y a la organización de distintas maneras y cada uno de sus servicios depende de a qué miembro del equipo Scrum esté interactuando.

3.4.3. Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum, todos los eventos son bloques de tiempo (time-boxes), de modo tal que todos tienen una duración máxima asociada.

3.4.3.1. Sprint

Es la base de Scrum, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “terminado” utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.

Durante el Sprint no se realizan cambios que puedan afectar el objetivo del Sprint (Sprint Goal) y los objetivos de calidad no disminuyen. Cada sprint puede considerar un proyecto con un horizonte no mayor de un mes.

3.4.3.2. Planificación de Sprint (Sprint Planning)

La planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes, se debe realizar un plan mediante el trabajo colaborativo del equipo Scrum completo. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, además de enseñar al equipo Scrum a mantenerse en el bloque de tiempo estipulado. La planificación de Sprint responde a las siguientes preguntas:

- a. **¿Qué puede hacerse en este Sprint?:** El equipo de desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. La entrada de esta reunión está constituida por la lista de producto, el último incremento de producto, la capacidad proyectada del equipo de desarrollo para el sprint y el rendimiento pasado del equipo de desarrollo. Después que el equipo de desarrollo planea los elementos de la lista de producto que entregará en el Sprint, el equipo de desarrollo elabora un objetivo del Sprint (Sprint Goal), el objetivo del Sprint debería lograrse durante el Sprint a través de la implementación de la lista del producto y proporciona una guía al equipo de desarrollo de por qué se está construyendo el incremento.

- b. **¿Cómo se conseguirá completar el trabajo seleccionado?:** El equipo de desarrollo decide cómo construirá esta funcionalidad para formar un incremento de producto “terminado” durante el Sprint, los elementos de la lista de producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de lista de pendientes del Sprint (Sprint Backlog).

Al finalizar la planificación de Sprint, el equipo de desarrollo debería ser capaz de explicar al dueño de producto y al Scrum Master cómo se pretende trabajar como un equipo organizado para lograr el objetivo del sprint y crear el incremento esperado.

3.4.3.3. Objetivo del Sprint (Sprint Goal)

Es una meta establecida para el Sprint que puede lograrse mediante la implementación de la lista de producto, se construye durante la planificación del Sprint, el objetivo del Sprint brinda al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint.

3.4.3.4. Scrum diario (Daily Scrum)

Es una reunión con un bloque de tiempo de 15 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum diario y haciendo una proyección acerca del trabajo que podría contemplarse antes del siguiente. El equipo de desarrollo usa el Scrum diario para evaluar el progreso hacia el objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la lista de pendientes del Sprint.

3.4.3.5. Revisión de Sprint (Sprint Review)

Al final del Sprint se lleva a cabo una revisión para inspeccionar el incremento y adaptar la lista de producto si fuese necesario. Durante la revisión de Sprint, el equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint, basándose en esto y en cualquier cambio a la lista de producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor.

3.4.3.6. Retrospectiva de Sprint (Sprint Retrospectiva)

Es una oportunidad para el equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint, esta revisión tiene lugar después de la revisión de Sprint y antes de la siguiente planificación de Sprint. El propósito de esta revisión es:

- ✓ Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.

- ✓ Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
- ✓ Crear un plan para implementar las mejoras a la forma en la que el equipo Scrum desempeña su trabajo.

3.4.4. Artefactos de Scrum

Representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación, los artefactos están diseñados para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

3.4.4.1. Lista de producto (Product Backlog)

Es una lista ordenada de todo lo que podría ser necesario en el producto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto, esta lista evoluciona a medida que también el producto y el entorno en que se usará. La lista de producto enumera las características, funcionalidades, requisitos, mejoras y correcciones que construyen cambios a realizarse sobre el producto para entregas futuras.

3.4.4.2. Lista de pendientes del sprint (Sprint Backlog)

Es el conjunto de elementos de la lista de producto seleccionados para el Sprint, más un plan para entregar el incremento de producto y conseguir el objetivo del Sprint, esta lista es una predicción hecha por el equipo de desarrollo acerca de qué funcionalidad formará parte del próximo incremento y del trabajo necesario para entregar esa funcionalidad en un incremento “terminado”.

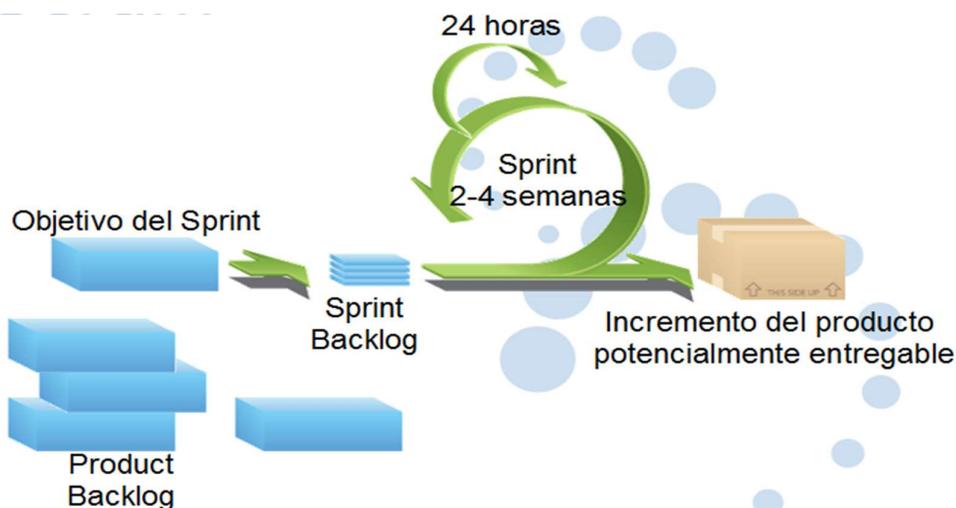


Ilustración 10: Desarrollo de software por etapas a través de Sprints.

Fuente: Grafeuille (2008).

3.5. Comparación entre metodologías ágiles

En la Tabla 3 se establece un cuadro comparativo entre las metodologías ágiles SCRUM, Extreme Programming (XP) y Kanban, las diferencias entre estas metodologías afectarán desde la estructura del equipo de desarrollo hasta la frecuencia de las entregas.

SCRUM	XP
La duración de un Sprint tiende a ser larga, de dos semanas a un mes.	Iteraciones con períodos cortos de tiempo.
Roles definidos como Scrum Master, Product Owner y el Team Member.	Ciertos roles predeterminados como Customer.
Útil en situaciones donde el trabajo puede ser dividido en lotes independientes.	Se enfoca en los ítems siguiendo una estricta prioridad.
No permite cambios a mitad de un Sprint establecido.	Permite cambios en la iteración siempre y cuando el equipo no haya empezado dicho ítem.

Tabla 3: Metodologías ágiles SCRUM y XP.

CAPÍTULO 4: MARCO APLICATIVO

Seguidamente, para el próximo capítulo se expone una descripción general de la solución en conjunto con las respectivas tecnologías utilizadas para su implementación. Adicionalmente, se especifica el uso de la metodología de desarrollo ágil SCRUM que se aplicó para la elaboración de este proyecto, así como los distintos entes que tienen interacción con el sistema, explicación y descripción de las interfaces de la aplicación y el conjunto de pruebas realizado que apoyaron en gran manera el desarrollo del sistema.

4.1 Descripción general de la solución

La aplicación desarrollada fue implementada haciendo uso de tecnologías web, el usuario final con rol empleado puede visualizar la lista de clientes, lista de reportes, crear reportes describiendo el problema presentado, definir citas para un reporte en una fecha específica, asignar un técnico a una cita, hacer seguimiento de un reporte con un número de identificación. A su vez el técnico se le permite obtener una lista de sus citas asignadas, hacer seguimiento, dejar comentario, subir documento y cambiar estatus de sus citas.

El cliente que reporta su inconveniente con el servicio puede consultar en la aplicación web información sobre el estado de su reporte por medio de su número de contrato y el número de reporte asignado.

La aplicación tiene como ventaja definir fechas para cada cita con la creación de un reporte para así dar respuesta a los clientes de manera eficiente, asignando a cada cita un técnico responsable. Los técnicos pueden dejar comentarios debidamente identificados, además de subir archivos referentes al caso. El seguimiento de los reportes por parte de los empleados se agiliza ya que se tendrá información detallada de quién está atendiendo el reporte y la fecha pautada para su solución.

El desarrollo fue llevado a cabo cumpliendo con la siguiente arquitectura tecnológica propuesta en la ilustración 11.

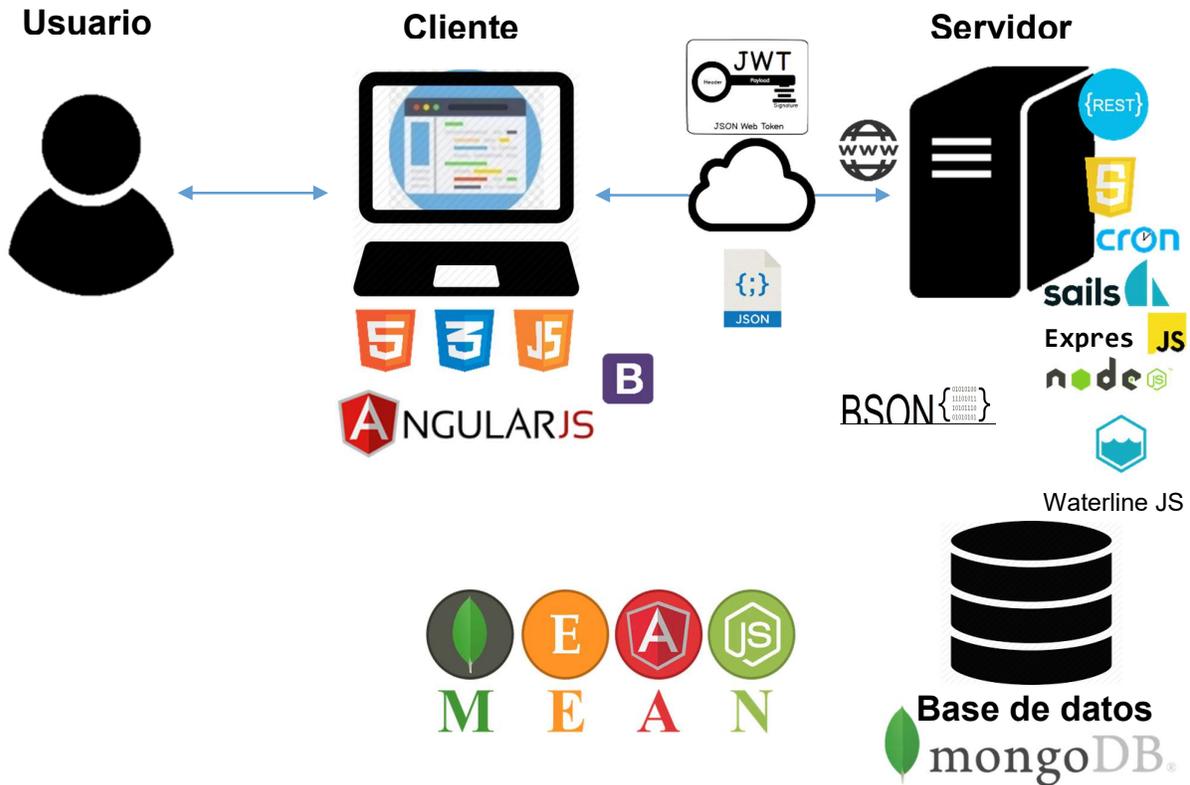


Ilustración 11: Arquitectura de la solución tecnológica.

La aplicación es desarrollada mediante un grupo de tecnologías de software que en conjunto hacen un subsistema basado en JavaScript llamado MEAN por sus siglas; MongoDB, ExpressJS, AngularJS y NodeJS, integrándose exitosamente de manera autosuficiente. MongoDB como base de datos central de la aplicación, usando estructuras de datos BSON (una especificación similar a JSON) para guardar los datos.

Para el intercambio de datos los servicios serán de tipo REST, como formato de intercambio de datos se utilizan JSON, además de valorar la compatibilidad absoluta que se tiene con Angular. La aplicación está desarrollada sobre el framework Sails que es el marco de aplicación MVC más popular para NodeJS, además Sails está basado en ExpressJS que cuenta con miles de métodos de programación de utilidad HTTP y la creación de una API sólida es rápida y sencilla. Apoyados en el ORM de Sails llamado Waterline que nos provee una capa de abstracción simple a los datos, sin dar importancia que base de datos estemos usando. Sails brinda la facilidad de desarrollar el front-end con cualquier tecnología, lo que nos permitió apoyarnos en AngularJS que cuenta con un conjunto de herramientas para crear un marco de desarrollo adecuado, además que es totalmente extensible, funciona bien con otras bibliotecas y es basado en JavaScript. Con esto seguimos un flujo total de datos en formatos JSON desde Front-end hasta el almacenamiento en base de datos.

4.2 Aplicación de la metodología SCRUM

4.2.1 Lista de Objetivos:

A continuación, en la tabla 4 se muestran las tareas o actividades programadas para cada Sprint, los cuales tuvieron cada uno un tiempo definido de dos semanas de desarrollo de proyecto.

SPRINT	ACTIVIDAD	FECHA
1	<ul style="list-style-type: none"> • Instalación de las herramientas de desarrollo necesarias. • Configuración del entorno de trabajo. • Elaboración de modelo de datos. • Creación de base de datos, documentos y referencias. • Estructuración de código y organización de carpetas. • Integración con Bootstrap. 	15/01/2019 A 18/01/2019
2	<ul style="list-style-type: none"> • Maquetado de interfaces, logo y paleta de colores. • Configuración del servidor. • Ejecución de pruebas de funcionalidad. • Pruebas de llamadas a servidor. 	21/01/2019 A 25/01/2019
3	<ul style="list-style-type: none"> • Desarrollo de CRUD de usuarios. • Desarrollo de módulo de autenticación. • Implementación de navegación entre interfaces. • Pase de datos entre interfaces. 	28/01/2019 A 05/02/2019
4	<ul style="list-style-type: none"> • Formulario para la creación de usuarios. • Validaciones de datos de entrada. • Envío de correo de confirmación de creación de cuenta en el sistema. 	07/02/2019 A 15/02/2019
5	<ul style="list-style-type: none"> • Implementación de Listas de Usuario. • Perfil de usuarios. • Edición de Datos de Usuario. 	18/02/2019 A 22/02/2019
6	<ul style="list-style-type: none"> • Integración de calendario. • Módulo de auditoria. • Permisologías dentro del sistema para cada usuario. 	25/02/2019 A 06/03/2019
7	<ul style="list-style-type: none"> • Módulo de reportes. • Módulo citas. 	07/03/2019 A 15/03/2019
8	<ul style="list-style-type: none"> • Asignación de citas entre un cliente y un técnico. • Calculo de fecha disponible, técnico disponible y cliente con reporte. 	18/03/2019 A 26/03/2019
9	<ul style="list-style-type: none"> • Integración con Dropbox para la subida de archivos en la cita, por parte de un técnico. • Cambio de estado de las citas por parte de usuario técnico asignado. 	28/03/2019 A 05/04/2019
10	<ul style="list-style-type: none"> • Corrección de almacenamiento de documentos en Dropbox motivado a fallo en el formato de envío de los documentos. • Consulta de estado de reporte por parte de cliente con su número de reporte. • Envío de correo automático dado un cambio en el estado del reporte. 	8/04/2019 A 17/04/2019
11	<ul style="list-style-type: none"> • Manejo de errores y excepciones. • Ajustes y mejoras visuales en la aplicación. • Ejecución de pruebas de funcionalidad. 	22/04/2019 A 03/05/2019

Tabla 4: Lista de Objetivos y tareas realizadas

4.2.2 Lista de tareas de la iteración

Sprint 1

Se efectuó la instalación de las tecnologías elegidas para el desarrollo de la aplicación como VScode como editor para el desarrollo, luego se realizó la instalación de NodeJS, mediante el manejador de paquetes de NodeJS (npm) se logró instalar SailsJS y luego se procedió a la creación del proyecto. Se integró AngularJS a la base del proyecto para desarrollo del front-end, se integró Bootstrap y LESS. Se diseñó el modelo de datos que soportaría el correcto funcionamiento y almacenamiento de los datos en la aplicación. Se instaló la base de datos mongodb. Se creó la instancia de base de datos, sus documentos y referencias.

Sprint 2

Se maquetaron todas las interfaces de la aplicación para seguir el flujo de funcionalidades y abarcar todo lo propuesto en el alcance. Se logró definir un logo y paleta de colores. Se realizaron las pruebas de funcionalidad necesarias para comprobar el funcionamiento del sistema con toda la integración de las nuevas dependencias. Se hizo un maquetado del menú lateral que posee la aplicación y se hicieron las llamadas al servidor correspondientes para probar la integración correcta del API. Configuración del servidor, dependencias y conexión a base de datos mediante el ORM.

Sprint 3

En esta iteración inició la creación del módulo de autenticación con parte de sus funcionalidades asociadas. Se realizó la navegación entre interfaces siguiendo el flujo de trabajo propuesto y se hizo el pase de datos como parámetros de manera satisfactoria.

Sprint 4

Se implementó el CRUD de usuarios, se realizaron los formularios correspondientes a cada creación de usuario con sus validaciones respectivas de los datos, se realizó la llamada al servicio de creación de usuario y se envía un correo de confirmación para indicarle al usuario que ya posee una cuenta para entrar al sistema.

Sprint 5

En este Sprint gracias a la implementación del CRUD de usuarios en el sprint anterior se mostró la lista de usuarios creados en el sistema por tipo de usuario, podemos ver el perfil de cada usuario y editarlo estos datos de ser necesario según el tipo de usuario que realice esta tarea

Sprint 6

Se integró un calendario y se realizaron pruebas con las funcionalidades que permitía esta librería para saber su alcance. Se terminó el módulo de auditoría que permite saber que usuario está realizando una acción en el sistema y se ajustaron las permisos para cada usuario en el sistema, de manera que cada usuario realice las funciones que les corresponden.

Sprint 7

Se creó todo el módulo de reportes y se dio pie al módulo de creación de citas, se crearon los formularios correspondientes y se validaron los campos de entrada de datos. Se hace llamada al servicio de creación de reportes.

Sprint 8

En este sprint se terminó el módulo de citas. Se crea una cita dado un reporte de servicio técnico y se implementa el cálculo automatizado de fecha disponible tomando en cuenta, técnico disponible, cliente con reporte y una fecha próxima. Se muestra en la lista de citas del técnico la cita creada y se muestra en el calendario resaltando que esa fecha ya está reservada. Como estas tareas involucran muchos módulos de la aplicación web se ejecutaron pruebas de funcionalidad en todo el sistema.

Sprint 9

Se integró Dropbox para el almacenamiento de archivos en la nube dentro de la cita por parte del técnico. Cuando un técnico necesite enviar archivos pertinentes al caso y que dejen evidencia de su trabajo. Estos archivos son enviados mediante un servicio web, que luego al tener el archivo en el servidor este lo pasa a Dropbox y lo almacenando el archivo.

Sprint 10

En esta iteración se corrigió el servicio de envío de archivos, se agregó la funcionalidad de eliminar archivo de cita, además de restringir que tipo de archivos pueden enviarse a solo archivos con formato pdf, doc, docx e imágenes con los formatos jpg, jpeg y png. Se consulta el estado de reporte por parte del cliente con su número de reporte, además del envío de correo electrónico automático dado un cambio en el estado del reporte.

Sprint 11

Finalmente, se realizó el manejo de errores y excepciones respectivo de la aplicación para prevenir o minimizar las fallas por errores comunes. Se dio otro retoque estético a la aplicación para

mejorar la usabilidad de la misma y se realizaron las últimas pruebas de funcionalidad correspondientes para finalizar todo el proceso de envío de solicitud.

4.3 Requerimientos del sistema

Una vez realizado el levantamiento de requerimientos definidos por el cliente, haciendo uso del rol de Product Owner, los mismos se distribuyen de la siguiente manera:

4.3.1 Requerimientos funcionales

- Autenticación con credenciales de la base de datos central.
- Crear, listar, visualización de perfil y editar usuarios del sistema.
- Registro de reporte de un cliente.
- Creación de cita técnica programada dado un reporte.
- Comentar cita técnica.
- Visualización de Calendario de un técnico marcando sus días de cita técnica.
- Envío de correo automático por creación de usuario, creación de reporte, creación de cita y cambio de estados de cita
- Almacenamiento de archivos de la cita en la nube.
- Calculo automático de técnico disponible, en un día próximo, para la creación de una cita.
- Envío de notificaciones en tiempo real cuando se asigne una cita a un técnico.

4.3.2 Requerimientos no funcionales

- Validar la entrada de datos de todas las interfaces para prevenir ataques a la integridad de la base de datos central.
- Mensajes de retroalimentación sobre las acciones del usuario.
- Manejo de errores y excepciones para minimizar fallas.

4.4 Perfiles de usuarios

- **Cliente:** Es el usuario que interactúa de manera externa con el sistema. El sistema está orientado a mejorar la gestión del reporte y ayudar a que la compañía proveedora de servicio internet brinde un mejor servicio a este usuario, el sistema le permitirá mediante un código, número de contrato y número de reporte visualizar el estado de su reporte, además que el sistema le estará notificando vía correo electrónico cuando se le cree un reporte, el estado del reporte y el estado de la cita, mientras el reporte esté siendo atendido por el técnico asignado.

- **Empleado:** Es el usuario encargado de crear los reportes, crear citas, desde su rol puede visualizar a los otros usuarios del sistema y editar alguno de sus datos de ser necesario. Sus funciones son:
 - Crear reportes.
 - Crear citas a un reporte.
 - Crear usuarios: cliente, empleado y técnico.
 - Visualización de perfil propio.
 - Visualización y Modificación de información del cliente y técnico.
 - Visualización del calendario de trabajo de los técnicos.
 - Visualización de lista y detalle del reporte.
 - Visualización de lista de citas de un reporte.

- **Técnico:** Este usuario es quien debe dar solución a los reportes generados por el usuario empleado, su rol es sumamente importante ya que de él depende que el reporte se cierre en un tiempo adecuado y con resultados positivos. Sus funcionalidades implican
 - Visualización de citas asignadas.
 - Comentar citas.
 - Agregar documentos o archivos a una cita.
 - Cambiar de estado la cita.
 - Visualización de perfil propio.

4.5 Descripción del flujo asociado a la solución

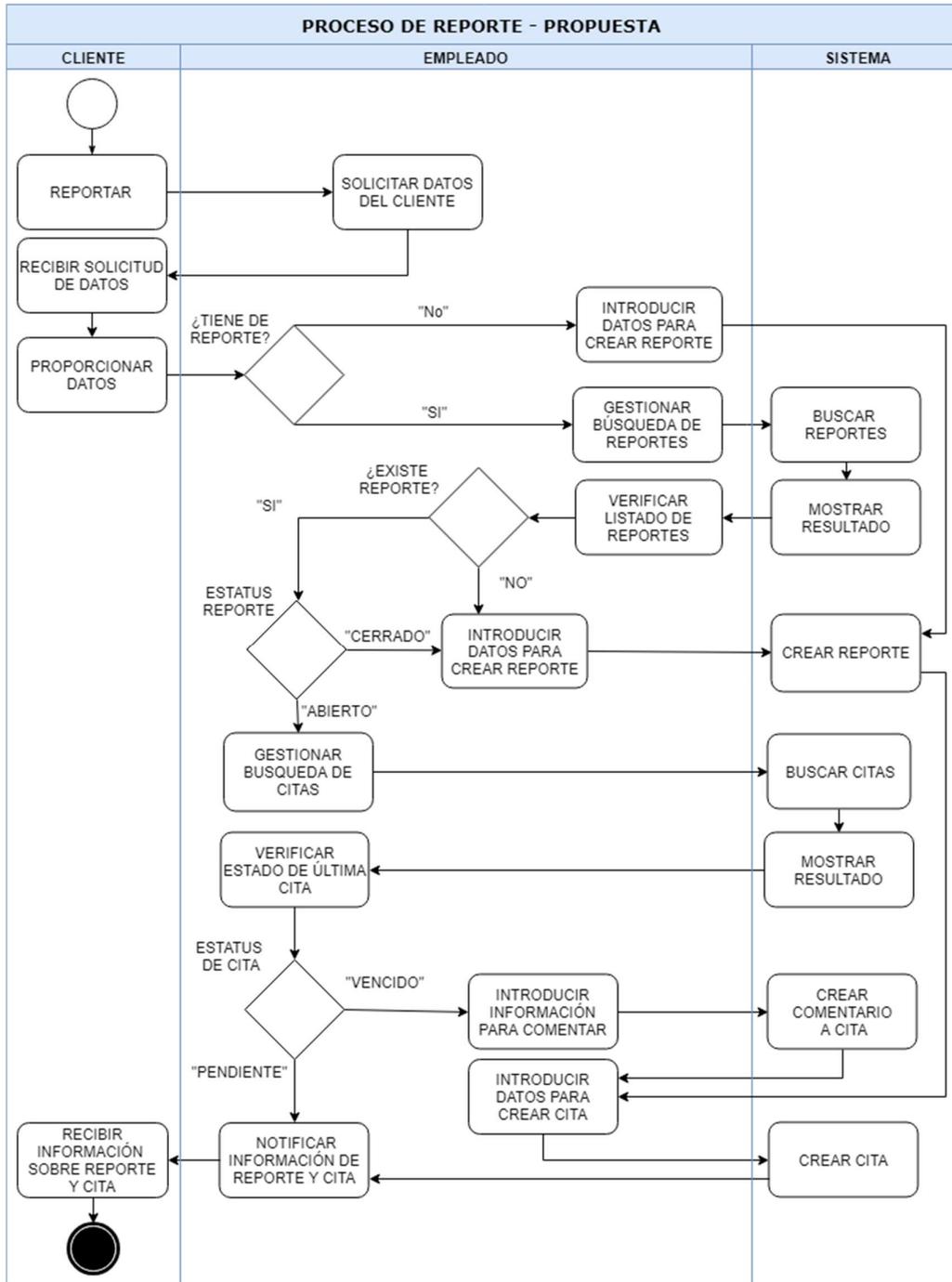


Ilustración 12: Flujograma del proceso propuesto de reporte

En la ilustración 12, se detalla en el proceso de reporte por parte de un cliente: Comienza cuando un cliente reporta un inconveniente con su servicio contactando a la compañía proveedora de servicio internet, un empleado de la compañía es quien atiende al cliente y solicita información como el número de contrato del cliente y número de reporte en caso de tenerlo. Si el cliente posee reporte,

el empleado hace una búsqueda en el sistema con el número de contrato y el número de reporte para detallar la información correspondiente, el empleado verifica que, si el reporte existe, luego se comprueba si el reporte está en estatus abierto, de ser así se gestiona la búsqueda de las citas que posee el reporte, se verifica el estatus de la última cita, si está en estatus pendiente el empleado le notifica al cliente la información de reporte y cita.

El empleado crea un nuevo reporte si: el cliente no posee reporte; el reporte no existe; el reporte está en estatus cerrado o la última cita del reporte está se encuentra en estatus vencido o cerrado. Posteriormente crea una cita para ese reporte, se le informa al cliente su número de reporte generado y la fecha en la que se fijó la cita con el técnico de la compañía para ir a resolver su problema.

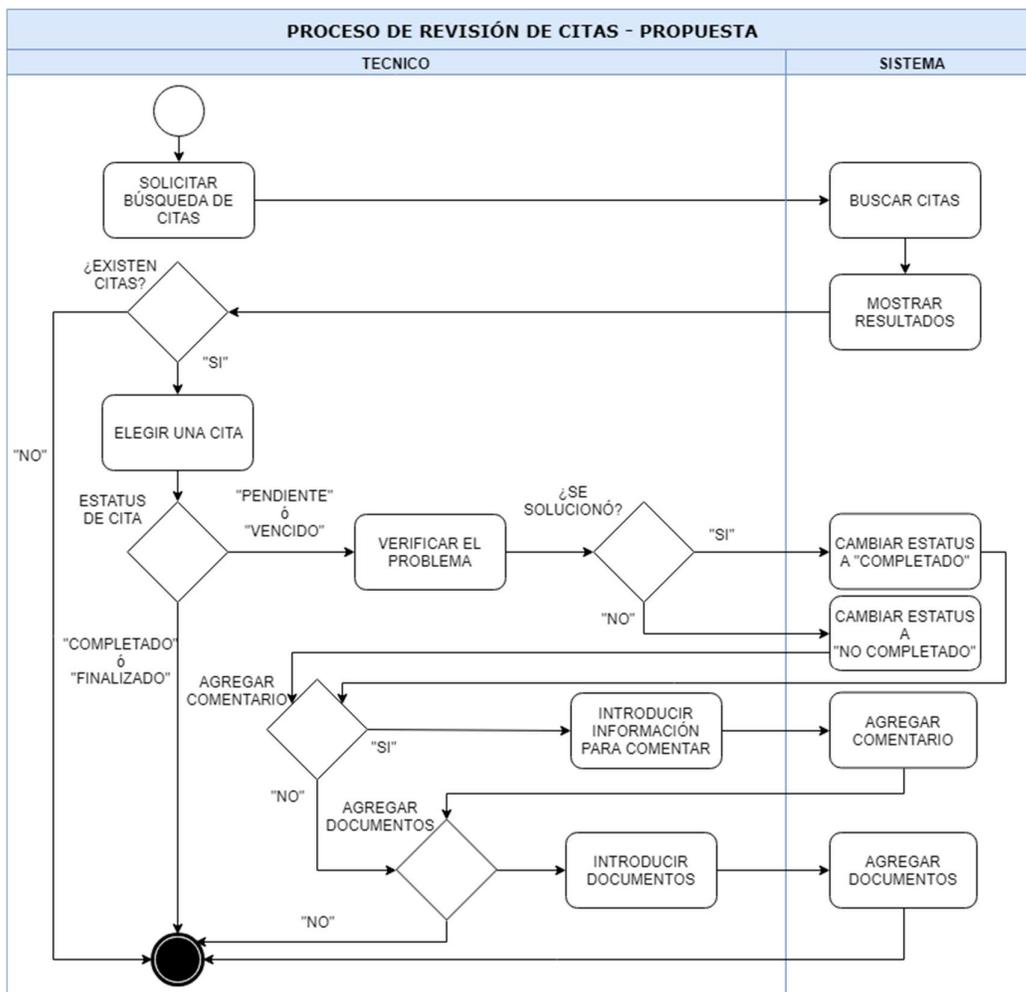


Ilustración 13: Flujograma del proceso propuesto de revisión de citas

En la ilustración 13, se especifica el proceso de revisión de citas por parte de un técnico perteneciente a la compañía proveedora de servicio internet: Comienza con el técnico realizando

una búsqueda en el sistema de sus citas, si no existen citas se termina el proceso, si existen citas se elige una, posteriormente se verifica el estatus de la cita, si está en estatus completado o finalizado el técnico termina el proceso.

Cuando el estatus de la cita es pendiente o vencido el técnico verifica si el problema está solucionado, en caso de estar solucionado el problema, el técnico mediante el sistema pasa la cita a estatus completado, si no se solucionó el problema se pasa a estatus no completado. Así mismo el técnico luego de cambiar el estatus de la cita decide si agregar un comentario, de esta misma forma si tiene documentos que sumen información importante a la cita los podrá adjuntar, terminando el proceso de revisión de citas por parte del técnico.

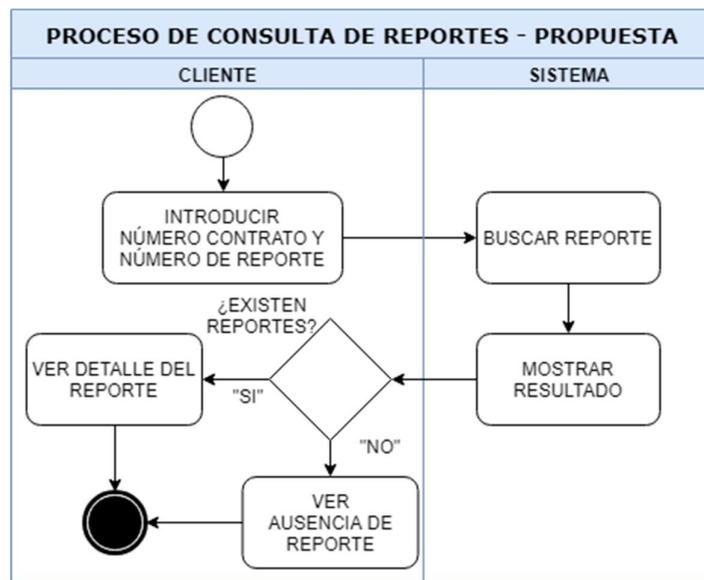


Ilustración 14: Flujograma del proceso propuesto de consulta de reportes.

Por último, en la ilustración 14, se detalla el proceso de revisión de citas por parte de un cliente: Inicia con el cliente navegando a la página web de la compañía donde en la sección de reportes podrá introducir su número de contrato que lo identifica como cliente de la compañía y el número de reporte. El sistema hace una búsqueda con los datos proporcionados y muestra la información sobre el reporte terminado el proceso, si el reporte no existe se le muestra un mensaje en la pantalla al cliente indicando lo que ocurrió.

4.6 Análisis del modelo de datos

4.6.1 Listado de tablas de la aplicación web

- **Usuario:** Almacena los valores de los usuarios, que serán los actores principales del sistema.
- **Rol:** Almacena las acciones y permisos para un usuario.
- **Teléfono:** Almacena los números telefónicos de los usuario y clientes.
- **Técnico:** Almacena los datos de los técnicos.
- **Empleado:** Almacena los datos de los empleados.
- **Cliente:** Almacena los datos de los clientes.
- **Comentario:** Almacena los comentarios realizados en una cita por un usuario.
- **Notificación:** Almacena las notificaciones de eventos en el sistema para los usuarios involucrados.
- **Reporte:** Almacena los reportes generados para un cliente.
- **Cita:** Almacena las citas creadas dentro de un reporte.
- **Estado:** Almacena información los nombres de los estados del país.

4.6.2 Modelo de datos.

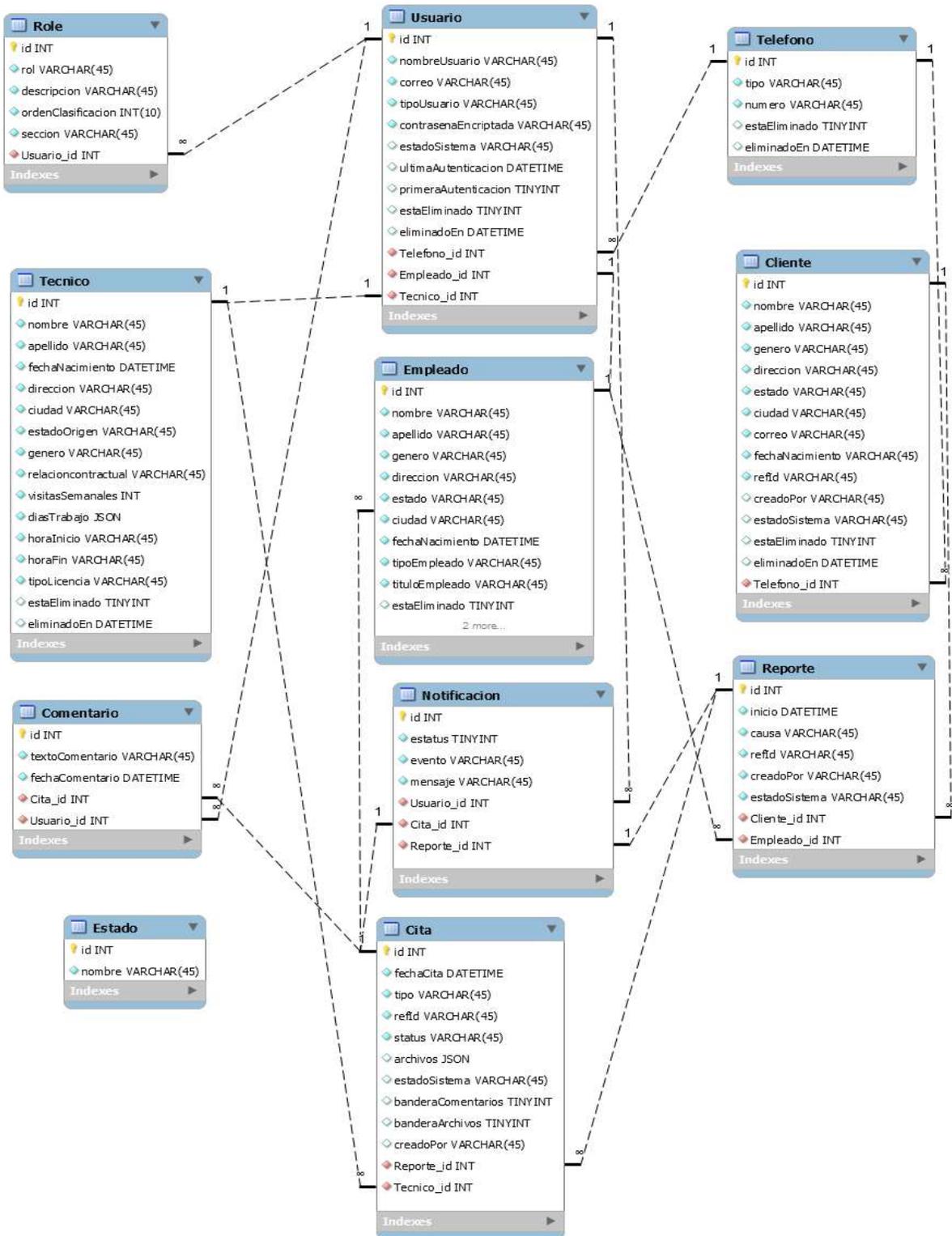


Ilustración 15: Modelo de datos

4.7 Descripción de los módulos del sistema e interfaces

4.7.1 Descripción de los módulos del sistema.

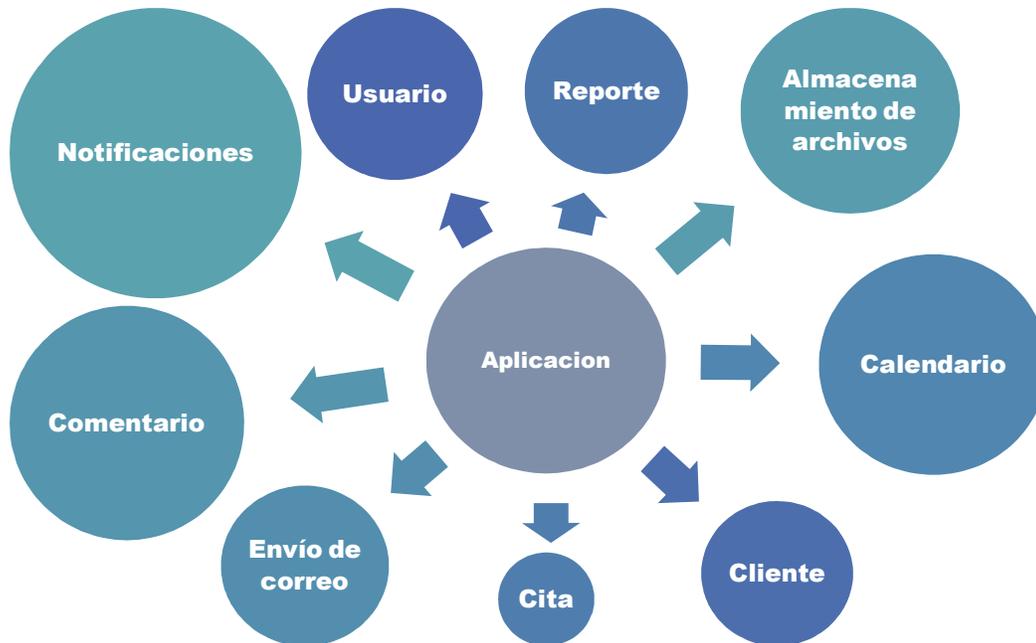


Ilustración 16: Modelo de datos

- **Autenticación:** en este módulo, los usuarios realizan el proceso de autenticación con sus credenciales para entrar en el sistema, además de mantener la sesión por un periodo de tiempo.
- **Usuario:** gracias a este módulo es posible crear, almacenar, listar, visualizar y editar el perfil del usuario.
- **Cliente:** por medio de este módulo es posible registrar a los clientes en el sistema con un identificador para así poder visualizar sus reportes y sus citas.
- **Reporte:** en este módulo podemos crear reportes, almacenarlos, listarlos, visualizarlos, además de cambiar el estado del reporte.
- **Citas:** permite creación de citas, almacenarlos, listarlos, visualizarlos, ver el detalle de esa cita y cambiar su estado.
- **Calendario:** por este módulo se almacenan las fechas de citas asignadas a un cliente.
- **Envío de correos:** por medio de este módulo es posible el envío de correos de manera automática.
- **Comentarios:** permite el agregar comentario a la cita, además de eliminarlo.

- **Almacenamiento de archivos:** por medio de este módulo es posible gestionar archivos relacionados a la cita.
- **Notificaciones:** permite el envío y recepción de notificaciones dentro del sistema.

4.7.2 Logo diseñado y Paleta de colores

A continuación, se muestra el logo sé que diseño para esta aplicación web



Ilustración 17: Logo diseñado para el sistema

Ahora se muestra la paleta de colores a utilizar dentro de la aplicación, para dar una sensación tecnológica y confortable.

#009AFF	COLOR PRINCIPAL
#FF6500	COLOR SECUNDARIO
#FFFFFF	COLOR TERCIARIO
#777777	COMPLEMENTARIO 1
#006BB1	COMPLEMENTARIO 2

4.7.3 Descripción de las interfaces del sistema.

- **Inicio de sesión:** Esta interfaz le permite al usuario ingresar en la aplicación web con su nombre de usuario y clave que les corresponde.



Ilustración 18: Inicio de sesión de la aplicación.

- **Recuperar contraseña:** Esta interfaz le permite al usuario recuperar su contraseña ingresando su correo electrónico.



Ilustración 19: Recuperar contraseña.

- **Consulta de reporte para el cliente:** Esta interfaz le permite al cliente consultar el estatus de su reporte con su número de cliente y número de reporte asignado.



Ilustración 20: Consulta de reporte para el cliente.

- **Detalle consulta de reporte para el cliente:** Esta interfaz le permite al cliente detallar información de su reporte, el estatus y las citas que han sido programadas.



Ilustración 21: Detalle consulta de reporte para el cliente.

- **Consulta de reporte para el empleado:** Esta interfaz le permite al empleado consultar información de un cliente con un número de identificación correspondiente.



Ilustración 22: Consulta de cliente para el empleado.

- **Detalle de cliente:** Esta interfaz le permite al empleado detallar información de un cliente.

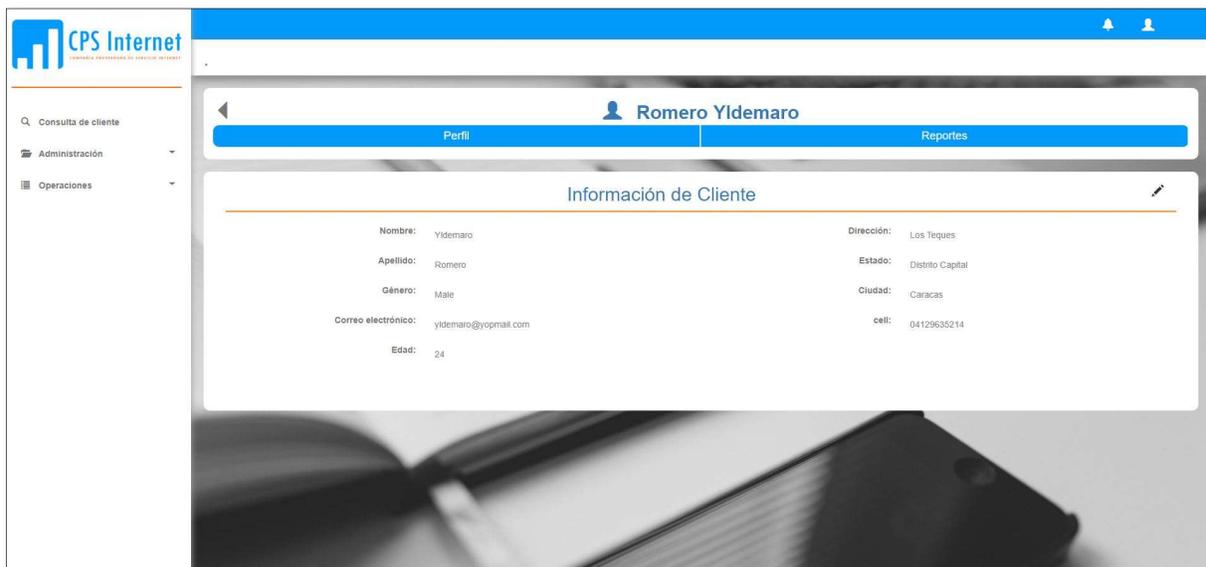


Ilustración 23: Detalle de cliente.

- **Lista de reportes:** Esta interfaz le permite al empleado ver la lista de reportes de un cliente, con su prioridad y estatus, además le da la opción de crear un nuevo reporte.

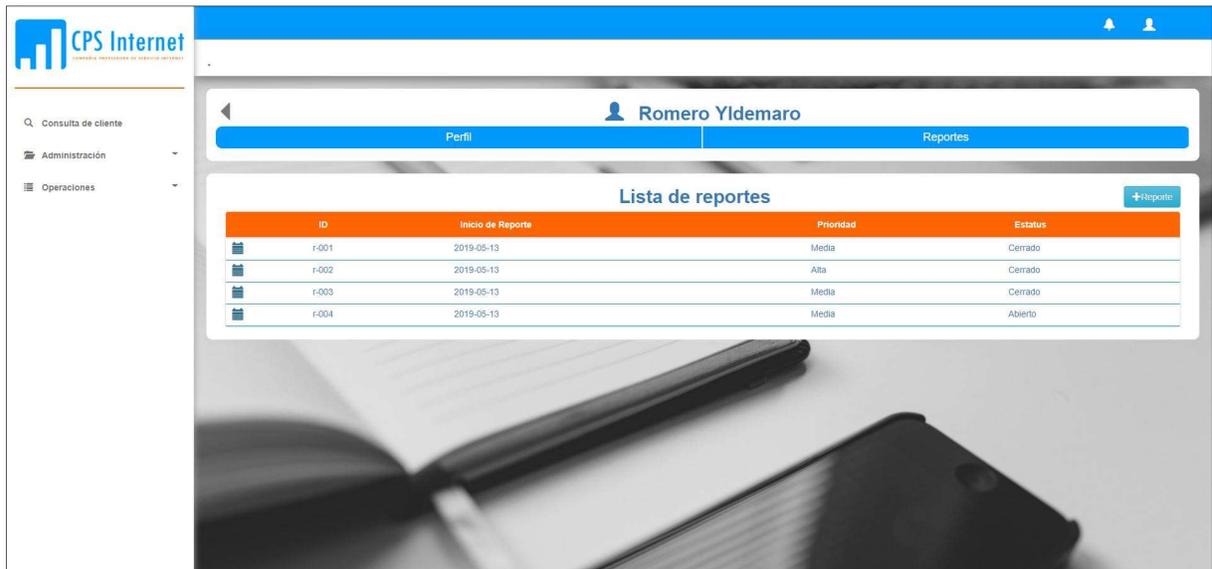


Ilustración 24: Lista de reportes.

- **Crear nuevo reporte:** Esta interfaz le permite al empleado crear un nuevo reporte para un cliente, indicando la causa y la prioridad. **Detalle de cliente:** Esta interfaz le permite al empleado detallar información de un cliente.



Ilustración 25: Crear nuevo reporte.

- **Calendario de citas:** Esta interfaz le permite al empleado detallar información del reporte, fecha de las citas asignadas, estatus de las citas asignadas, agregar nueva cita en una fecha con un nuevo técnico de manera automática y elegir el tipo de cita a asignar.

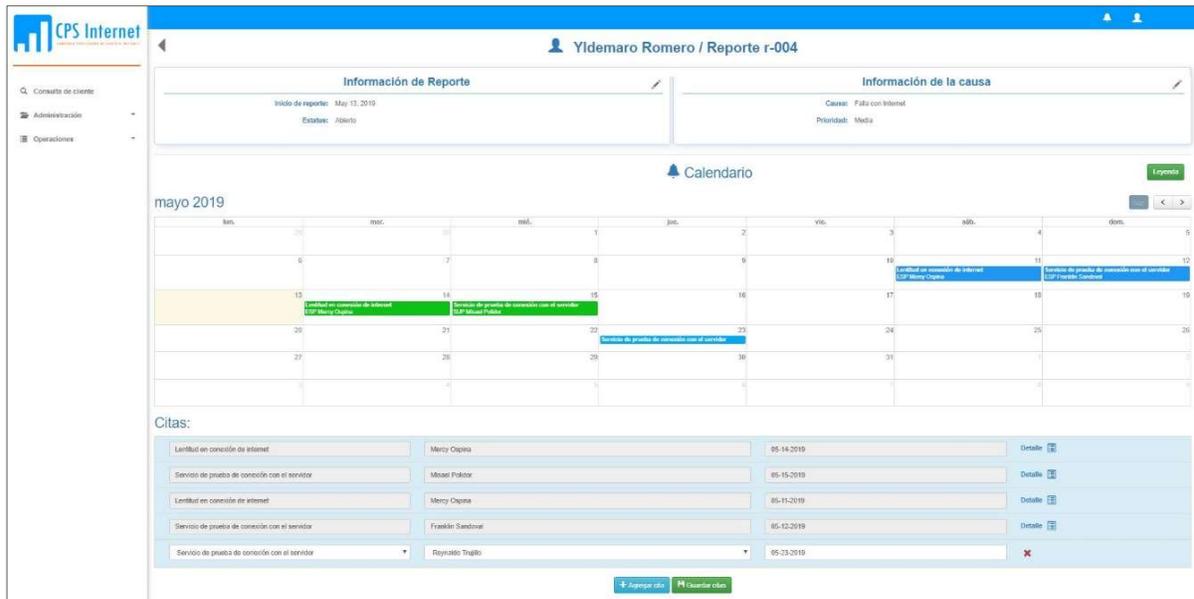


Ilustración 26: Calendario de citas.

- **Detalle de cita:** Esta interfaz le permite al empleado detallar la cita, su estatus, cliente y técnico involucrados; además permite gestionar documentos y agregar comentario.

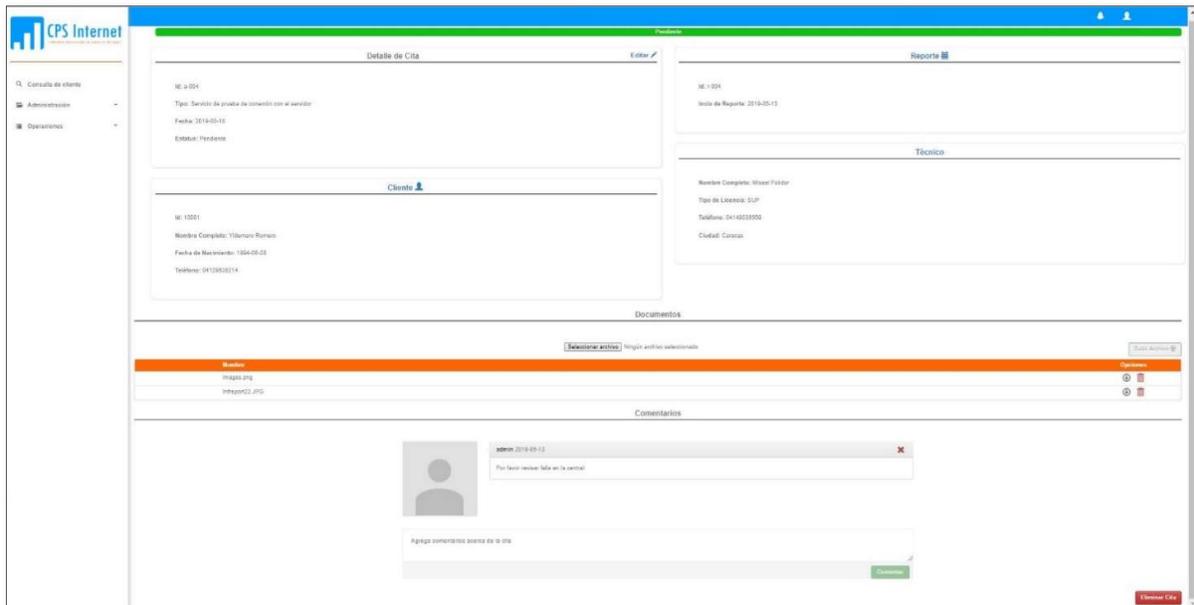


Ilustración 27: Detalle de cita.

- **Lista de clientes:** Esta interfaz le permite al empleado ver lista de clientes registrados en el sistema y muestra la opción para crear un nuevo cliente.

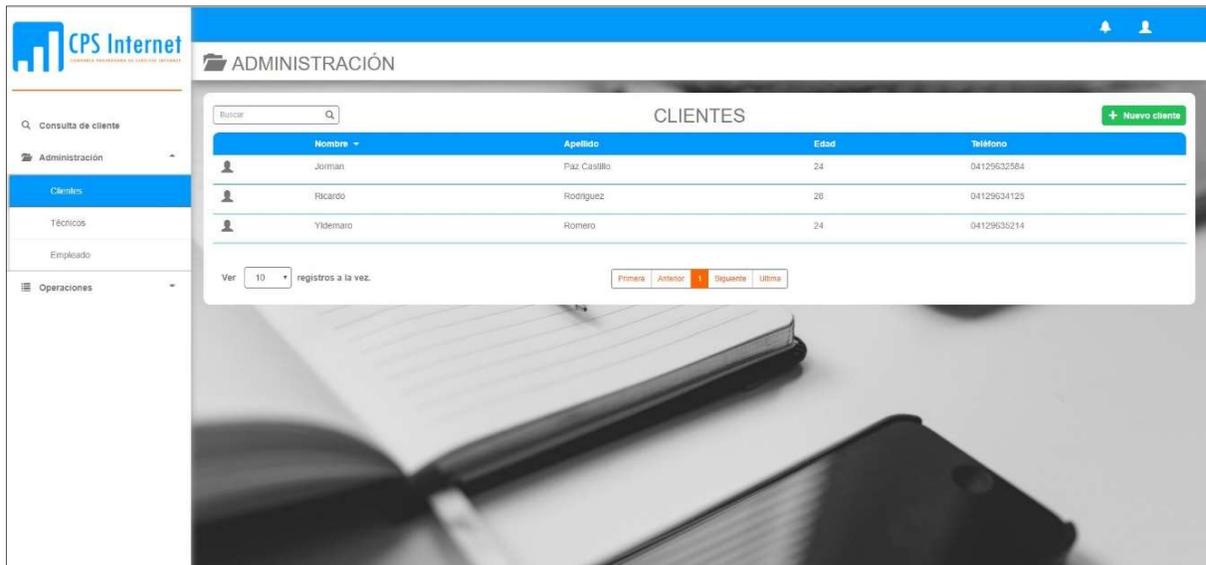


Ilustración 28: Lista de clientes.

- **Formulario de nuevo cliente:** Esta interfaz le permite al empleado registrar un cliente en la aplicación.



Ilustración 29: Formulario de nuevo cliente.

- **Lista de técnicos:** Esta interfaz le permite al empleado ver lista de técnicos registrados en el sistema y ofrece la opción de crear un nuevo técnico.

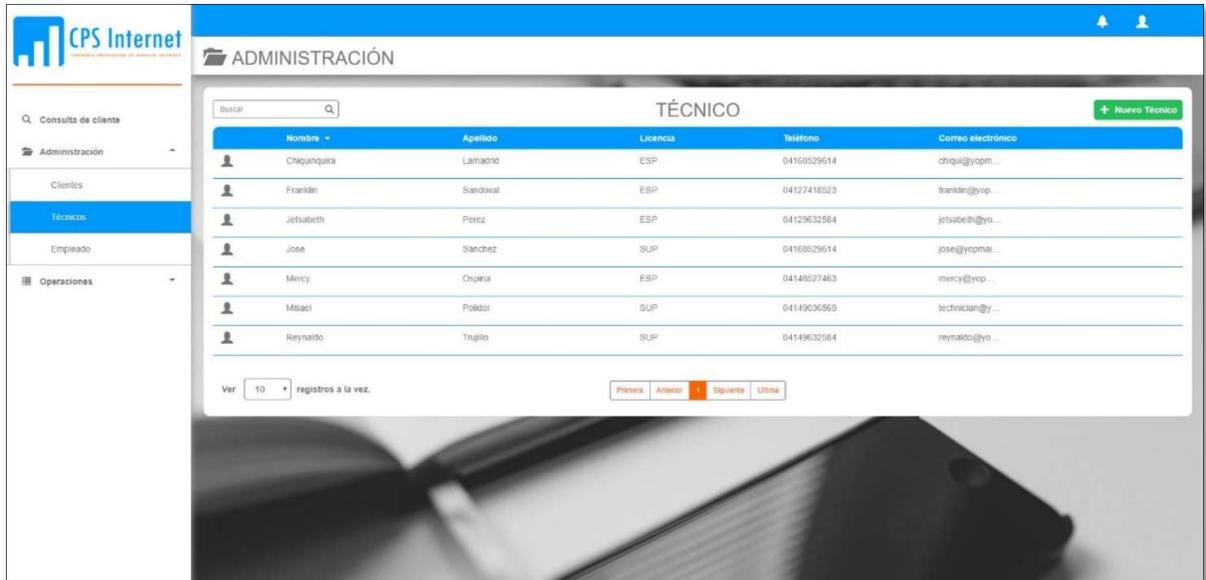


Ilustración 30: Lista de técnicos.

- **Formulario de nuevo técnico paso 1:** Esta interfaz le permite al empleado iniciar el registro de un técnico, llenando la información personal correspondiente.

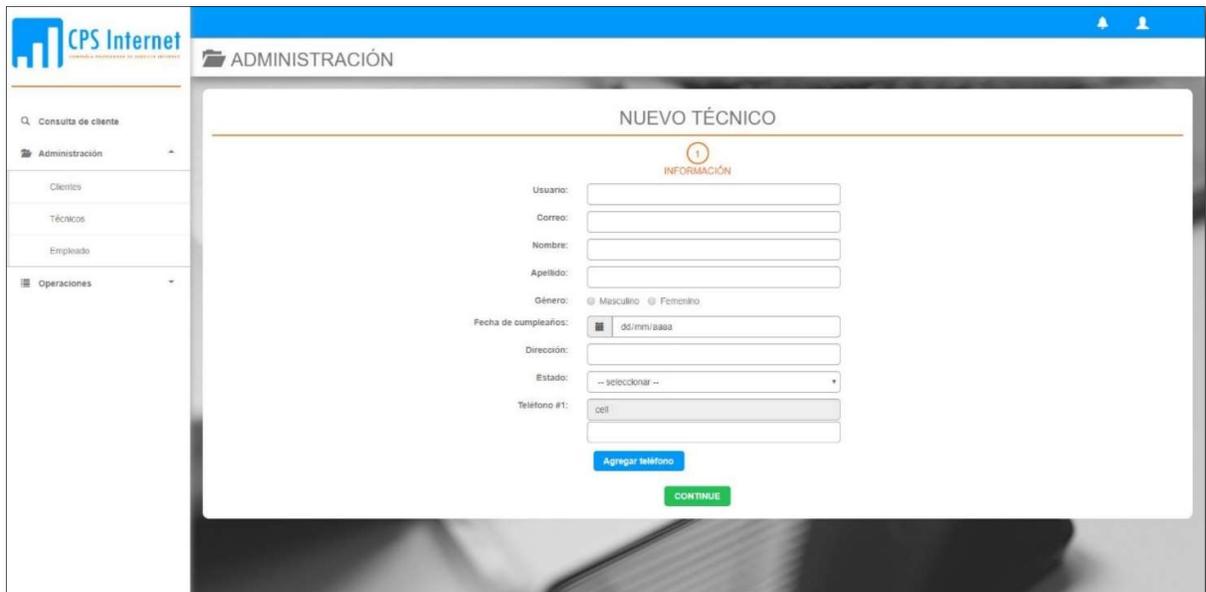


Ilustración 31: Formulario de nuevo técnico paso 1.

- **Formulario de nuevo técnico paso 2:** Esta interfaz le permite al empleado finalizar el registro de un técnico, llenando los datos profesionales correspondientes.

Ilustración 32: Formulario de nuevo técnico paso 2.

- **Lista de empleados:** Esta interfaz le permite al empleado ver lista de empleados registrados en el sistema y ofrece la opción de crear un nuevo empleado.

Nombre	Apellido	Posición	Correo electrónico
Kevin	Main	employee	employee@yopmail.com
Misael	Poldor	administrator	misaelemp@yopmail.com

Ilustración 33: Lista de empleados.

- **Formulario de nuevo empleado:** Esta interfaz le permite al empleado registrar un empleado en la aplicación.

The screenshot shows the 'NUEVO EMPLEADO' (New Employee) form. The form is titled 'NUEVO EMPLEADO' and has a sub-header 'INFORMACIÓN'. The fields are as follows:

- Usuario: [input field]
- Correo electrónico: [input field]
- Nombre: [input field]
- Apellido: [input field]
- Género: Masculino Femenino
- Fecha de cumpleaños: [calendar icon] dd/mm/aaaa
- Tipo de empleo: [dropdown menu: --seleccionar--]
- Cargo: [dropdown menu: --seleccionar--]
- Dirección: [input field]
- Estado: [dropdown menu: --seleccionar--]
- Teléfono #1: [input field with 'cell' placeholder]

Buttons: 'Agregar teléfono' (blue), 'CREATE' (green).

Ilustración 34: Formulario de nuevo empleado.

- **Detalle de técnico:** Esta interfaz le permite al empleado detallar información de un técnico.

The screenshot shows the 'Detalle de técnico' (Technician Detail) page for 'Trujillo Reynaldo'. The page has a header 'Trujillo Reynaldo' and a sub-header 'Perfil'. The main content is divided into two sections:

- Información de Técnico:**
 - Nombre de Usuario: morales
 - Correo: reynaldo@yopmail.com
 - Contraseña: [button: Resetear contraseña]
 - Nombre: Reynaldo
 - Apellido: Trujillo
 - Género: Male
 - Dirección: Charallave
 - Estado: Distrito Capital
 - Cell: 04149632084
 - Edad: 25
- Profesión:**
 - Tipo de Licencia: SUP
 - Citas semanales: 7
 - Días de trabajo: Lunes, Jueves, Sabado, Domingo
 - Horario de trabajo: 12:00 a 17:00

Ilustración 35: Detalle de técnico.

- **Detalle de empleado:** Permite al empleado detallar información de otro empleado.

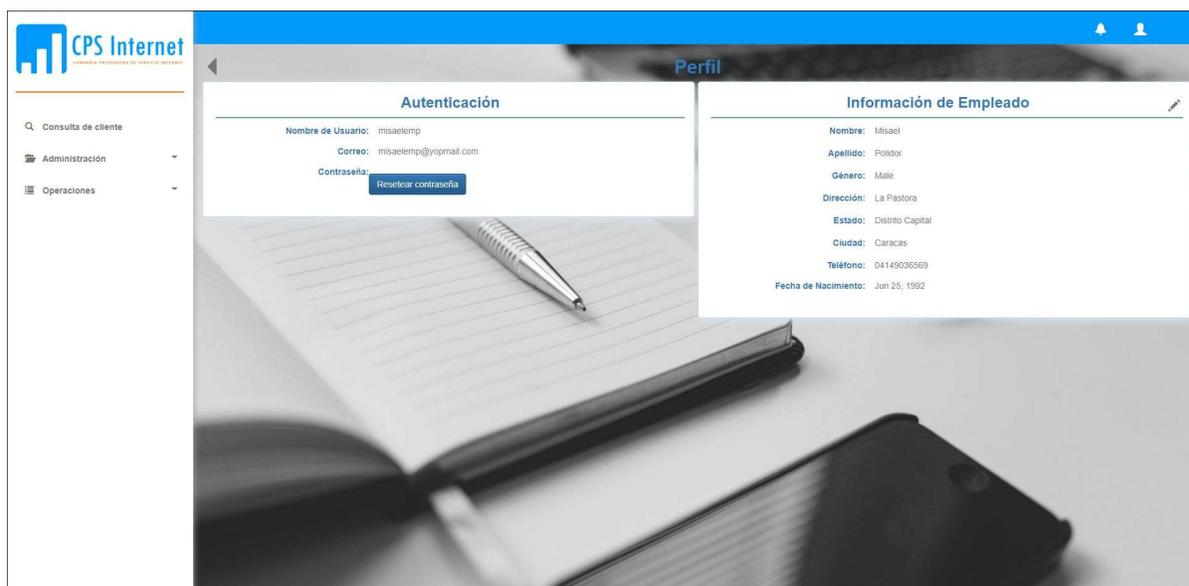


Ilustración 36: Detalle de empleado.

- **Gerencia de calidad:** Esta interfaz le permite al empleado ver lista de citas que han sido completadas y las que no, para realizar un seguimiento del motivo de su estatus. Además, le permite a un empleado cambiar el estatus de la cita.

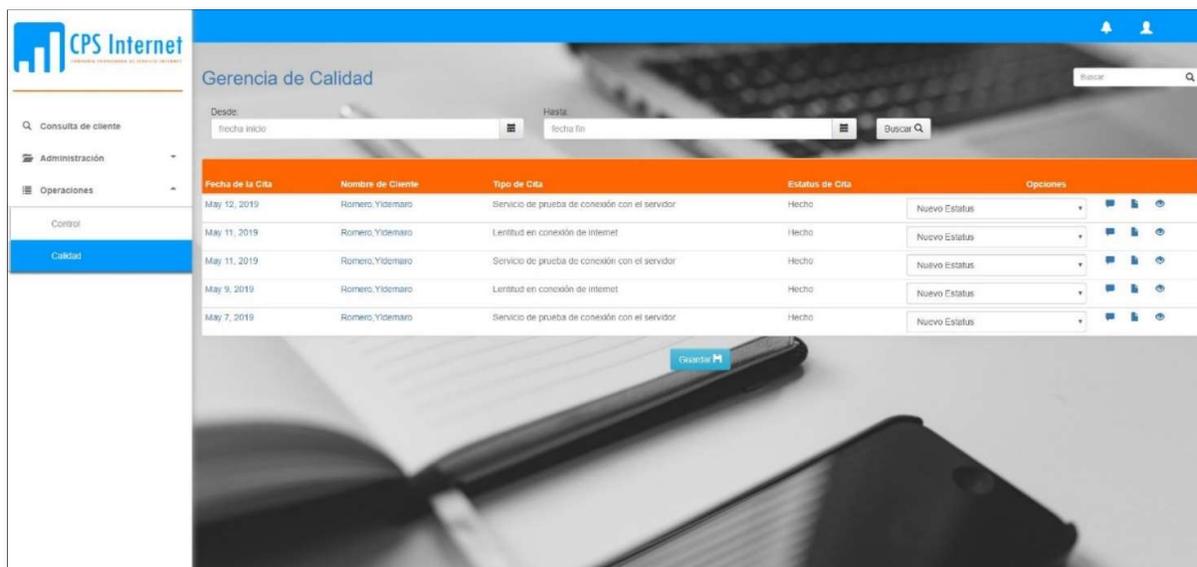


Ilustración 37: Gerencia calidad.

- **Gerencia de técnica:** Esta interfaz le permite al técnico ver lista de citas que le han sido asignadas, cambiar su estatus, ver detalle, realizar comentarios y ver archivos adjuntos relacionados.

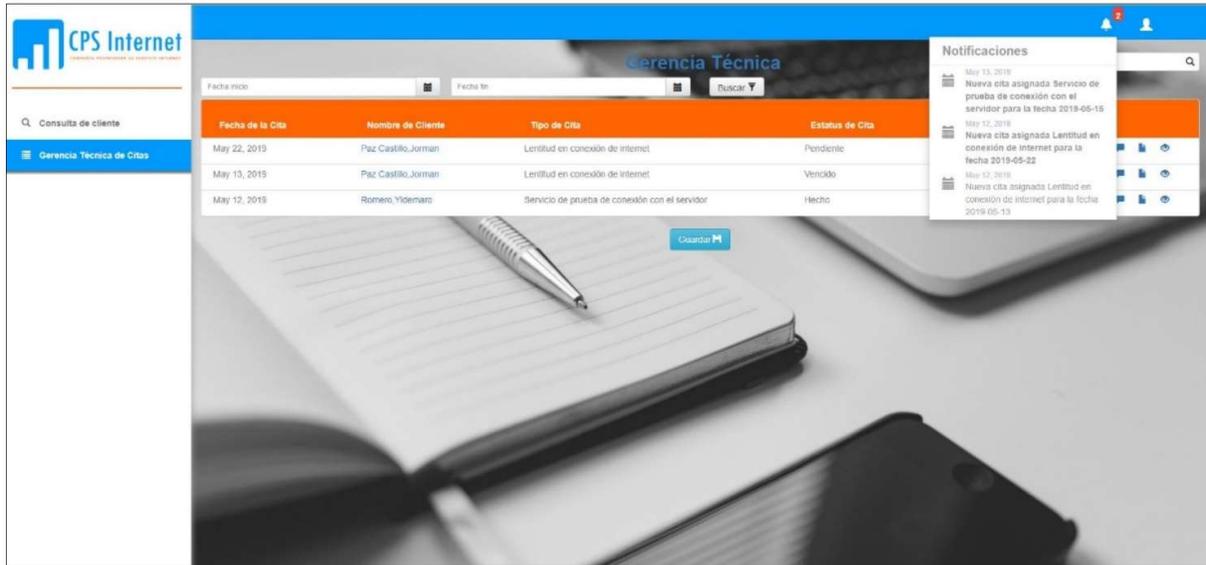


Ilustración 38: Gerencia técnica.

- **Gerencia de técnica:** Esta interfaz le permite al técnico ver sus datos de perfil.

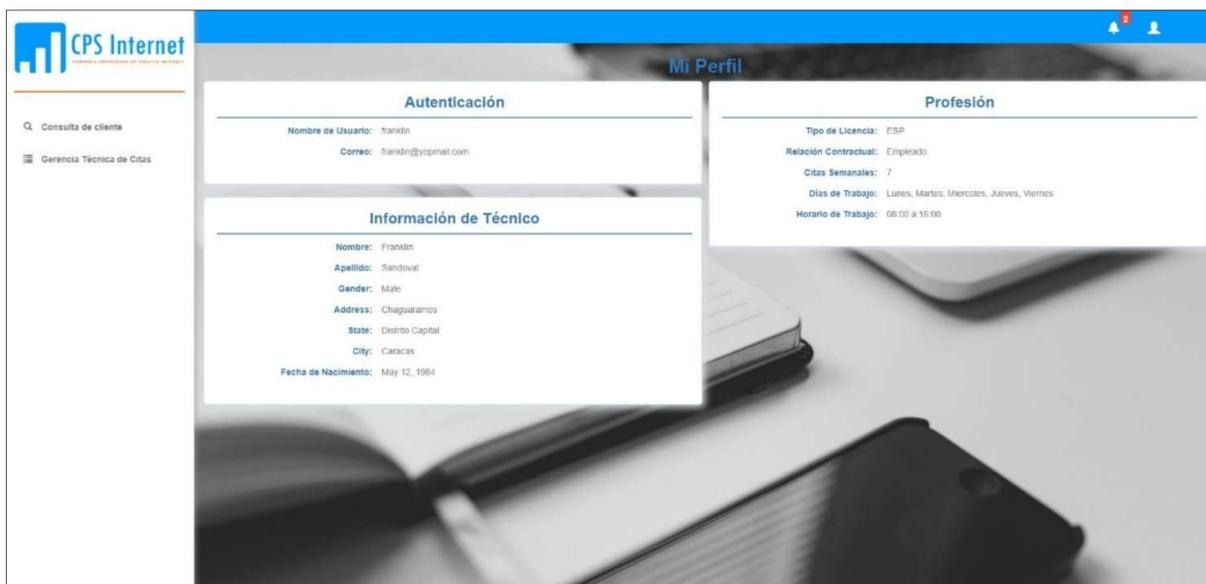


Ilustración 39: Perfil del técnico.

4.8 Fase de pruebas

Se realizó un conjunto de pruebas para comprobar el correcto funcionamiento del sistema, además de la aceptación o receptividad de los usuarios con la aplicación. Para ello, se llevaron a cabo dos tipos de pruebas. Seguidamente, una explicación de cada tipo.

4.8.1 Pruebas funcionales

Una de las etapas más importantes en la metodología SCRUM es la etapa de pruebas funcionales, donde cada módulo del sistema y sus funcionalidades son puesta a prueba.

MÓDULO	FUNCIONALIDADES PROBADAS
Autenticación	<ul style="list-style-type: none">• Validación de datos de entrada.• Envío de datos al servidor.• Persistencia de sesiones.
Usuario	<ul style="list-style-type: none">• Creación de Usuario.• Lista de usuarios.• Perfil de usuario.• Edición de datos de usuario.
Cliente	<ul style="list-style-type: none">• Registro de cliente con identificador.• Consultar reportes y citas por parte de cliente.
Reporte	<ul style="list-style-type: none">• Creación de reportes.• Listado de reportes.• Visualización de reporte.• Cambio de estado en reporte.
Citas	<ul style="list-style-type: none">• Creación de citas.• Listado de citas de cliente.• Lista de citas pertenecientes a un técnico.• Visualización de cita.• Cambio de estado en la cita por parte de un técnico.
Calendario	<ul style="list-style-type: none">• Calendario mostrando fechas de citas asignadas a un cliente.• Ir al detalle de una cita desde calendario.
Envío de correos	<ul style="list-style-type: none">• Envío de correo automático por creación de usuario.• Envío de correo automático a un cliente notificando creación del reporte, creación de citas y cambios en el estado del reporte y de la cita.
Comentarios	<ul style="list-style-type: none">• Agregar comentarios a cita.• Eliminar comentario por parte empleado.
Almacenamiento de archivos	<ul style="list-style-type: none">• Subir archivo relacionado con cita.• Descargar archivo relacionado con la cita.
Notificaciones	<ul style="list-style-type: none">• Envío de notificaciones en tiempo real aun técnico cuando le sea asignado una cita a un técnico.• Recepción de notificación en tiempo real aun técnico cuando se le sea asignado una cita.

Tabla 5: Pruebas funcionales por módulo

Las pruebas presentadas en la tabla 5 se ejecutaron de manera cíclica hasta lograr una iteración con la aprobación de todos los módulos, dicho proceso es expuesto en la tabla 6 a continuación.

Usuario	Iteración #1	Iteración #2	Iteración #3	Iteración #4	Iteración #5
Empleado #1	5 funcionalidades aprobadas. 3 funcionalidades no aprobadas.	6 funcionalidades aprobadas. 2 funcionalidades no aprobadas.	5 funcionalidades aprobadas. 1 funcionalidad no aprobada.	2 funcionalidades aprobadas. 1 funcionalidades aprobadas.	Todas las funcionalidades fueron aprobadas
Empleado #2	5 funcionalidades aprobadas. 3 funcionalidades no aprobadas.	6 funcionalidades aprobadas. 2 funcionalidades no aprobadas.	5 funcionalidades aprobadas. 1 funcionalidad no aprobada.	2 funcionalidades aprobadas. 1 funcionalidad no aprobada.	Todas las funcionalidades fueron aprobadas
Técnico #1	3 funcionalidades aprobadas.	2 funcionalidades aprobadas. 1 funcionalidad no aprobada.	2 funcionalidades aprobadas.	2 funcionalidades aprobadas. 1 funcionalidad no aprobada.	Todas las funcionalidades fueron aprobadas
Técnico #2	3 funcionalidades aprobadas.	2 funcionalidades aprobadas. 1 funcionalidad no aprobada.	2 funcionalidades aprobadas.	2 funcionalidades aprobadas. 1 funcionalidad no aprobada.	Todas las funcionalidades fueron aprobadas
Cliente	Sin funcionalidades por evaluar	1 funcionalidad aprobadas.	2 funcionalidades aprobadas.	Sin funcionalidades nuevas por evaluar, las funcionalidades anteriores siguen funcionando	Todas las funcionalidades fueron aprobadas

Tabla 6: Matriz de ciclos de prueba

ITERACIÓN #1

Funcionalidades a ser evaluadas por el **empleado** en la iteración #1 son:

- Validación de datos de entrada.
- Envío de datos al servidor.
- Persistencia de sesiones.
- Creación de Usuario.
- Lista de usuarios.
- Perfil de usuario.
- Edición de datos de usuario.
- Registro de cliente con identificador

Empleado #1: Se probaron ocho (8) funcionalidades dirigidas al empleado y fueron aprobadas sólo cinco (5) en esta primera iteración del ciclo de pruebas, siguen pendientes tres (3) funcionalidades dado las siguientes observaciones:

- Se muestran todos los usuarios creados.
- No se muestran todos los datos de los usuarios.
- No es editable los datos del usuario.

Empleado #2: Se probaron ocho (8) funcionalidades dirigidas al empleado y fueron aprobadas sólo cinco (5) en esta primera iteración del ciclo de pruebas, siguen pendientes tres (3) funcionalidades dado las siguientes observaciones:

- Se muestran todos los usuarios creados.
- No se muestra todos los datos de los usuarios.
- No es editable los datos del usuario.

Funcionalidades a ser evaluadas por el **técnico** en la iteración #1 son:

- Validación de datos de entrada.
- Envío de datos al servidor.
- Persistencia de sesiones.

Técnico #1: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas todas en esta primera iteración del ciclo de pruebas.

Técnico #2: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas todas en esta primera iteración del ciclo de pruebas.

No hay funcionalidad a ser evaluada por el **cliente** en la iteración #1 dado que no hay reportes, ni citas creadas.

ITERACIÓN #2

Funcionalidades a ser evaluadas por el **empleado** en la iteración #2 junto con las funcionalidades de las iteraciones anteriores son:

- Creación de reportes.
- Listado de reportes.
- Visualización de reporte.
- Cambio de estado en reporte.
- Creación de citas.
- Listado de citas de cliente.
- Lista de citas pertenecientes a un técnico.
- Visualización de cita

Empleado #1: Se probaron ocho (8) funcionalidades dirigidas al empleado y fueron aprobadas sólo seis (6) en esta segunda iteración del ciclo de pruebas, siguen pendientes dos (2) funcionalidades dado las siguientes observaciones:

- No se cambió el estado del reporte de manera automática al cambiar el estado de la última cita del reporte, se recargo para poder ver el cambio.
- La cita no muestra todos los datos.

Empleado #2: Se probaron ocho (8) funcionalidades dirigidas al empleado y fueron aprobadas sólo seis (6) en esta segunda iteración del ciclo de pruebas, siguen pendientes dos (2) funcionalidades dado las siguientes observaciones:

- No se cambia el estado del reporte de manera automática al cambiar el estado de la última cita en estado cerrado.
- El empleado cambio el estado de la cita, no es correcto, solo puede cambiarlo él técnico.

Funcionalidades a ser evaluadas por el **técnico** en la iteración #2 junto con las funcionalidades de las iteraciones anteriores son:

- Lista de citas pertenecientes a un técnico.
- Visualización de cita.
- Cambio de estado en la cita por parte de un técnico.

Técnico #1: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas sólo dos (2) en esta segunda iteración del ciclo de pruebas, siguen pendiente una (3) funcionalidad dado la siguiente observación:

- La organización de como se muestra la cita no es ordenada.

Técnico #2: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas sólo dos (2) en esta segunda iteración del ciclo de pruebas, siguen pendientes una (1) funcionalidad dado la siguiente observación:

- Mejorar estilos en la visualización de la cita.

Funcionalidad a ser evaluadas por el **cliente** en la iteración #2 es:

- Consultar reportes y citas por parte de cliente.

Cliente: Se probó una (1) funcionalidad dirigidas al cliente y fue aprobada. De igual forma se dio la siguiente observación:

- Mostrar el reporte a cuál pertenece la cita y datos del técnico asignado a la cita.

ITERACIÓN #3

Funcionalidades a ser evaluadas por el **empleado** en la iteración #3 junto con las funcionalidades de la iteración anteriores son:

- Calendario mostrando fechas de citas asignadas a un cliente.
- Ir al detalle de una cita desde calendario.
- Envío de correo automático por creación de usuario.
- Envío de correo automático a un cliente notificando creación del reporte, creación de citas, cambio en el estado del reporte y cambio en el estado de la cita.
- Agregar comentarios a la cita.
- Eliminar comentario por parte empleado.

Empleado #1: Se probaron seis (6) funcionalidades dirigidas al empleado y fueron aprobadas sólo seis (5) en esta tercera iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado la siguiente observación:

- No llega correo al cliente, ni técnico, cuando hay creación de reportes ó citas.

Empleado #2: Se probaron seis (6) funcionalidades dirigidas al empleado y fueron aprobadas sólo seis (5) en esta tercera iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado la siguiente observación:

- Él envió de correo automático no funciona cuando se crean reportes y citas.

Funcionalidades a ser evaluadas por el **técnico** en la iteración #3 junto con las funcionalidades de la iteración anteriores son:

- Envío de correo automático a un cliente notificando cambio en el estado del reporte y cambio en el estado de la cita.
- Agregar comentarios a cita.

Técnico #1: Se probaron dos (2) funcionalidades dirigidas al técnico y fueron aprobadas todas sin dar ninguna observación:

Técnico #2: Se probaron dos (2) funcionalidades dirigidas al técnico y fueron aprobadas todas sin dar ninguna observación:

Funcionalidad a ser evaluadas por el **cliente** en la iteración #3 es:

- Envío de correo automático por creación de usuario.
- Envío de correo automático a un cliente notificando creación del reporte, creación de citas y cambios en el estado del reporte y de la cita.

Cliente: Se probaron dos (2) funcionalidades dirigidas al técnico y fueron aprobadas todas sin dar ninguna observación

ITERACIÓN #4

Funcionalidades a ser evaluadas por el **empleado** en la iteración #4 junto con las funcionalidades de la iteración anteriores son:

- Subir archivo relacionado con cita.
- Descargar archivo relacionado con la cita.
- Envío de notificaciones en tiempo real aun técnico cuando le sea asignado una cita a un técnico.

Empleado #1: Se probaron tres (3) funcionalidades dirigidas al empleado y fueron aprobadas sólo dos (2) en esta iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado las siguientes observaciones:

- Permite la subida de cualquier tipo de archivo.
- Mejorar apariencia.

Empleado #2: Se probaron tres (3) funcionalidades dirigidas al empleado y fueron aprobadas sólo dos (2) en esta iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado la siguiente observación:

- Permite la subida de cualquier tipo de archivo.

Funcionalidades a ser evaluadas por el **técnico** en la iteración #4 junto con las funcionalidades de la iteración anterior son:

- Subir archivo relacionado con cita.
- Descargar archivo relacionado con la cita.

- Recepción de notificación en tiempo real aun técnico cuando se le sea asignado una cita.

Técnico #1: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas sólo dos (2) en esta iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado la siguiente observación:

- Permite cualquier tipo de archivo.

Técnico #2: Se probaron tres (3) funcionalidades dirigidas al técnico y fueron aprobadas sólo dos (2) en esta iteración del ciclo de pruebas, siguen pendiente (1) funcionalidad dado la siguiente observación:

- Se pueden subir cualquier archivo.

No hay funcionalidad a ser evaluadas por el **cliente** en la iteración #4.

Cliente: no hay funcionalidades nuevas que evaluar y las funcionalidades anteriores siguen funcionando correctamente.

ITERACIÓN #5

Durante esta última iteración de pruebas, todos los usuarios aprobaron las funcionalidades dirigidas a su rol de manera satisfactoria.

4.8.2 Pruebas de aceptación y usabilidad

Finalmente, se realizaron las pruebas de aceptación y usabilidad a una muestra de 15 usuarios internos y externos al sistema con edades comprendidas entre 20 y 32 años. Estas pruebas se hicieron para comprobar la tolerancia a fallas del sistema, la exactitud, el tiempo para concluir una actividad, respuesta emocional del usuario, y verificar que cumple adecuadamente con el proceso de automatización del control de citas dado un reporte de servicio técnico. Las pruebas de usabilidad implican una observación sistemática en condiciones controladas para determinar cómo las personas interactúan con el sistema. Para llevar a cabo estas pruebas, se construyeron un conjunto de instrumentos de evaluación presentados en la tabla 7, donde cada instrumento cuenta con cinco posibles respuestas a enunciados que realizan afirmaciones acerca de la aplicación.

		Totalmente de acuerdo	De acuerdo	Neutro	En desacuerdo	Totalmente desacuerdo
1	Las funciones del sistema son fáciles de usar					
2	Las acciones realizadas son sencillas de memorizar					
3	La aplicación es intuitiva					
4	El diseño de la aplicación es consistente					
5	Los colores del sistema son placenteros a la vista					
6	El sistema apoya al usuario en caso de errores					
7	El lenguaje empleado en los mensajes de usuario es claro					
8	La navegación en la aplicación es sencilla					
9	La aplicación se siente rápida y fluida					
10	El periodo de tiempo para terminar una actividad en la aplicación es breve.					
11	Finalizada una tarea su estado de ánimo es estable y controlado.					
12	La aplicación facilita la automatización y control de citas dado un reporte de servicio técnico					

Tabla 7: Instrumento de evaluación para pruebas de aceptación

4.8.2.1 Resultados de pruebas de aceptación

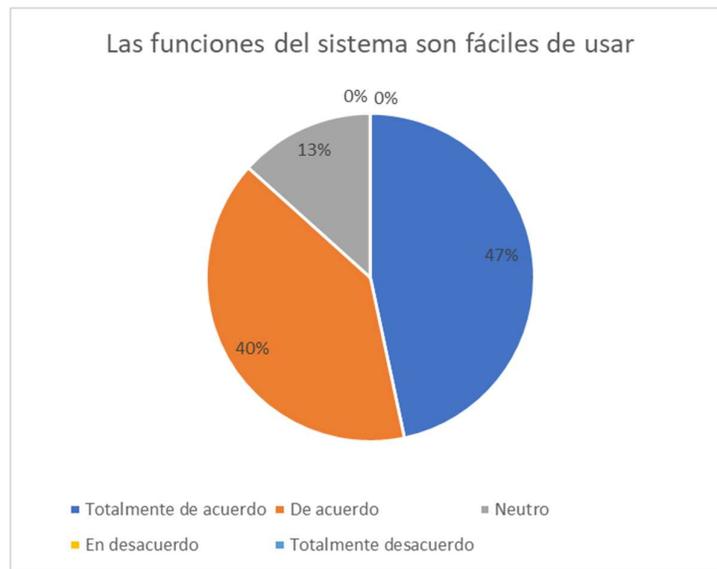


Ilustración 40: Resultados de pregunta 1 del instrumento de evaluación

El resultado de la encuesta muestra que las funciones son fáciles de usar para un 87% de encuestados, sumando 47% que están totalmente de acuerdo y 40% que están de acuerdo.

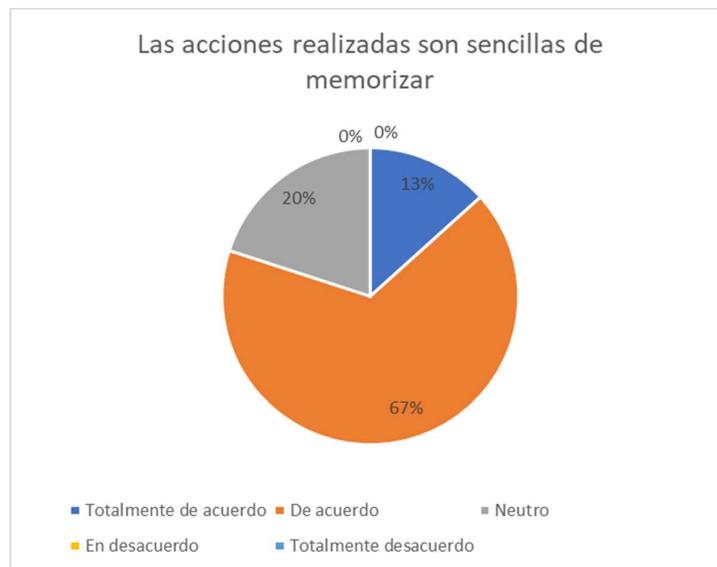


Ilustración 41: Resultados de pregunta 2 del instrumento de evaluación

Tomando en cuenta el resultado de la encuesta, se aprecia que las acciones realizadas son fáciles de memorizar luego de un periodo de tiempo sin usar la aplicación.

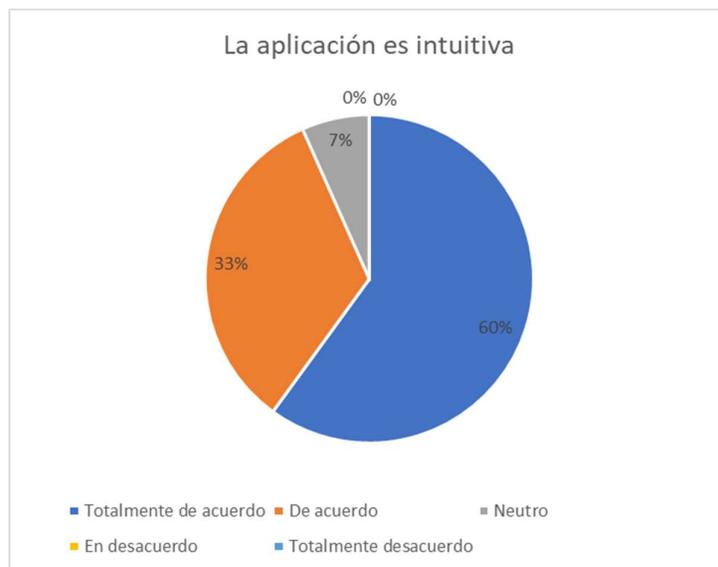


Ilustración 42: Resultados de pregunta 3 del instrumento de evaluación

La aplicación es intuitiva para más del 90% de la muestra, lo que hace cumplir con uno de los estándares de usabilidad.

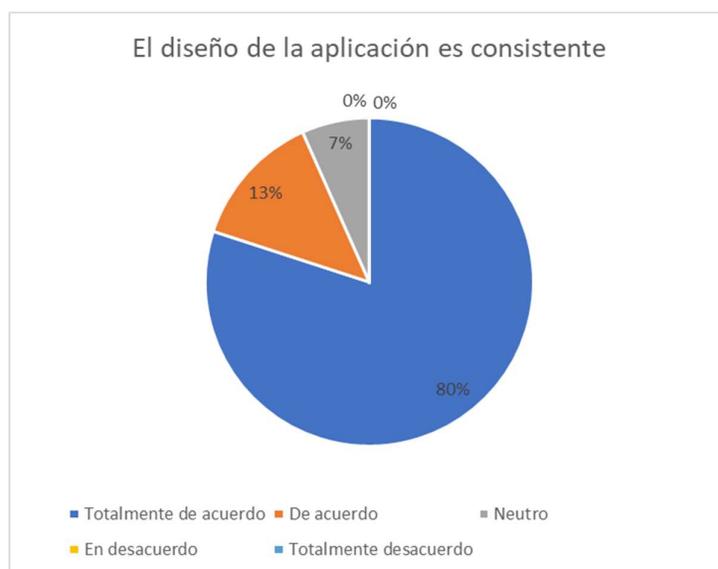


Ilustración 43: Resultados de pregunta 4 del instrumento de evaluación

Gracias a las tecnologías de diseño de aplicaciones web usadas en el sistema como Bootstrap, podemos tener una aplicación consistente en cuanto a diseño. Los resultados los muestran este gráfico.

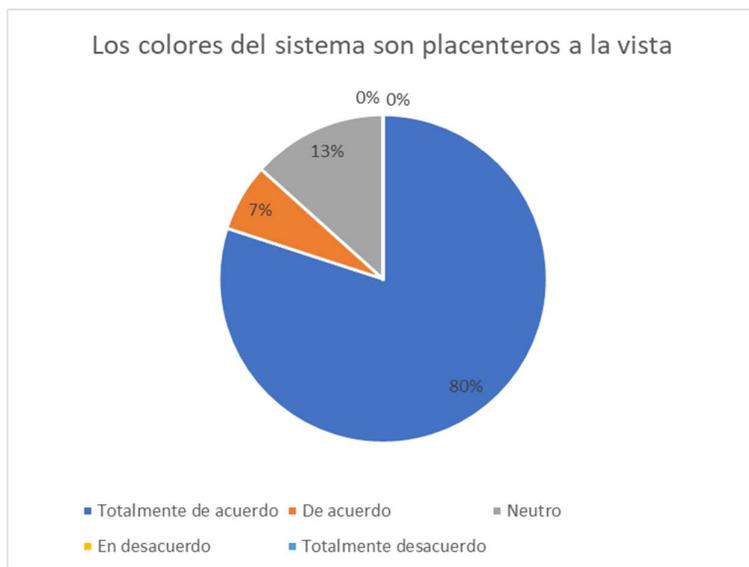


Ilustración 44: Resultados de pregunta 5 del instrumento de evaluación

La combinación de colores es placentera a la vista, sin embargo, tomando en cuenta que no causo gran impacto a un 13% de la muestra, se ajustaron detalles en secciones donde no se usaron colores de la paleta propuesta, buscando mejorar la experiencia visual.

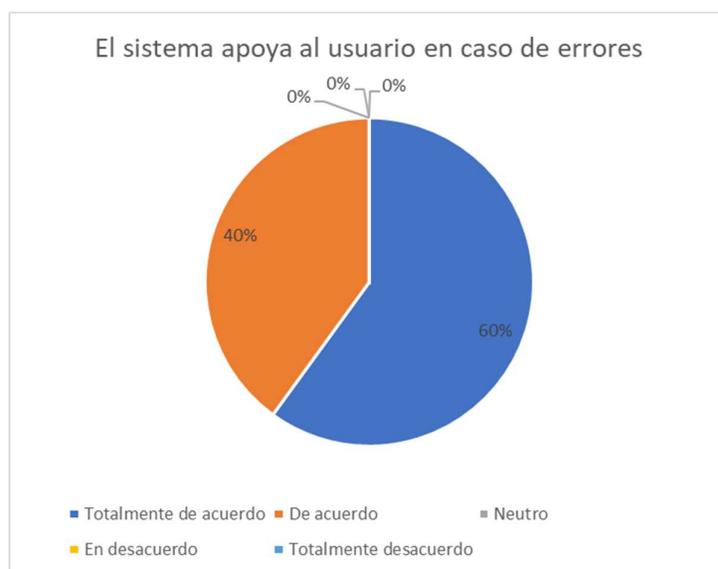


Ilustración 45: Resultados de pregunta 6 del instrumento de evaluación

En vista que totalidad de usuarios encuestados están totalmente de acuerdo o de acuerdo, se mantiene dentro del sistema el modo de indicar al usuario la presencia de errores.



Ilustración 46: Resultados de pregunta 7 del instrumento de evaluación

Podemos llegar a concluir en esta gráfica que la manera de expresarnos dentro de la aplicación tiene un lenguaje entendible por los usuarios.

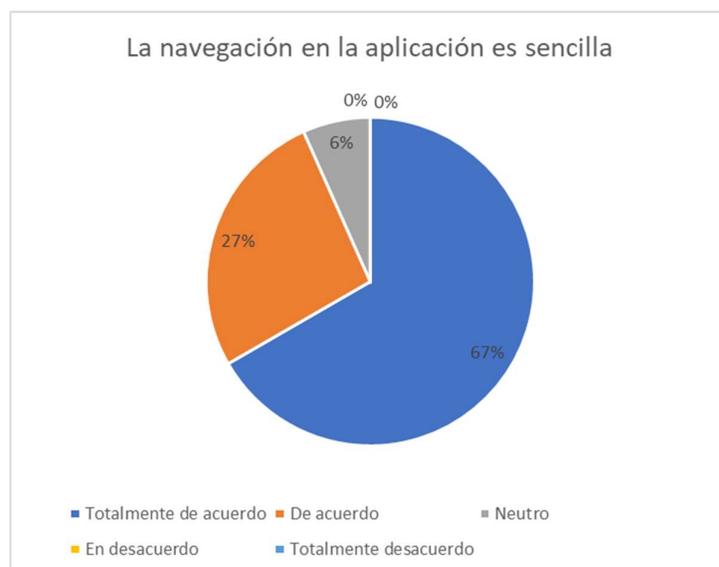


Ilustración 47: Resultados de pregunta 8 del instrumento de evaluación

Solo para un 6% de los usuarios, la navegación parece estar cerca de ser algo compleja, por consiguiente, se tomaron en cuenta sus respuestas, y se mejoraron varios aspectos visuales que permitieron hacer entendible y fácil la navegación del sistema.

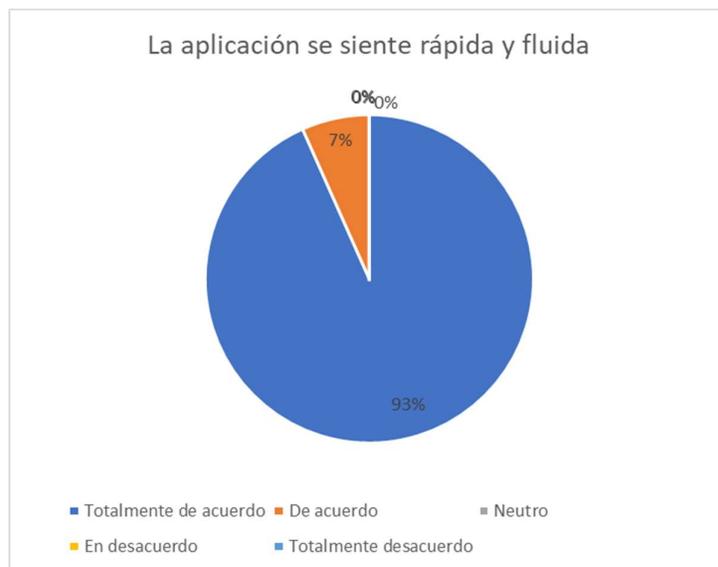


Ilustración 48: Resultados de pregunta 9 del instrumento de evaluación

El 93% de la muestra señaló que la aplicación es rápida y fluida, esto demuestra que las herramientas elegidas para el desarrollo de la aplicación forman un subsistema de tecnologías que funcionan de manera autosuficiente.



Ilustración 49: Resultados de pregunta 10 del instrumento de evaluación

Todas las actividades son tareas sumamente sencillas y el tiempo requerido para realizarla es muy corto, dado que la aplicación se enfoca en cada interfaz en una tarea en específica.

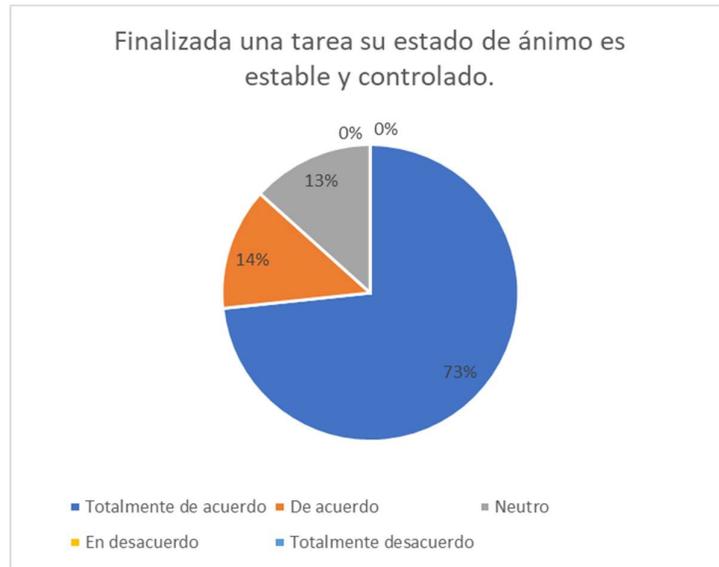


Ilustración 50: Resultados de pregunta 11 del instrumento de evaluación

Una vez que se finalizada la tarea, se evaluó el estado de ánimo de los usuarios y dado los resultados podemos concluir que el sistema no eleva los niveles de estrés.

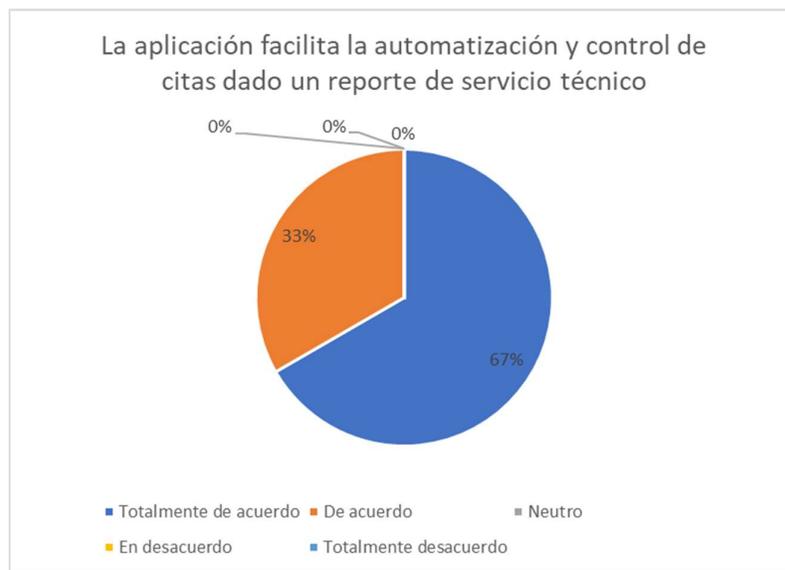


Ilustración 51: Resultados de pregunta 12 del instrumento de evaluación

De acuerdo a los resultados mostrados en este grafico de torta, podemos concluir que la aplicación cumple con su objetivo principal, ya que los usuarios respondieron afirmativamente a interrogante expuesta de que si la aplicación facilita la automatización y control de citas dado un reporte de servicio técnico.

CONCLUSIONES

En virtud de los resultados, se cumplió satisfactoriamente el objetivo del presente Trabajo Especial de Grado, el cual consistió en el desarrollo de una solución para el proceso de asignación de cita técnica programada de un proveedor de servicio de internet. Se realizó una aplicación web, la cual cumple con todos los requisitos planteados al principio del desarrollo.

Asimismo, se implementó el modelo de datos en mongoDB satisfactoriamente, permitiendo así una persistencia centralizada de los datos en el servidor. Se ejecutaron todas las pruebas funcionales a cada módulo del sistema comprobando así su óptimo funcionamiento. Por otro lado, con las pruebas de aceptación se obtuvieron resultados que permiten tener una retroalimentación de parte de los usuarios finales, con ello se puede tomar en cuenta los factores a mejorar en un futuro desarrollo o mantenimiento de la aplicación.

Haciendo uso de la metodología de desarrollo ágil SCRUM, se mantuvo una organización en cuanto a las tareas y actividades, así como con la programación de pruebas mejorando cada vez con cada iteración en la que participan los respectivos actores para mejorar el proceso de desarrollo del producto final.

Igualmente, el framework SailsJS fue de gran ayuda por su compatibilidad con el conjunto de tecnologías MEAN, facilidad del uso de socket, fácil manejo de base de datos mediante su ORM WaterlineJS, además de permitir la integración de AngularJS para el desarrollo del front-end, el cual provee un conjunto de librerías bastante útiles para el manejo, desarrollo y navegación de las interfaces. AngularJS fue de gran utilidad al contar con una comunidad en línea que mantiene vivo al framework.

Por último, utilizando LESS como pre-compilador de CSS se pudo tener ventajas al momento de agregar estilos y mejorar el aspecto visual de las interfaces.

En síntesis, el uso de todas estas tecnologías, estilos de arquitectura y metodologías de desarrollo permitieron cumplir con todos objetivos planteados para este proyecto.

RECOMENDACIONES

Las siguientes recomendaciones pueden mejorar el rendimiento del sistema, y servir de apoyo o referencia para trabajos futuros.

- Se recomienda el desarrollo de una aplicación móvil que le facilite a los técnicos, la gestión de las citas, indicando en un mapa la dirección y les permita a los empleados tener mayor seguimiento de los técnicos.
- Agregar gráficos de comportamiento para todo el sistema, que puedan medir el rendimiento a través del tiempo de los técnicos y ayude a la toma de decisiones en el sistema.
- Escalar la asignación de citas para especificar la hora y permitir que un técnico tenga más de una cita por día.
- Se recomienda realizar un cambio en el diseño para que se adapte siempre a la organización que quiera hacer uso de ella.
- Desarrollar alertas para los técnicos cuando una cita asignada pase a estatus vencido.
- Agregar recaptcha para mayor seguridad al entrar al sistema, evitando ataques de software maliciosos.

REFERENCIAS BIBLIOGRÁFICAS Y DIGITALES

- Acuña, B. (s.f.). *Eumed*. Obtenido de Metodologías tradicionales y metodologías ágiles: <http://www.eumed.net/libros-gratis/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>
- ADAM LITH, J. M. (2010). *Investigating storage solutions for large data*. Obtenido de ecma-international.: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>
- Aizaga, A. (s.f.). *Mofografias.com*. Obtenido de Base de datos: <https://www.monografias.com/trabajos55/base-de-datos/base-de-datos.shtml>
- Albarán, I. (2008). *La actividad aseguradora: importancia, revisión e integración de conceptos*. Recuperado el 03 de Noviembre de 2017, de <http://eprints.ucm.es/6723/1/0022.pdf>
- Amaro Calderón, S. D., & Valverde Rebaza, J. C. (2007). *Metologías Ágiles*. Tesis, Escuela de Informática, Trujillo.
- Baz Alonso, A., Ferreira Artime, I., Álvarez Rodríguez, M., & García Baniello, R. (2011). *Dispositivos móviles*. Universidad de Oviedo, Ingeniería de Telecomunicación. Recuperado el 13 de Octubre de 2017, de http://isa.uniovi.es/docencia/SIGC/pdf/telefonía_movil.pdf
- Burgos Cardemil, M. S. (s.f.). *Clasificación de los Sistema de Información*. Obtenido de <http://www.rafaelmellado.cl/material/inf3242/complemet/01.pdf>
- Castro Bernal, A. M. (21 de Septiembre de 2013). *Slidesahre*. Recuperado el 23 de Octubre de 2017, de Aplicaciones web: <https://www.slideshare.net/auramariacastro/aplicaciones-web-26419120>
- Concepto y definición de base de datos. (s.f.). En D. R. Llanos Ferraris, *Fundamentos de informática y programación en C*. Obtenido de <https://books.google.es/books?id=FfEFCB-hXCgC&pg=PT297&dq=base+datos+relacional+codd&hl=es&sa=X&ved=0ahUKEwjxLfyXcPXAhXMWhQKHQImAfUQ6AEIQjAF#v=onepage&q=SGBD&f=false>
- Domínguez-Dorado, M. (2008). Todo Programación N° 12. En M. Domínguez-Dorado, *Todo Programación N° 12* (págs. 48-51). Madrid: Editorial Iberprensa. Obtenido de JavaScript: <https://es.wikipedia.org/wiki/JavaScript>
- Elmasri, R., & Navathe, S. (2007). *Fundamentos de Sistemas de Bases de Datos* (Quinta ed.). Pearson Addison Wesley.
- Eneko. (s.f.). *Express.JS*. Obtenido de <http://enekodelatorre.com/expressjs-instalacion-primeros-pasos/>
- Fling, B. (2009). *Mobile Design and Development: Practical concepts and techniques for creating mobile sites*. O'Reilly Media. Recuperado el 17 de Diciembre de 2017
- Fundeu BBVA Fundación del español urgente. (7 de Agosto de 2012). Obtenido de <https://web.archive.org/web/20120807024401/http://www.fundeu.es/vademecum-W-www-928.html>
- Gonzalez-Longatt, F. M. (s.f.). *Universidad Veracruzana*. Obtenido de <https://www.uv.mx/personal/artulopez/files/2012/08/FundamentosSistemasInformacion.pdf>
- Google. (28 de 01 de 2018). *AngularJS: Development Guide Introduction*. Obtenido de AngularJS: <https://docs.angularjs.org/guide/introduction>
- Google. (s.f.). *AngularJS*. Obtenido de Developer Guide Introduction: <https://docs.angularjs.org/guide/introduction>
- Gutiérrez, E. G. (s.f.). *¿Qué es y para qué sirve HTML? El lenguaje más importante para crear páginas webs*. Obtenido de <https://www.aprenderaprogramar.es/attachments/article/435/CU00704B> Que es y para que sirve HTML lenguaje web mas importante.pdf

- Gutiérrez, J. (2014). *¿Qué es un framework web?* Recuperado el 26 de Octubre de 2017, de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- homeworkdatabase. (s.f.). *Sistema Manejador de Base de Datos*. Obtenido de <https://homeworkdatabase.wordpress.com/2015/06/27/sghbd-o-smbd-sistema-manejador-de-base-de-datos/>
- Johansen Bertoglio, O. (1993). *Introducción a la teoría general de los sistemas* (Primera ed.). México: Limusa S.A.
- Kapic Edin. (15 de Abril de 2016). Obtenido de JSON: <https://es.wikipedia.org/wiki/JSON>
- Kyocera Document Solutions. (s.f.). *Los 6 principales tipos de sistemas de información*. Obtenido de <https://smarterworkspaces.kyocera.es/blog/los-6-principales-tipos-sistemas-informacion/>
- Laudon, K., & Laudon, J. (2012). *Sistemas de Información Gerencial*.
- López Ortiz, F. (2010). *El estándar IEEE 802.11 Wireless LAN*. Obtenido de <http://www.dit.upm.es/~david/tar/trabajos2002/08-802.11-Francisco-Lopez-Ortiz-res.pdf>
- McWherter, J., & Gowell, S. (2012). *Professional Mobile Application Development*. Wrox. Recuperado el 17 de Diciembre de 2017
- Mozilla. (26 de 01 de 2018). *Introducción - Javascript MDN*. Obtenido de MDN Web Docs: <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>
- Mozilla. (s.f.). *CSS | MDN*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/CSS>
- Mozilla. (s.f.). *Introducción a XML | MDN*. Obtenido de https://developer.mozilla.org/es/docs/Introducci%C3%B3n_a_XML
- Orozco Tapia, A. M. (15 de marzo de 2012). *Modelos de BD*. Obtenido de PNFI – IUTEP 2012: <https://modelosbd2012t1.wordpress.com/2012/03/15/base-de-datos-moviles-3/>
- Palacio, J. (2014). *Gestión de Proyectos Scrum Manager*. Scrum Manager.
- Pandora FMS. (s.f.). Obtenido de NoSQL vs SQL: Principales diferencias y cuándo elegir cada una de ellas: <https://blog.pandorafms.org/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>
- Parra, I. (s.f.). *Uno de Piera*. Obtenido de Introducción a Sails.js: <https://www.uno-de-piera.com/introduccion-a-sails-js/>
- QBS Software Ltd. (2017). *qbs software*. Recuperado el Octubre de 2017, de http://www.qbssoftware.com/products/SQLBase/overview/_prodsqldb
- Ramírez Vique, R. (2011). *Métodos para el desarrollo de aplicaciones móviles*. Universidad Oberta de Catalunya. Recuperado el 26 de Octubre de 2017, de [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)
- Schwaber, K., & Sutherland, J. (2016). *La Guía de Scrum*. Recuperado el 24 de Octubre de 2017
- Silberschatz, A., Korth, H., & Sudarshan, S. (2002). *Fundamentos de Bases de Datos* (Cuarta ed.). Madrid, España: McGraw-Hill Inc.
- Silberschatz, A., Korth, H., & Sudarshan, S. (3 de Septiembre de 2006). *Fundamentos de Bases de Datos* (Quinta ed.). Madrid: Mc Graw Hill. Recuperado el 25 de Noviembre de 2017, de Portal de sitios web de asignaturas: http://www.ciens.ucv.ve/portaliasig/bases_de_datos/1-2015/descarga#
- Twitter. (s.f.). *Bootstrap*. Obtenido de <https://getbootstrap.com/>
- W3C. (June de 1999). *RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1* .
- Weizenbaum, N. (s.f.). *Wayback Machine*. Obtenido de Sass and Less: <http://web.archive.org/web/20090621074106/http://nex-3.com/posts/83-sass-and-less>
- Wikipedia. (s.f.). Obtenido de Simple Object Access Protocol: https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol