

“Diseño e implementación de un sistema de información web para el direccionamiento de pacientes basado en la clasificación TRIAGE para la atención en los diferentes servicios de urgencias de la ciudad de Popayán.”



ALEJANDRO RESTREPO SÁNCHEZ

JULIÁN DAVID GARCÍA PAZ

Director

Ing. Julio Mosquera

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS INFORMÁTICOS

“Diseño e implementación de un sistema de información web para el direccionamiento de pacientes basado en la clasificación TRIAGE para la atención en los diferentes servicios de urgencias de la ciudad de Popayán.”



ALEJANDRO RESTREPO SÁNCHEZ

JULIÁN DAVID GARCÍA PAZ

**Trabajo de Grado para optar al título de Ingenieros de Sistemas
Informáticos**

Director

Ing. Julio Mosquera

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS INFORMÁTICOS

AGRADECIMIENTOS

En primer lugar, quiero agradecer a Dios por haber sido mi guía en el cumplimiento de esta meta. A mis compañeros y docentes universitarios, por todos los conocimientos brindados para poder crecer día a día como ingeniero y primordialmente como persona.

A todo el personal médico que nos brindó su tiempo, espacio y conocimientos para poder llevar a cabo nuestro proyecto de grado.

A nuestro director de tesis por su entrega incondicional con nuestro trabajo, por su apoyo y por ser la persona que nos supo guiar desde el inicio hasta el final de este proyecto.

Agradecer a mis padres Elmer García Caicedo y Lucy Stella Paz Suarez, por haber estado día a día apoyándome y brindándome esa fuerza necesaria para afrontar aquellos momentos difíciles en el proceso de formación de esta carrera. Por esa lucha inquebrantable que no tiene precio alguno, pero si un valor incalculable para cada miembro de mi familia.

JULIAN DAVID GARCIA PAZ

Agradezco a Dios y a la Santísima Virgen por permitirme la oportunidad de culminar esta etapa de mi vida. A mi madre hermosa quien siempre creyó en mí, con quien luchamos todos estos años, quien ha sido mi compañía y mi ejemplo a seguir y a quien debo todo lo que soy, gracias madre querida por nunca rendirte conmigo y sobre todo por tu ayuda invaluable. A mi precioso hijo quien es mi motivación para poder ser mejor cada día gracias por llegar a mi vida. A mi familia quienes han jugado un papel importante en esta etapa; también a mis amigos y compañeros con quienes compartimos esta grata experiencia.

Gracias a todo el personal médico que de manera desinteresada colaboraron con nuestro proyecto brindándonos su tiempo, conocimiento y experiencia. Al ingeniero Julio Mosquera, nuestro director, gracias por ser nuestro guía, por el interés que tuvo siempre en nuestro proyecto y por su experiencia aportada en el mismo. Gracias a todo el personal de la Universidad quienes permiten el funcionamiento óptimo de la misma y quienes siempre estuvieron atentos para cualquier solicitud.

y un agradecimiento especial a Elifar Sánchez, mi abuelo, quien ya partió de este mundo y que en vida fue siempre mi padre, donde quiera que se encuentre, este logro también es suyo.

ALEJANDRO RESTREPO

Contenido

1. PLANTEAMIENTO DEL PROBLEMA.....	4
1.1 PREGUNTA DE INVESTIGACIÓN	5
1.2 JUSTIFICACIÓN	5
1.2.2 JUSTIFICACIÓN SOCIAL.....	6
1.3 OBJETIVOS	7
1.3.1 OBJETIVO GENERAL	7
1.3.2 OBJETIVOS ESPECÍFICOS.....	7
2. MARCO TEÓRICO	8
2.1 ANTECEDENTES	8
2.2 BASES TEÓRICAS.....	9
2.2.2 LENGUAJES DE PROGRAMACIÓN.....	11
2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE Y MARCO DE TRABAJO PARA LA GESTIÓN DEL PROYECTO	13
2.3.1 XTREME PROGRAMMING/XP	14
2.3.2 Marco de trabajo SCRUM.....	17
2.3.3 Patrón de diseño (MVC)	26
2.4 HERRAMIENTAS DE APOYO	28
2.4.1 BITBUCKET.....	28
2.4.2 GITHUB	29
2.4.3 Sublime text	30
2.4.4 Ems SQL Manager	31
2.4.5 PHPMYADMIN.....	32
2.4.6 XAMPP	33
2.4.7 CMDER.....	34
2.4.8 COMPOSER	35
2.4.9 ANDROID STUDIO.....	36
2.5.0 SCRUM PÓKER / PLANNING POKER	37

3. METODOLOGÍA.....	40
3.1 IMPLEMENTACIÓN DE SCRUM.....	40
3.1.1 PLANIFICACIÓN SPRINT 0	40
3.1.2 PLANIFICACIÓN SPRINT 1	41
3.1.3 PLANIFICACIÓN SPRINT 2	42
3.1.4 PLANIFICACIÓN DEL SPRINT 3	42
3.1.5 PLANIFICACIÓN DAILY SCRUM.....	42
3.1.6 SPRINT REVIEW.....	43
3.1.8 RETROSPECTIVA.....	43
3.2 XP (XTREME PROGRAMMING)	43
3.2.1 PLANIFICACIÓN	44
3.2.2 DISEÑO	57
3.2.3 CODIFICACIÓN.....	57
3.2.4 PRUEBAS.....	58
4. INGENIERÍA	59
4.1 IMPLEMENTACIÓN DEL SPRINT 0.....	59
4.1.1 ARQUITECTURA DEL SISTEMA	59
4.1.2 DIAGRAMA DE BASE DE DATOS	60
4.1.3 DESARROLLO DE AGILE INCEPTION	61
4.2 IMPLEMENTACIÓN SPRINT 1.....	64
4.2.1 PLANIFICACIÓN	65
4.2.2 DISEÑO	74
4.2.4 PRUEBAS.....	75
4.2.5 SPRINT REVIEW.....	76
4.2.6 RETROSPECTIVA.....	76
4.3 IMPLEMENTACIÓN SPRINT 2.....	76
4.3.1 PLANIFICACIÓN	77
4.3.2 DISEÑO	84
4.3.4 PRUEBAS.....	85
4.3.5 SPRINT REVIEW.....	86
4.3.6 RETROSPECTIVA.....	86

4.4 IMPLEMENTACIÓN SPRINT 3.....	87
4.4.1 PLANIFICACIÓN	87
4.4.2 DISEÑO	97
4.4.4 PRUEBAS.....	97
4.4.5 SPRINT REVIEW.....	98
4.4.6 RETROSPECTIVA.....	98
RESULTADOS	100
CONCLUSIONES.....	102
TRABAJO FUTURO.....	103

LISTA DE FIGURAS

Figura 1. Clasificación de TRIAGE	6
Figura 2. Framework laravel.....	12
Figura 3. Estructura de programación extrema XP	14
Figura 4. Flujo de trabajo de SCRUM.....	18
Figura 5. Ciclo desarrollo SCRUM	20
Figura 6. Eventos de SCRUM.	22
Figura 7. Ciclo del sprint – Daily SCRUM.....	23
Figura 8. Diagrama flujo SCRUM.	25
Figura 9. Diagrama de secuencia Modelo vista controlador.....	26
Figura 10. Bitbucket.....	28
Figura 11. Entorno GitHub.....	29
Figura 12. Entorno sublime text.....	30
Figura 13. Ambiente SQL manager for MySQL.....	31
Figura 14. Entorno PHPMyAdmiN.....	32
Figura 15. XAMPP.....	33
Figura 16. Entorno Cmder.....	34
Figura 17. Composer.....	35
Figura 18. Entorno Android Studio.	36
Figura 19. Planning póker.	37
Figura 20. Estructura cliente / servidor.....	59
Figura 21. Diagrama Base de Datos.	60
Figura 22. Interfaz de Usuario “Registrar Usuario”	74
Figura 23. Prueba Funcional Registrar Usuario.	75
Figura 24. Interfaz de Registrar Síntoma.....	84
Figura 25. Prueba Funcional Ingresar Síntoma.....	85
Figura 26. Interfaz de Usuario ver Estado de Hospitales.	97
Figura 27. Prueba Funcional.	98

LISTA DE TABLAS

Tabla 1. Metodología tradicional vs metodología ágil.....	9
Tabla 2. SCRUM master	40
Tabla 3. Development team	40
Tabla 4. Development team	41
Tabla 5. Product owner	41
Tabla 6. Retrospectiva del sprint 0, proyecto en general.....	64
Tabla 7. Historia de usuario autenticar usuario	65
Tabla 8. Historia de usuario Registrar Usuario.....	67
Tabla 9. Historia de Usuario Actualizar Usuario.....	69
Tabla 10. Historia de Usuario Eliminar Usuario.....	71
Tabla 11. Historia de Usuario Recuperar Contraseña.....	73
Tabla 12. Prueba funcional.....	75
Tabla 13. Retrospectiva del sprint 1.....	76
Tabla 14. Historia de usuario Registrar Síntoma.....	77
Tabla 15. Historia de Usuario Actualizar Síntoma.....	79
Tabla 16. Historia de Usuario Eliminar Síntoma.....	81
Tabla 17. Historia de usuario Generar nivel de urgencia.....	83
Tabla 18. Prueba funcional registrar síntoma.....	85
Tabla 19. Retrospectiva del sprint 2.....	86
Tabla 20. Historia de usuario consultar síntoma.....	87
Tabla 21. Historia de usuario Registrar Hospital.....	89
Tabla 22. Historia de Usuario Actualizar Hospital.....	91
Tabla 23. Historia de Usuario Eliminar Hospital.....	93
Tabla 24. Historia de usuario consultar información de hospital.....	95
Tabla 25. Registrar Calificación de Hospital.....	96
Tabla 26. Prueba funcional.....	97
Tabla 27. Retrospectiva del sprint 3.....	99

LISTA DE GRAFICAS

Gráfica 1. Morbilidad Municipio De Popayán 2015.....	45
Gráfica 2. Morbilidad Municipio De Popayán 2016.....	46
Gráfica 3. Morbilidad Municipio De Popayán 2017.....	47
Gráfica 4. Morbilidad Municipio De Popayán 2018.....	48
Gráfica 5. Morbilidad año 2018 hospital Toribio maya	49
Gráfica 6. Morbilidad año 2017 hospital Toribio maya	50
Gráfica 7. Morbilidad año 2016 hospital Toribio maya	51
Gráfica 8. Morbilidad año 2015 hospital Toribio maya	52
Gráfica 9. Morbilidad año 2015 hospital San José	53
Gráfica 10. Morbilidad año 2016 hospital San José	54
Gráfica 11. Morbilidad año 2017 hospital San José	55
Gráfica 12. Morbilidad año 2018 hospital San José	56
Gráfica 13. Reporte alermed síntomas más consultados mes a mes.....	100
Gráfica 14. Reporte alermed síntomas más consultados mes a mes.....	101
Gráfica 15. Reporte alermed síntomas más consultados mes a mes.....	101

RESUMEN

El presente proyecto de grado está dirigido a toda la comunidad de la ciudad de Popayán, el desarrollo de éste inicia con una investigación que proporciona algunos datos relevantes acerca de los motivos de consulta en la ciudad, trabajando de la mano con el marco de trabajo SCRUM y la metodología XP (Extreme Programming). El proyecto se enfoca en el desarrollo de una plataforma web y a su vez móvil en la cual los usuarios finales pueden tener información útil para saber dónde pueden dirigirse en caso de una urgencia médica, ingresando sus síntomas y basándose en la ubicación geográfica. Ya que el diagnóstico médico es un tema muy subjetivo, esta plataforma se usará como una guía que pueda aportar en la descongestión de los servicios de urgencias y en la atención oportuna de los pacientes.

Palabras Clave: TRIAGE, plataforma Web, Móvil, Pacientes, Metodologías Ágiles, SCRUM

ABSTRACT

The present project of degree is directed to the whole community of the city of Popayán, the development of this starts with a research that provides some relevant data about the reasons for consultation in the city, working hand in hand with the SCRUM framework and the XP (Extreme Programming) methodology. The project focuses on the development of a web and mobile platform in which end users can have useful information to know where they can go in case of a medical emergency, entering their symptoms and based on geographical location. Since the medical diagnosis is a very subjective issue, this platform will be used as a guide that can contribute to the decongestion of emergency services and the timely care of patients.

Key Words: TRIAGE, Web platform, Mobile, Patients, Agile Methodologies, SCRUM

INTRODUCCIÓN

En reiteradas ocasiones y por diferentes motivos, las personas se han visto en la necesidad de visitar una sala de urgencias, ya sea de manera directa o indirecta, es un lugar del cual se podría deducir, todos han conocido. Se ha podido evidenciar el colapso de las mismas por disímiles motivos, ya que el problema no es reciente; las causas son innumerables y los afectados van en sentido bilateral, debido que no solo la comunidad, como usuarios del sistema de salud y en este caso de los servicios de urgencias, se ve afectada por los tiempos de espera o la mala calidad de la atención, sino también el personal asistencial, como los médicos, enfermeras y auxiliares, teniendo en cuenta que se enfrentan a múltiples prejuicios de los usuarios y sanciones tanto disciplinarias como económicas, pues aparte de la prestación del servicio como profesionales en el área de la salud, deben justificar los procedimientos en pro de no adquirir algún tipo de inconveniente con la parte administrativa, especialmente con la facturación de las cuentas médicas de cada paciente atendido.

Por lo anterior y en vista de las múltiples soluciones software que hoy en día existen y que son usadas para mejorar o tecnificar nuestra cotidianidad, se ha encontrado no solo una propuesta como trabajo de grado, sino también una forma de aportar desde esta área de conocimiento y campo profesional, a una problemática que desde antaño afecta la ciudad y el país. Es por esto que la motivación principal para el desarrollo de este trabajo es aportar a la comunidad de la ciudad de Popayán un mejor servicio y más eficaz en el área de urgencias.

1. PLANTEAMIENTO DEL PROBLEMA

El sistema de salud colombiano, actualmente regido por la Ley 100 de 1993, [1] es en el papel, un sistema adecuado para el país, puesto que según esta, el Estado tiene la capacidad de suplir las necesidades de la población y aunque tenga connotaciones importantes que benefician a los usuarios, la realidad que atañe a nuestro país es que, como toda ley en Colombia tiene “vicios” y vacíos legales que son aprovechados por diferentes actores, forjando un sistema de salud deficiente en muchos aspectos [2]. Esta noción es presuntamente la misma para los usuarios, quienes consideran que las EPS e IPS tienen una atención deshumanizada con sus afiliados, donde una de las principales molestias es la atención en citas médicas y el servicio de urgencias [3].

En Popayán la situación no se aleja de la realidad, la ciudad cuenta con seis centros hospitalarios que prestan el servicio de urgencias, no solo del municipio sino también del departamento del Cauca y otros departamentos del suroccidente colombiano; en ocasiones un gran número de pacientes llegan remitidos de sus respectivos centros asistenciales y es este un motivo más por el cual se pueden encontrar las salas de urgencias colapsadas.

Según el informe de quejas de los años 2015 y 2016 presentado por el Sistema Integral de Atención al Usuario SIAU a la oficina de control interno del Hospital Universitario San José de Popayán [4], una de las principales molestias informadas por los ciudadanos es no recibir atención oportuna en el servicio de urgencias y que es generada debido a que la relación entre la cantidad de pacientes para ser atendidos y el personal que cumple con esta función, se torna desproporcionada[5,6], originando malestar entre los mismos sobre todo en los pacientes quienes esperan una atención rápida y oportuna.

La Asociación Colombiana de Hospitales y Clínicas ACHC, realizó una investigación de Servicios de Urgencias en 27 IPS, encontrando un aumento en las consultas y del total obtenido se pudo evidenciar que el 46% no eran tan urgentes[7]; en muchos casos, los pacientes se dirigen a los diferentes centros asistenciales desconociendo varios factores que podrían ser tenidos en cuenta para su atención en dichas instituciones, como son: la contratación de su EPS con la IPS, la clasificación de TRIAGE[8] de su padecimiento y el nivel en el que puede ser atendido dependiendo de dicha clasificación.

1.1 PREGUNTA DE INVESTIGACIÓN

¿Cómo se pueden apoyar los procesos de atención de urgencias en las instituciones prestadoras de este servicio en la ciudad de Popayán, que siguen la normatividad de TRIAGE, haciendo uso de las tecnologías de la información y las comunicaciones como lo son las aplicaciones móviles?

1.2 JUSTIFICACIÓN

1.2.1 JUSTIFICACIÓN TÉCNICA

Con el desarrollo de este proyecto se pretende brindar a la comunidad del municipio de Popayán, una solución tecnológica que permita formar un correcto direccionamiento basado en la clasificación TRIAGE y en los niveles de complejidad, el cual es criterio importante para la clasificación de pacientes en el servicio de urgencias y que es de obligatorio cumplimiento por parte de los prestadores de servicios de salud. El TRIAGE tiene como objetivo asegurar una valoración oportuna y ordenada de todos los pacientes que acceden a los servicios de urgencias, principalmente identifica la urgencia clínica y de esta forma busca disminuir el riesgo de complicaciones, discapacidad o muerte. Se encuentra clasificado en 5 categorías, las tres primeras incluyen atención en servicio de urgencias y las dos restantes atención en consulta externa donde la EPS del paciente ofrezca este tipo de servicios. Es por esto que se desarrollará una aplicación que sea capaz de direccionar, teniendo en cuenta la clasificación TRIAGE, a quienes la consulten con la asistencia de un cuestionario, de esta manera poder optimizar los tiempos en los sistemas de atención en urgencias de la ciudad de Popayán.

En la siguiente Figura 1. Se detallan las cinco categorías de El TRIAGE, sus características y su respectivo tiempo de atención para cada una de ellas,

Figura 1. Clasificación de TRIAGE



Fuente [9]

En la parte funcional de la plataforma se cuenta con el framework laravel por su facilidad para adaptarse a los cambios y para la parte visual se optó por usar HTML, CSS y trabajando de la mano con el lenguaje de programación PHP que brinda la posibilidad de que el sistema de información web sea adaptativo. El diseño web adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos (equipos de escritorio, tabletas, celulares, entre otros).

1.2.2 JUSTIFICACIÓN SOCIAL

Trasladarse a un centro asistencial más que una forma de alivio puede considerarse un proceso tortuoso para la comunidad, debido a las demoras en la atención o a las inconsistencias en la afiliación al sistema de salud [10], estos impases en la mayoría de los casos son por causas ajenas al personal médico o a la institución. El desconocimiento del nivel de complejidad del centro al que se decide asistir conlleva

a que se tome una decisión arbitraria, teniendo en cuenta que se ignora la clasificación de TRIAGE de su padecimiento, el nivel de complejidad correcto para su atención [11] y la contratación que tiene la EPS a la cual se encuentra afiliado, esto último es de gran importancia para la permanencia de un paciente dentro del centro hospitalario.

Los afiliados no tienen el hábito de mantenerse al tanto de los anteriores aspectos y las EPS por su parte, en muchos de los casos, pasan por alto brindar esta información de una manera más expedita; es así como las salas de urgencias se llenan de pacientes que no requieren del nivel de complejidad a las cuales están asistiendo y que pueden ser atendidas en otro centro de menor complejidad, dichos pacientes retrasan de manera indirecta la atención, colapsando el servicio, impidiendo la puntual revisión de otros pacientes, generando un mal diagnóstico o uno apresurado y en el peor de los casos la muerte. Es por esto que si la comunidad obtiene un conocimiento previo y actualizado de un correcto direccionamiento, tendrían opciones para la elección del centro asistencial adecuado al cual concurrir, de esta manera se pueden evitar los traslados, las consultas innecesarias y los colapsos en los servicios de urgencias de los centros de más alta complejidad, beneficiando al personal médico, a los usuarios y a la parte financiera de los centros médicos, ya que las EPS que son las entidades encargadas del pago de las cuentas médicas, en algunas ocasiones glosan dichas cuentas argumentando que el paciente no requería de atención en este nivel.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar e implementar un sistema de información web/móvil para el direccionamiento de pacientes basado en la clasificación TRIAGE para la atención en los diferentes servicios de urgencias de la ciudad de Popayán.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analizar los antecedentes epidemiológicos de la ciudad de Popayán dentro de un conjunto de datos estadísticos proporcionados por los centros hospitalarios para conocer los motivos de consulta más recurrentes y poder hacer mayor énfasis en los mismos.
- Identificar los aspectos relevantes de la semiología médica para el diagnóstico de enfermedades.
- Recomendar por medio de la aplicación móvil el centro médico donde se podrá atender al usuario teniendo en cuenta la ubicación geográfica y el nivel de complejidad necesario para su atención,

2. MARCO TEÓRICO

2.1 ANTECEDENTES

Han sido encontrados algunos desarrollos software web y móvil que realizan el proceso del diagnóstico médico, todas aclaran que este diagnóstico jamás reemplaza al que es dictado por profesionales del área de la salud, como lo es el obtenido en una consulta presencial en un consultorio médico en el área de urgencias y/o consulta externa de cualquier centro hospitalario.

A continuación, se listarán los desarrollos más relevantes encontrados en internet:

Symcat: Es una calculadora de enfermedades que utiliza cientos de miles de registros de pacientes para estimar la probabilidad de enfermedad. Es un diagnóstico basado en datos. Esta herramienta es solo para fines educativos. No está destinado a fines de diagnóstico. Solo un médico o proveedor autorizado puede diagnosticar y tratar enfermedades.

Dificultades que presenta: por ser originalmente de Estados Unidos, su lenguaje es en inglés, además de que su finalidad es solo brindar diagnósticos de la dolencia sin dar una posible alternativa de solución.

Isabel Symptomchecker: Esta aplicación utiliza la base de datos del Isabel Healthcare para realizar una lista de verificación y listar los posibles diagnósticos para posteriormente brindar una clasificación por colores junto a una leyenda que informa al paciente si ir o no a un centro médico.

Isabel Symptomchecker toma todos sus síntomas y devuelve una lista de posibles diagnósticos que podrían ser la causa. Cada diagnóstico se vincula con un extenso conocimiento médico para ayudarlo a aprender más.

Dificultades que presenta: El lenguaje que usa es inglés, además que su diagnóstico no es legible por un usuario sin conocimientos médicos.

Figure 1: Esta aplicación móvil gratuita disponible para Android y IOS es dirigida a personal médico, en ella se pueden compartir casos médicos para ser resueltos entre los usuarios de la aplicación brindando así un diagnóstico acertado.

Dificultades que presenta: Al igual que Symcat e Isabel Symptomchecker la mayoría de las aplicaciones existentes están desarrolladas en Estados Unidos, por lo cual se encuentran en idioma inglés y sus datos son obtenidos de las bases de datos de centros médicos que se encuentran en ese país, por lo cual dificulta el uso de estas en la zona de la investigación de este proyecto. Cabe resaltar que la legislación colombiana en el tema de la salud es totalmente diferente, por lo que no se tiene en

cuenta el tema de las Entidades Promotoras de Servicios (EPS) ni los niveles de complejidad de los centros médicos hospitalarios que aquí se encuentran.

2.2 BASES TEÓRICAS

2.2.1.1 METODOLOGÍAS Y MANIFIESTO ÁGIL

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software, el objetivo fue diseñar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto, se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales caracterizados por ser rígidos. [12]

Antes de continuar, se plantean las principales diferencias respecto de las metodologías tradicionales y las metodologías ágiles, como se muestra en la Tabla 1, en la cual se observan las fases por las cuales pasa un desarrollo en cada metodología.

Tabla 1. Metodología tradicional vs metodología ágil

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente [12]

Metodología tradicional: En la metodología tradicional cada proyecto pasa por un planteamiento, análisis, diseño, programación, pruebas y posteriormente, puesto en marcha.

Metodología ágil: En la metodología ágil cada proyecto pasa por un planteamiento, toma de requerimientos ya priorizados para posteriormente tener una iteración constante en la parte del diseño, programación y pruebas esto con el fin de adaptarse a cambios y finalmente ser puesto en marcha

2.2.1.2 MANIFIESTO ÁGIL

El manifiesto ágil valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funciona más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

2.2.1.3 PRINCIPIOS

- ✓ La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- ✓ Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- ✓ Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre las entregas.
- ✓ La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- ✓ Construir el proyecto en torno a individuos motivados, darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- ✓ El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- ✓ El software que funciona es la medida principal del progreso.
- ✓ Los procesos ágiles promueven un desarrollo sostenible, los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- ✓ La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- ✓ La simplicidad es esencial.
- ✓ Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- ✓ En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo. Y según esto ajusta su comportamiento. [12]

2.2.2 LENGUAJES DE PROGRAMACIÓN

2.2.2.1 PHP: Hyper text preprocessor

Es un lenguaje de programación interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (*open source*). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML.

Es un popular lenguaje de *scripting* de propósito general que es especialmente adecuado para el desarrollo web.

Rápido, flexible y pragmático, PHP impulsa todo, desde su blog hasta los sitios web más populares del mundo. [13]

2.2.2.2 JAVASCRIPT:

Es un lenguaje interpretado basado en guiones que son integrados directamente en el código HTML. El código es transferido al cliente para que este lo interprete al cargar la página. Con JavaScript no pueden crearse programas independientes.

Principales características de JavaScript:

- ✓ Es un lenguaje interpretado.
- ✓ No necesita compilación.
- ✓ Multiplataforma.
- ✓ Lenguaje de alto nivel.
- ✓ Admite programación estructurada.
- ✓ Basado en objetos.
- ✓ Maneja la mayoría de los eventos que se pueden producir sobre la página web.
- ✓ No necesita ningún kit o entorno de desarrollo.

2.2.2.3 FRAMEWORK

En general, el término framework, hace referencia a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework es considerado como una aplicación genérica incompleta y configurable a la que es posible añadirle las últimas piezas para construir una aplicación concreta. Es un conjunto de componentes (por ejemplo, clases en java y descripciones y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. [14]

Objetivos principales de un framework:

- ✓ Acelerar el proceso de desarrollo.
- ✓ Reutilizar código ya existente
- ✓ Promover buenas prácticas de desarrollo como el uso de patrones.

2.2.2.4 LARAVEL

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP5 y PHP7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el “código espagueti”, este nombre se usó para designar aquellos programas cuyo flujo de ejecución es demasiado enredado convirtiéndolo en algo tedioso e imposible de seguir.

Se trata de un framework de desarrollo con una curva de aprendizaje muy rápida y que maneja una sintaxis expresiva, elegante, con el objetivo de eliminar la molestia del desarrollo web facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y cache. [15]

Figura 2. Framework laravel



Fuente [16]

Ventajas de usar el framework laravel:

- ✓ Reducción de costos y tiempos en el desarrollo y posterior mantenimiento de la aplicación creada.
- ✓ Flexible y adaptable no solo al uso del sistema MVC (Modelo Vista Controlador) tradicional, sino que para reducir las líneas de código propone lo que denomina rutas con cierres (*routes with clousures*).
- ✓ Facilita el manejo de las rutas de nuestra aplicación, así como la generación de URLs (Localizador uniforme de recursos) amigables que ayudan a mejorar el posicionamiento web.
- ✓ Una gran comunidad y mucha documentación. Sobre todo, su sitio oficial.

2.2.2.5 MYSQL

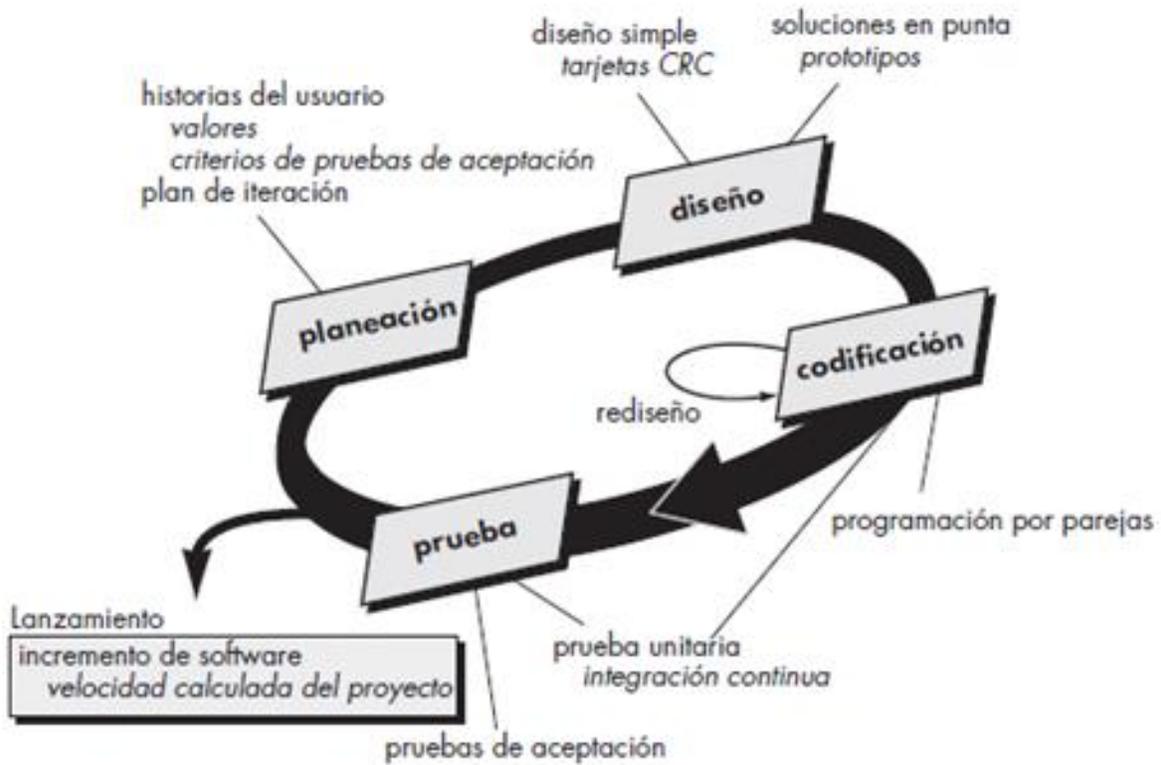
Es un sistema gestor de bases de datos. Pero la virtud fundamental y la clave de su éxito es que se trata de un sistema de libre distribución y de código abierto (*open source*), significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Esa es también la base del funcionamiento del sistema Linux, es por eso que MYSQL se distribuye fundamentalmente para Linux, aunque también hay versiones para Windows. [17]

2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE Y MARCO DE TRABAJO PARA LA GESTIÓN DEL PROYECTO

2.3.1 XTREME PROGRAMMING/XP

XP, es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en un desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo, XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes donde existe un alto riesgo técnico. [18]

Figura 3. Estructura de programación extrema XP



Fuente [19]

2.3.1.1 Principios básicos de XP

XP, es una metodología ágil que tiene como pilares el trabajo en equipo brinda 10 prácticas que involucran a todo el equipo. Estas prácticas garantizan un proyecto con disposición al cambio.

- Planificación incremental: Se toman los requerimientos en historias de usuario las cuales son negociadas progresivamente con el cliente.
- Entregas pequeñas: Se desarrolla primero la más mínima útil que le proporcione funcionalidad al sistema, y poco a poco se efectúan incrementos que añaden funcionalidad a la primera entrega, cada ciclo termina con una entrega del sistema.
- Diseño sencillo: Solo se efectúa el diseño necesario para cumplir con los requerimientos actuales, es decir, no se abordan requerimientos futuros.
- Desarrollo previamente aprobado: Una de las características relevantes y propias de XP es que primero se escriben las pruebas y luego se da la codificación, esto con la finalidad de asegurar la satisfacción del requerimiento.
- Limpieza del código o refactorización: Consiste en simplificar y optimizar el programa sin perder funcionalidad, es decir, alterar su estructura interna sin afectar su comportamiento externo.
- Programación en parejas: Es otra de las características de esta metodología, que propone que los desarrolladores trabajen en parejas en una terminal, verificando cada uno el trabajo del otro y ayudándose para buscar las mejores soluciones. Se entiende que de esta forma el trabajo será más eficiente y de mayor calidad.
- Propiedad colectiva: El conocimiento y la información deben ser de todos, por lo tanto, no se desarrollan islas de conocimiento, todos los programadores poseen todo el código y cualquiera puede sugerir y realizar mejoras.
- Integración continua: Al terminar una tarea, esta se integra al sistema entero y se realizan pruebas de unidad a todo el sistema, esta práctica permite que la aplicación sea más funcional en cada iteración y garantiza su funcionamiento con los demás módulos del sistema.
- Ritmo sostenible: No es aceptable trabajar durante grandes cantidades de horas ya que se considera que puede reducir la calidad del código y productividad del equipo a mediano plazo, se sugieren 40 horas semanales.
- Cliente presente: Se debe tener un representante (Cliente o usuario final) tiempo completo, ya que en XP este hace parte del equipo de desarrollo y es responsable de formular los requerimientos para el desarrollo del sistema. [19]

2.3.1.2 Roles de XP

A continuación, los roles que la metodología XP brinda:

- Programador: escribe las pruebas unitarias y produce el código del sistema.

- Cliente: escribe las historias de usuario y las pruebas funcionales para validar su implementación, además asigna la prioridad a las historias de usuario y decide cuales se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- Encargado de pruebas (*Tester*): ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas, regularmente difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- Encargado de seguimiento (*Tracker*): proporciona realimentación al equipo, verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones y realiza el seguimiento del progreso de cada iteración.
- Entrenador (*Coach*): responsable del proceso global, debe proveer guías al equipo de forma que se apliquen las prácticas XP.
- Consultor: es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- Gestor (*Big Boss*): es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación. [12]
-

2.3.1.3 Fases de XP

Al igual que las metodologías ágiles, el proceso XP (Figura 3), cuenta con cuatro fases que al ser concluidas dan origen a una versión del producto software:

- Fase de planeación: inicia con las historias de usuario que describen las características y funcionalidades del software, el cliente asigna un valor o prioridad a la historia, los desarrolladores evalúan cada historia y le determinan un costo, el cual se mide en semanas de desarrollo.
- Fase de diseño: el proceso de diseño debe procurar diseños simples y sencillos para facilitar el desarrollo, se recomienda elaborar un glosario de términos y la correcta especificación de métodos y clases para facilitar posteriores modificaciones, ampliaciones o reutilización de código. Anteriormente este proceso se apoyaba en el uso de tarjetas CRC (colaborador – responsabilidad - clase) la cual identifica las clases orientadas a objetos que son relevantes para el incremento del software.
- Fase codificación: en esta, los desarrolladores deben diseñar las pruebas de unidad que ejercitarán cada historia de usuario. Después de tener las pruebas, los

desarrolladores trabajarán en parejas para concentrarse en lo que debe implementarse para pasar la prueba de unidad.

- Fase de pruebas: las pruebas de unidad deben implementarse con un marco de trabajo que permita automatizarlas, con la finalidad de realizar pruebas de integración y validación diarias. Esto proporciona al equipo un indicador del progreso y revelará a tiempo si existe alguna falla en el sistema. [19]

2.3.1.4 Valores de XP

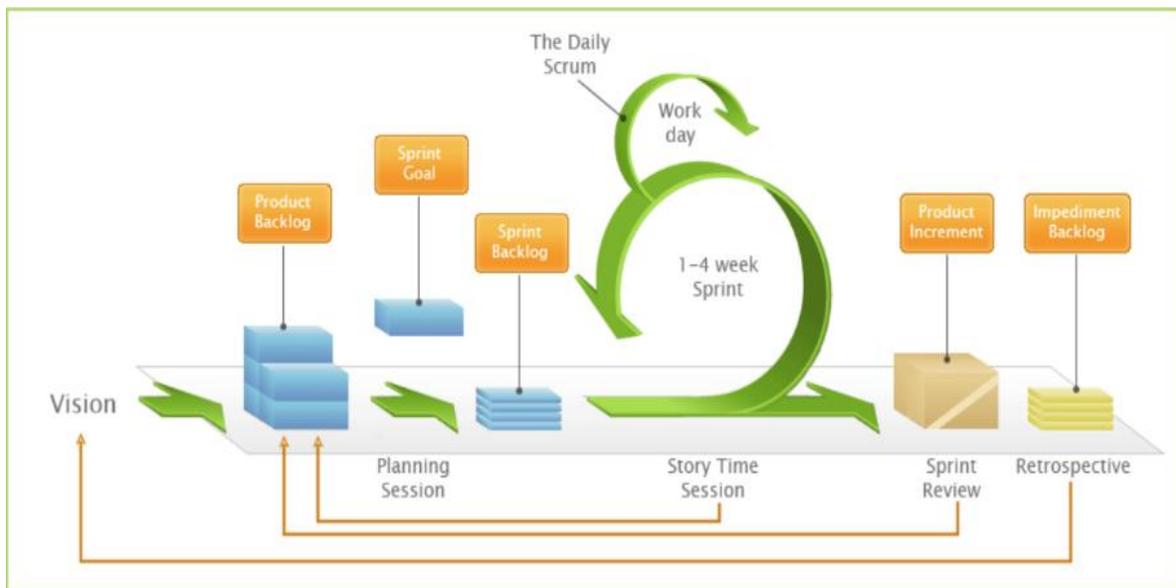
La metodología ágil XP, mide su efectividad y producción en base a cuatro valores que van a garantizar en todo desarrollo el éxito del proyecto. A continuación, se resume brevemente su finalidad:

- **Comunicación:** debe ser fluida entre los participantes en el proyecto; además el entorno tiene que favorecer la comunicación espontánea, ubicando a todos los miembros en un mismo lugar, la comunicación directa aporta mucho más valor que la escrita, se puede observar los gestos del cliente o la expresión de cansancio de los compañeros.
- **Simplicidad:** cuanto más sencilla sea la solución, más fácil será adaptarla a los cambios. Las complejidades aumentarán el coste del cambio y disminuyen la calidad del software.
- **Realimentación:** el usuario debe utilizar desde la primera entrega el software desarrollado, manifestando sus impresiones y sus necesidades no satisfechas, de manera que estas historias vuelvan a formar parte de los requisitos del sistema.
- **Coraje:** coraje para vencer las frases más típicas de los desarrolladores: si funciona no lo toques. Con XP se debe manipular continuamente las cosas que ya funcionan con el fin de mejorarlas. [20]

2.3.2 Marco de trabajo SCRUM

SCRUM es adaptable, ágil, auto organizado y con pocos tiempos muertos, puede ser aplicado teóricamente a cualquier contexto en el que un grupo de individuos necesita trabajar para lograr una meta en común. Cabe resaltar que SCRUM más que una metodología es un MARCO DE TRABAJO el cual sirve de guía para orientar proyectos.

Figura 4. Flujo de trabajo de SCRUM



Fuente [22]

SCRUM es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Pone su atención y hace foco sobre valores prácticos de gestión, en vez de requerimientos, prácticas de desarrollo, implementación y otras cuestiones técnicas, esta metodología delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles, es decir que es flexible y los integrantes del equipo pueden optar por organizar la forma de interactuar entre ellos. [21]

2.3.2.1 Principios de SCRUM

- Privilegiar el valor de la gente sobre el valor de los procesos.
- Entregar el software funcional lo más pronto posible.

- Predisposición y respuesta al cambio.
- Fortalecer la comunicación y la colaboración.
- Comunicación verbal directa entre los implicados en el proyecto.
- Simplicidad; eliminación de artefactos innecesarios en la gestión del proyecto.

2.3.2.2 Roles de SCRUM

SCRUM tiene una estructura muy simple, todas las responsabilidades del proyecto se reparten en tres roles los cuales serán empleados en el proyecto:

- **Product Owner (Dueño del producto):** representa a todos los interesados en el producto final, es el responsable oficial del proyecto, gestión, control y visibilidad de la lista de acumulación o lista de retraso del producto (product backlog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar. [21]

Tiene bajo su responsabilidad:

- ✓ Financiación del proyecto.
- ✓ Requisitos del sistema.
- ✓ Retorno de la inversión del proyecto.
- ✓ Lanzamiento del proyecto.

- **SCRUM master (Líder del proyecto):** responsable del proceso SCRUM, de cumplir la meta, resolver los problemas y asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de SCRUM y que progrese según lo previsto. Interactúa con el cliente y el equipo, coordina los encuentros diarios y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par. [21]

- **Team (Equipo de trabajo):** responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software, es decir, de convertir el *product backlog* en un software entregable. El equipo tiene la autoridad para reorganizarse y definir las acciones necesarias o sugerir eliminación de impedimentos. [21]

Características del equipo de trabajo:

- ✓ Autogestionado.
- ✓ Autoorganizado.
- ✓ Multifuncional.

2.3.2.3 Artefactos de SCRUM

- **Product Backlog (Lista de necesidades del cliente):**

Es el inventario en el que se almacenan todas las funcionalidades o requisitos en forma de lista priorizada, estos requisitos serán los que tendrá el producto o los que irá adquiriendo en sucesivas iteraciones.

La lista será gestionada y creada por el cliente con la ayuda del SCRUM Máster, quien indicará el coste estimado para completar un requisito, y además contendrá todo lo que aporte un valor final al producto. [22]

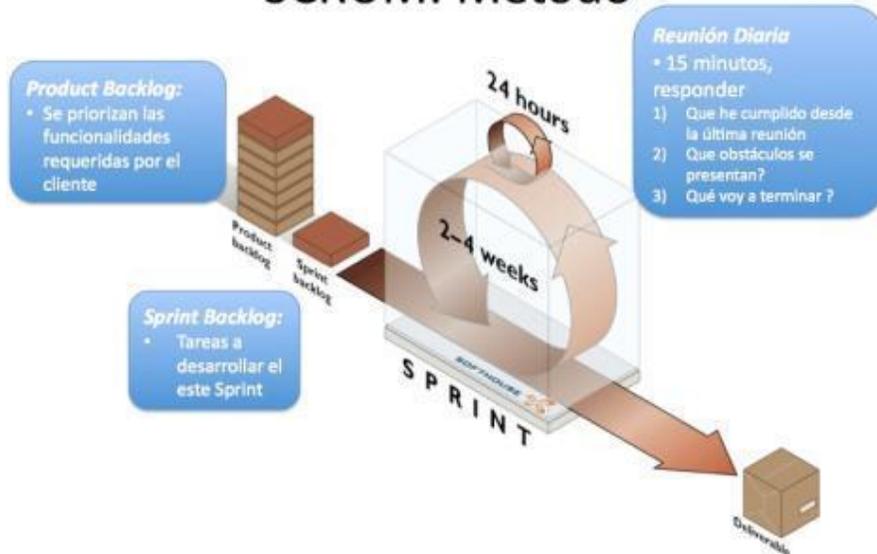
- **Sprint Backlog (Lista de tareas que se realizan en un sprint):**

Es la lista de tareas que elabora el equipo durante la planificación de un sprint, se asignan las tareas a cada persona y el tiempo que queda para terminarlas, de esta manera el proyecto se descompone en unidades más pequeñas y se puede determinar o ver en qué tareas no se está avanzando con el fin de intentar eliminar el problema. [22]

También se define como la lista de requerimientos a implementarse en un sprint. A diferencia del product backlog es estable hasta que finalice el sprint. [22]

Figura 5. Ciclo desarrollo SCRUM

SCRUM: Método



Fuente [22]

Incremento: representa los requisitos que se han completado en una iteración y que son perfectamente operativos. Según los resultados que se obtengan, el cliente puede ir haciendo los cambios necesarios y replanteando el proyecto.

2.3.2.4 Eventos de SCRUM.

SCRUM está conformado por cuatro eventos formales, los cuales están contenidos dentro de cada sprint para poder así asegurar la adaptación e inepción.

Cuenta con un sprint planning meeting (Planificación del Sprint) en él cual, se priorizan requerimientos en el product backlog, un SCRUM diario que permite conocer fallas constantes y avances del desarrollo. También habrá una revisión “sprint review” y una retrospectiva de cada sprint “sprint retrospective”.

Eventos formales:

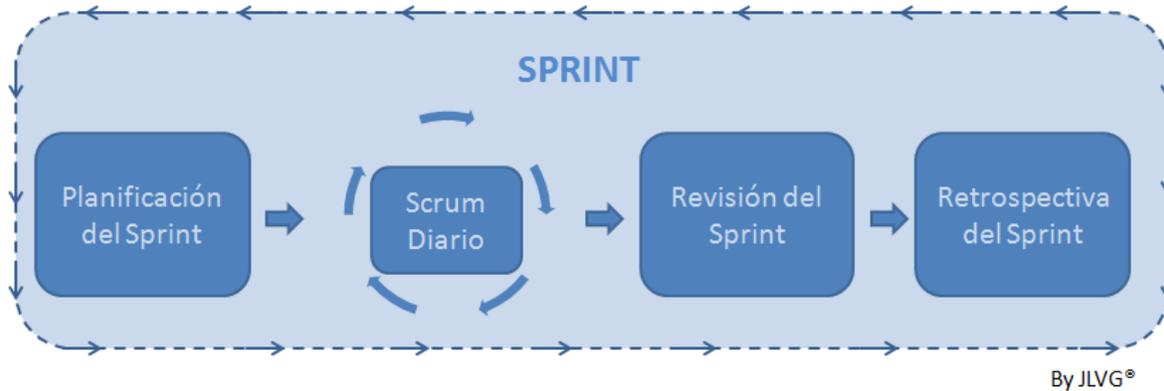
Planificación del sprint (Sprint planning meeting).

SCRUM diario (Daily SCRUM).

Revisión del sprint (Sprint Review).

Retrospectiva del sprint (Sprint Retrospective).

Figura 6. Eventos de SCRUM.



Fuente [23]

- **Planificación del sprint (sprint planning meeting)**

Se define el product backlog, si no ha sido definido, consiste en una lista priorizada de requerimientos que puede ser modificado continuamente. En cada iteración (sprint) el product backlog es revisado por el equipo y se planifica el primer sprint. [24]

Se definen las tareas necesarias para poder completar cada objetivo y se realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea en donde cada miembro del equipo se auto asigna a las tareas que puede realizar.

- **SCRUM diario (Daily SCRUM)**

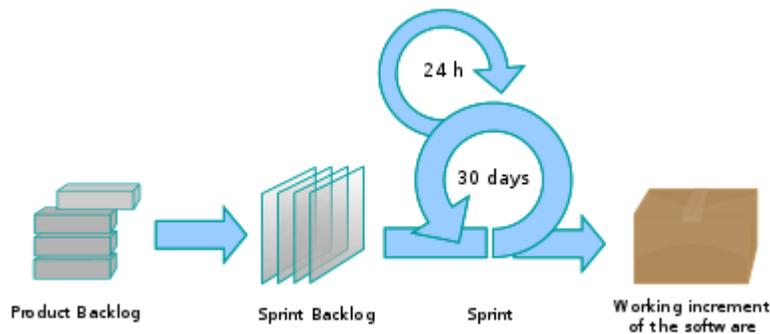
Se hacen reuniones diarias para así poder conocer el avance de cada tarea y estimar el trabajo para la siguiente jornada. En esta reunión se tendrá como referencia el backlog del sprint y el equipo gráfico burn-down con la información de la reunión anterior y además, qué tareas hizo cada persona del equipo, la reunión no podrá consumir más de 15 minutos en la cual se contestará a tres preguntas básicas:

- ¿Qué se ha hecho de nuevo con respecto a la última reunión diaria?
- ¿Qué será lo siguiente a realizar?
- ¿Qué problema hay para realizarlos?

- **Revisión del sprint (Sprint Review)**

Terminado el sprint se realiza análisis y revisión del incremento generado. En esta reunión se presentan resultados finales, sugiriendo presentar un prototipo. [24] Los desarrolladores presentan el producto entregable que han implementado y, los gestores, clientes, usuarios y product owner analizan esa entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso. [22] Esta reunión servirá para tomar decisiones que ayuden a escoger un adecuado camino para poder cumplir y alcanzar las metas.

Figura 7. Ciclo del sprint – Daily SCRUM



Fuente [24]

- **Retrospectiva del sprint (Sprint retrospective)**

Reunión que se celebra al final del sprint y en la que el equipo presenta las historias conseguidas mediante una demostración del producto. Posteriormente, el equipo analiza qué se hizo bien, qué procesos serían mejorables y discute acerca de cómo perfeccionarlos [25]

Esta reunión generalmente es realizada por el SCRUM master y dura máximo 3 horas. Se contestan las siguientes preguntas:

- ¿Qué ha ido bien durante el último sprint?
- ¿Qué será mejorado para el siguiente sprint?

- **Agile Inception**

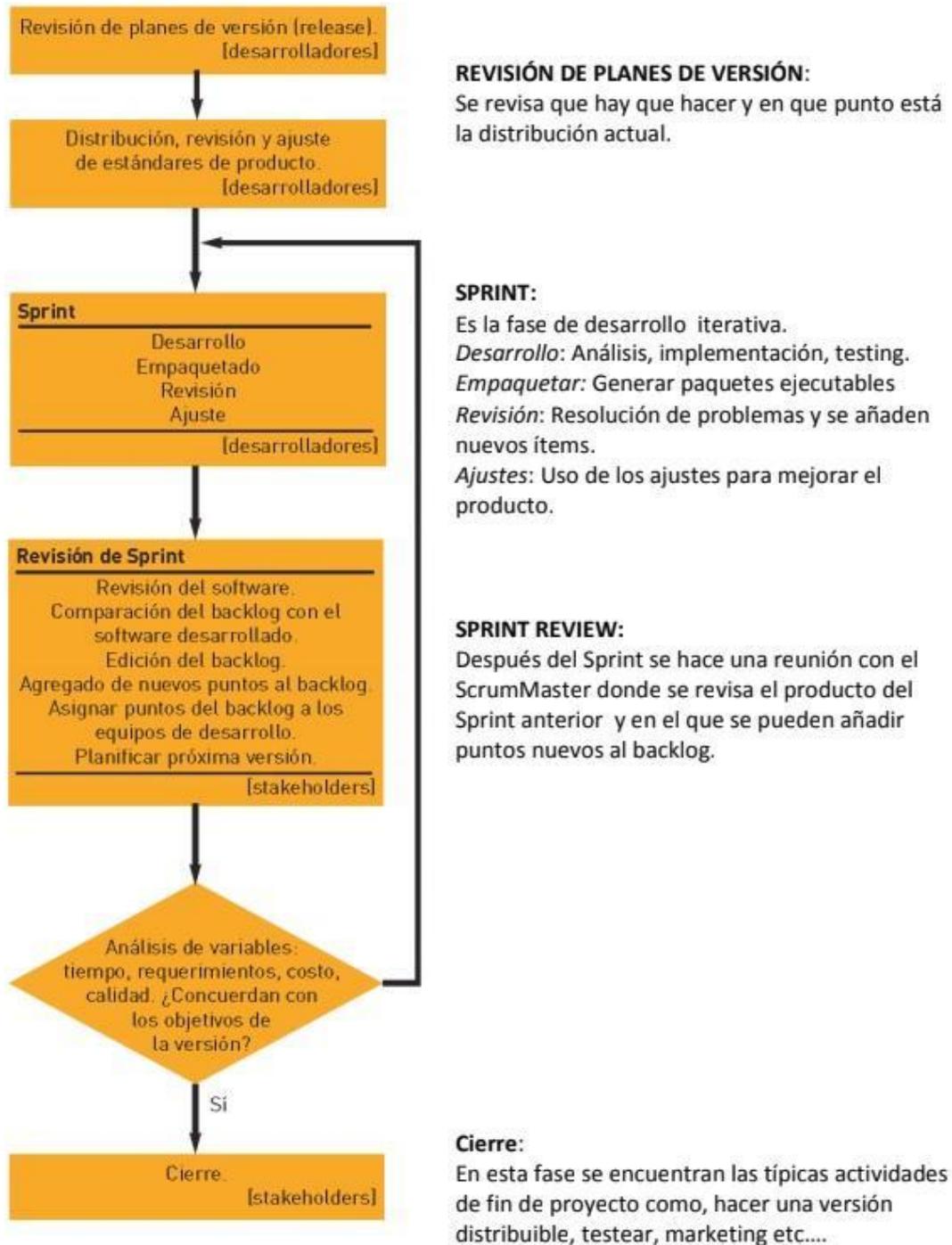
Para poder guiar a todos los miembros del equipo hacia un mismo objetivo, se implementó un conjunto de dinámicas orientadas para este trabajo.

Esencialmente se trata de técnicas de conceptualización que deberían emplearse en el proceso de iniciación de un proyecto para aumentar la probabilidad de éxito del producto resultante. Estas técnicas se pueden usar igualmente para calificar estrategia (misión, visión y propuesta de valor) del proyecto de empresa de cualquier compañía o bien para reorientar proyectos de cualquier índole. [26]

Para poder realizar un inception se deben dar respuesta a las siguientes preguntas:

- ✓ ¿Por qué estamos aquí? Hacer una frase que resuma el propósito del producto.
- ✓ Elevator Pitch. definir el proyecto en un tiempo determinado.
- ✓ Crea tu Not List. describir lo que se hará, lo que se no se hará, y lo que se puede discutir para luego hacerlo.
- ✓ ¿Qué te quita el sueño? Describir los miedos y cómo podemos minimizarlos o evitarlos.

Figura 8. Diagrama flujo SCRUM.

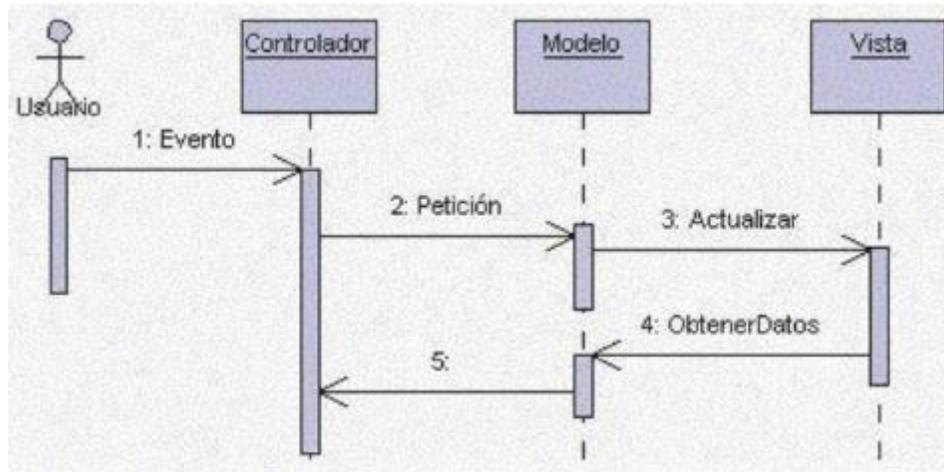


Fuente [22]

2.3.3 Patrón de diseño (MVC)

El modelo vista controlador es el encargado en todo proyecto software de separar la parte lógica y la parte visual, garantizando así una estructura más ordenada, fácil de diferenciar, además permite la reutilización de código.

Figura 9. Diagrama de secuencia Modelo vista controlador.



Fuente [27]

2.3.3.1 Modelo

El modelo es el objeto que representa los datos del programa, maneja los datos y controla todas las transformaciones. No tiene conocimiento específico de los controladores o de las vistas, contiene referencia a ellos. [27]

Tiene bajo su responsabilidad:

- ✓ Definir las reglas de negocio (la funcionalidad del sistema).
- ✓ Acceder a la capa de almacenamiento de datos.
- ✓ Notificar a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo.

2.3.3.2 La vista

La vista es el objeto que maneja la presentación visual de los datos representados por el modelo, genera una representación visual de este y muestra los datos al usuario. Interactúa preferentemente con el controlador, pero es posible que trate directamente con el modelo a través de una referencia al propio modelo. [27]

Tiene bajo su responsabilidad:

- ✓ Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- ✓ Tienen un registro de su controlador asociado.
- ✓ Pueden dar el servicio de “actualización”, para que sea invocado por el controlador o por el modelo cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes.

2.3.3.3 Controlador

El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo, centra toda la interacción entre la vista y el modelo. Cuando se realiza algún cambio, entra en acción, ya sea por cambios en la información del modelo o por alteraciones de la vista. [27]

Tienen bajo su responsabilidad:

- ✓ Recibir los eventos de entrada.
- ✓ Encargado de realizar conexión con la parte visual.

2.4 HERRAMIENTAS DE APOYO

2.4.1 BITBUCKET

Bitbucket es un servicio de alojamiento basado en web para los proyectos que utilizan el sistema de control de versiones de Mercurial y Git. Es un servicio gratuito que brinda además planes comerciales. En este proyecto se hizo uso de Git.

Figura 10. Bitbucket.



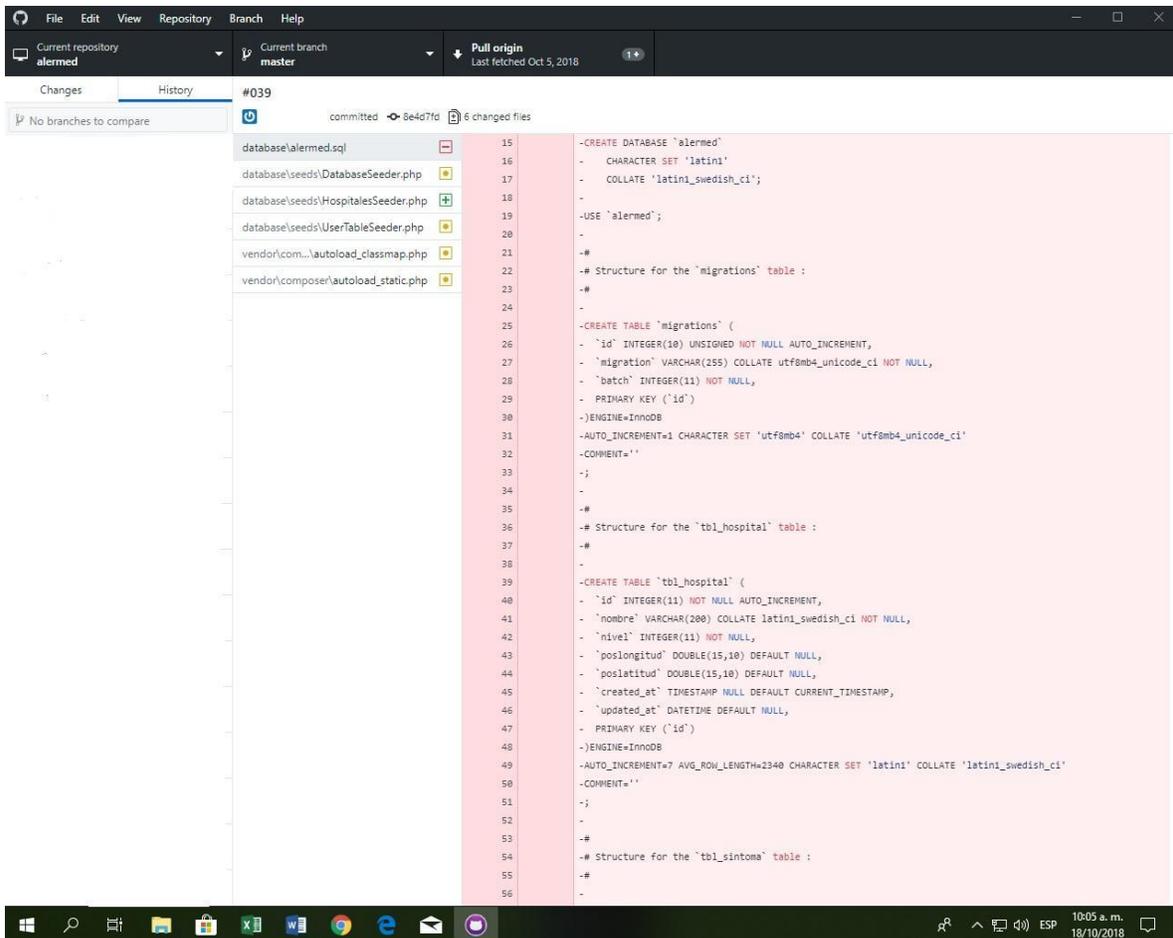
Fuente [28]

Bitbucket es más que solo gestión de código Git. Bitbucket ofrece a los equipos un sitio para planificar proyectos, colaborar en el código, probar y desplegar. [28]

2.4.2 GITHUB

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos los cuales utilizan el sistema de control de versiones Git, brinda la posibilidad de tener una programación en parejas con un control de versiones y códigos en tiempo real para el equipo de desarrollo.

Figura 11. Entorno GitHub.



The screenshot displays the GitHub web interface for a pull request #039. The top navigation bar includes 'File', 'Edit', 'View', 'Repository', 'Branch', and 'Help'. The current repository is 'alarmed', and the current branch is 'master'. The pull request is titled '#039' and shows it was committed by '8e4d7fd' with 6 changed files. The diff view shows changes to several files, including database seeders and migration files. The diff content is as follows:

```
15 database\alarmed.sql
16
17 database\seeds\DatabaseSeeder.php
18 database\seeds\HospitalesSeeder.php
19 database\seeds\UserTableSeeder.php
20
21 vendor\composer\autoload_classmap.php
22 vendor\composer\autoload_static.php
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
```

```
-CREATE DATABASE `alarmed`
- CHARACTER SET `latin1`
- COLLATE `latin1_swedish_ci`;
-
-USE `alarmed`;
-
-# Structure for the `migrations` table :
-#
-
--CREATE TABLE `migrations` (
- `id` INTEGER(10) UNSIGNED NOT NULL AUTO_INCREMENT,
- `migration` VARCHAR(255) COLLATE utf8mb4_unicode_ci NOT NULL,
- `batch` INTEGER(11) NOT NULL,
- PRIMARY KEY (`id`)
-)ENGINE=InnoDB
-AUTO_INCREMENT=1 CHARACTER SET `utf8mb4` COLLATE `utf8mb4_unicode_ci`
-COMMENT=''
-;
-
-# Structure for the `tbl_hospital` table :
-#
-
--CREATE TABLE `tbl_hospital` (
- `id` INTEGER(11) NOT NULL AUTO_INCREMENT,
- `nombre` VARCHAR(200) COLLATE latin1_swedish_ci NOT NULL,
- `nivel` INTEGER(11) NOT NULL,
- `posLongitud` DOUBLE(15,10) DEFAULT NULL,
- `poslatitud` DOUBLE(15,10) DEFAULT NULL,
- `created_at` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
- `updated_at` DATETIME DEFAULT NULL,
- PRIMARY KEY (`id`)
-)ENGINE=InnoDB
-AUTO_INCREMENT=7 AVG_ROW_LENGTH=2340 CHARACTER SET `latin1` COLLATE `latin1_swedish_ci`
-COMMENT=''
-;
-
-# Structure for the `tbl_sintoma` table :
-#
```

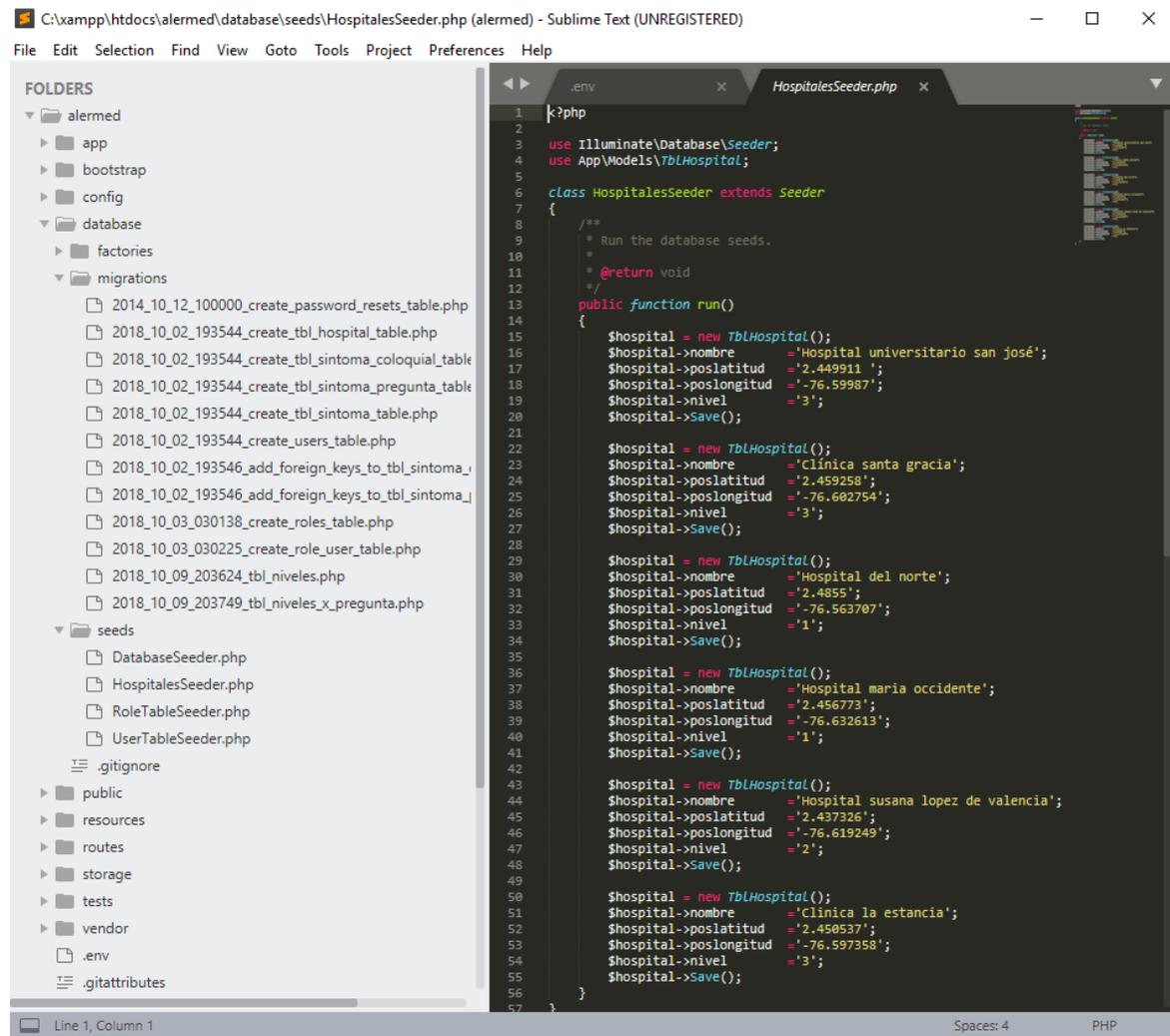
Fuente [Propia]

GitHub es una plataforma de desarrollo inspirada en la forma de trabajo de la comunidad. Desde el código abierto hasta los negocios, puede alojar y revisar códigos, administrar proyectos y crear software junto con 31 millones de desarrolladores. [29]

2.4.3 Sublime text

Sublime text es un editor de código multiplataforma y ligero. Es una herramienta concebida para programar sin distracciones, sublime, soporta un gran número de lenguajes entre ellos (C, C++, PHP, PERL...ENTRE OTROS).

Figura 12. Entorno sublime text.



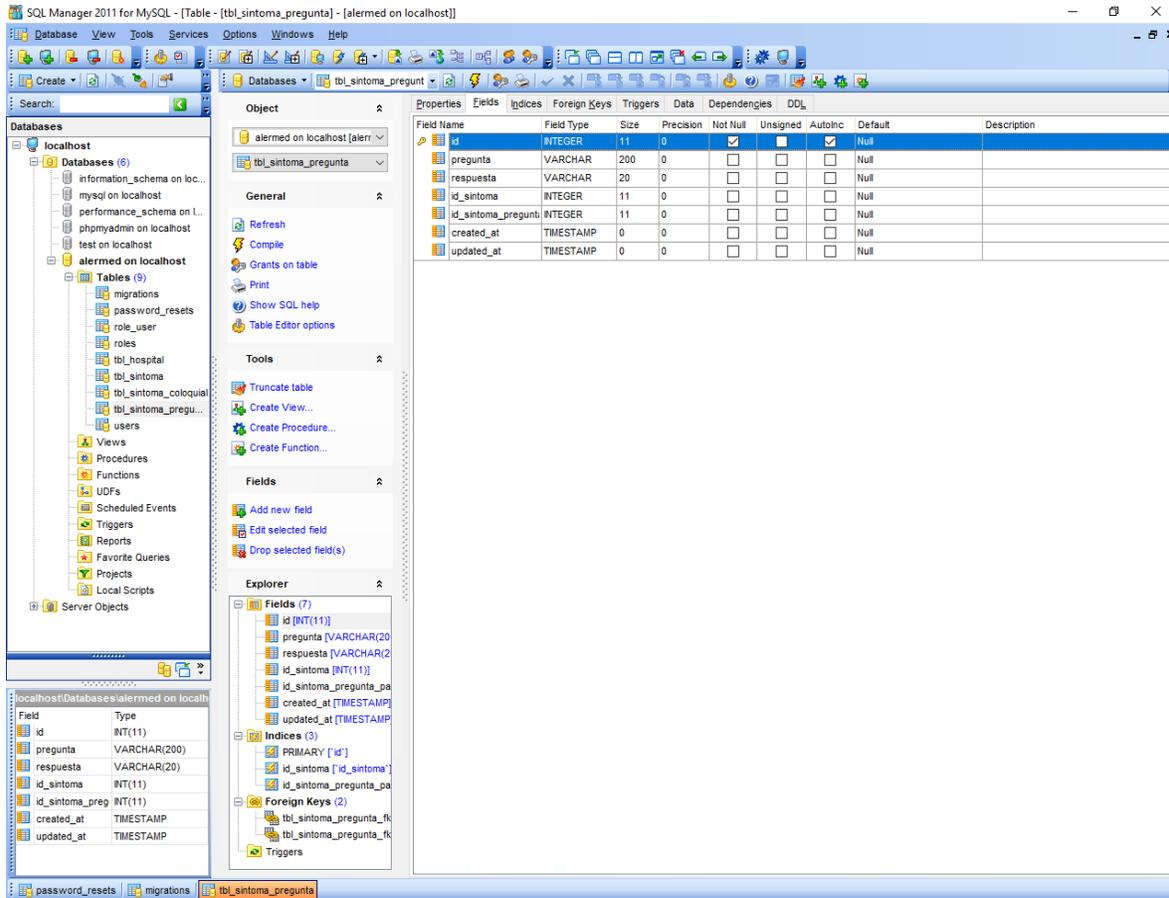
Fuente [Propia]

Este editor se puede descargar en versión de prueba, plenamente funcional y sin limitación de tiempo y está disponible para OS X, LINUX Y WINDOWS.

2.4.4 Ems SQL Manager

Es una poderosa herramienta para administrar y desarrollar bases de datos de servidor MySQL. El administrador SQL para MySQL es adecuado para todas las versiones MySQL. Esta herramienta le permite crear/editar todos los objetos de las bases de datos MySQL, diseñar gráficamente bases de datos, iniciar scripts SQL e importar y exportar datos de bases de datos. [30]

Figura 13. Ambiente SQL manager for MySQL.



Fuente [Propia]

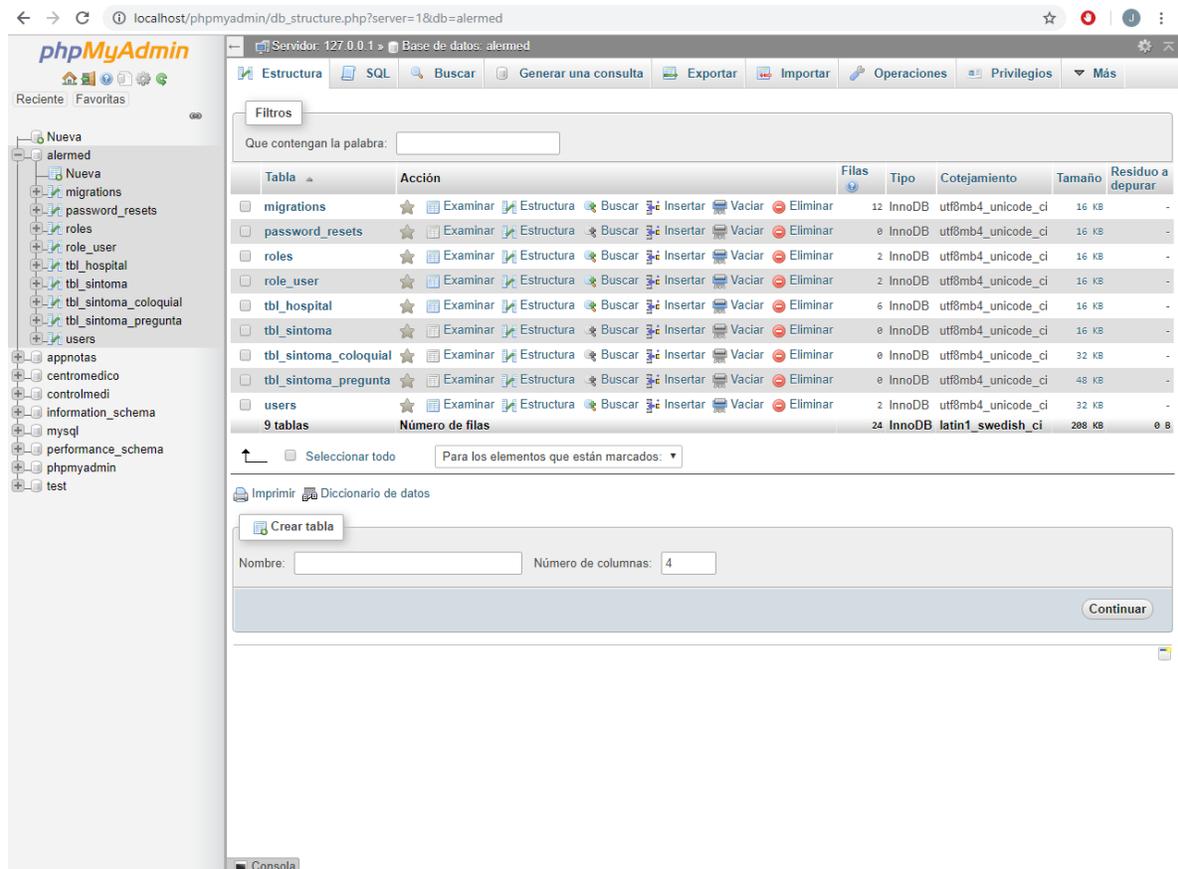
Características:

- ✓ Eficiente gestión de bases de datos y navegación.
- ✓ Fácil administración de todos los objetos MySQL.
- ✓ Excelente gestión de seguridad.
- ✓ Potente diseñador de bases de datos visual.
- ✓ Conexión al servidor MySQL vía HTTP.
- ✓ Conexión al servidor MySQL vía SSH.

2.4.5 PHPMYADMIN

Es una herramienta software libre escrita en PHP, destinada a manejar la administración de MYSQL a través de la web. Es compatible con una amplia gama de operaciones en MYSQL y mariaDB, las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras que usted todavía tiene la capacidad de ejecutar directamente cualquier declaración SQL. [31]

Figura 14. Entorno PHPMyAdmin.



Fuente [Propia]

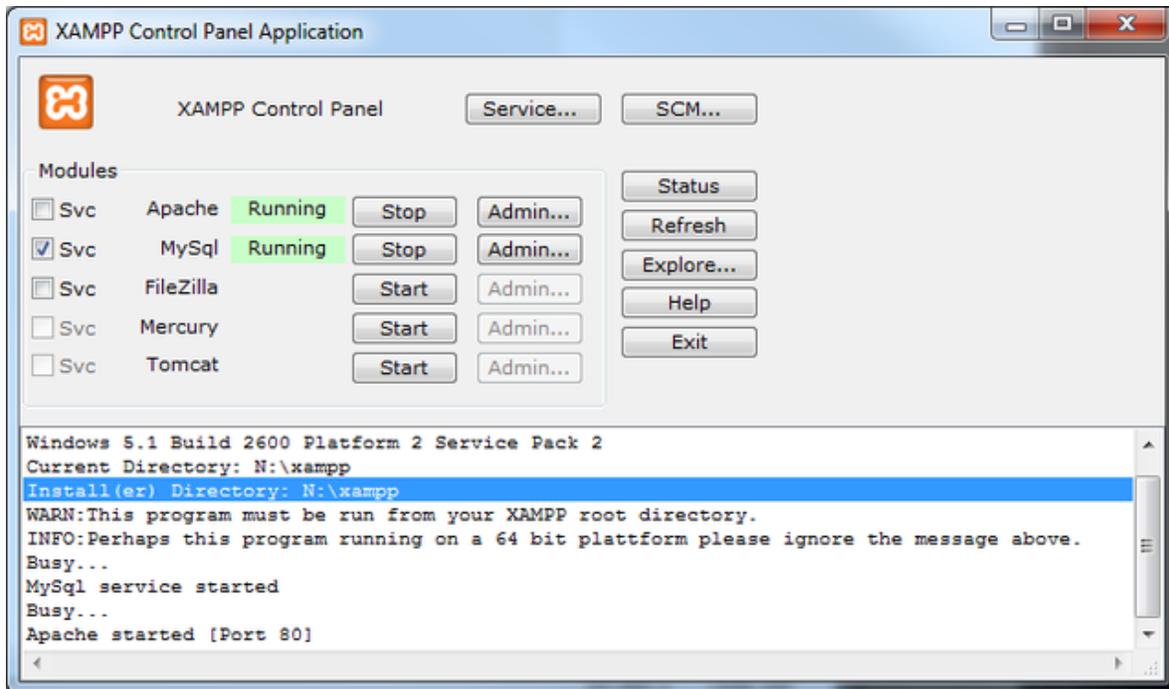
Características:

- ✓ Interfaz web intuitiva.
- ✓ Administrar múltiples servidores.
- ✓ Portar datos de CSV y SQL.
- ✓ Soporte para la mayoría de las características de MySQL.

2.4.6 XAMPP

Es un servidor independiente en base a software libre, el cual dispone de un servidor propio o simplemente permite su uso para hacer pruebas de páginas web, bases de datos, para desarrollar aplicaciones en PHP, con conexión a bases de datos SQL.

Figura 15. XAMPP.



Fuente [Propia]

Xampp es una distribución de apache completamente gratuita y fácil de instalar contiene mariaDB PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar. [32]

2.4.7 CMDER

Cmder es un programa portable para todas las versiones de Windows que se ofrece en dos variantes: una versión mínima que funciona como la terminal de Windows que ya conocemos y una completa que, además, incluye compatibilidad con comandos UNIX, de forma que todos aquellos que tengan a la mano la terminal de Linux puedan usar la de Windows casi de la misma forma. [33]

Figura 16. Entorno Cmder.

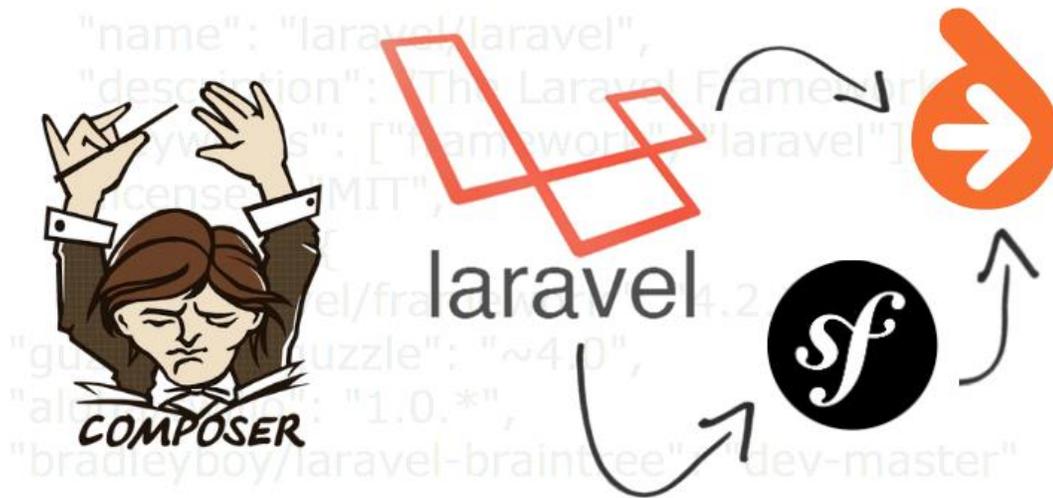


Fuente [Propia]

2.4.8 COMPOSER

Composer es un manejador de dependencias, no un gestor de paquetes, que trata con paquetes y librerías. La instalación siempre es local para cualquier proyecto, las librerías se instalan en un directorio por defecto, composer es capaz de instalar las librerías que requiere el proyecto con las versiones que se necesiten. Si las librerías dependen de otras, descarga todo lo necesario para que sea funcional. [34]

Figura 17. Composer.



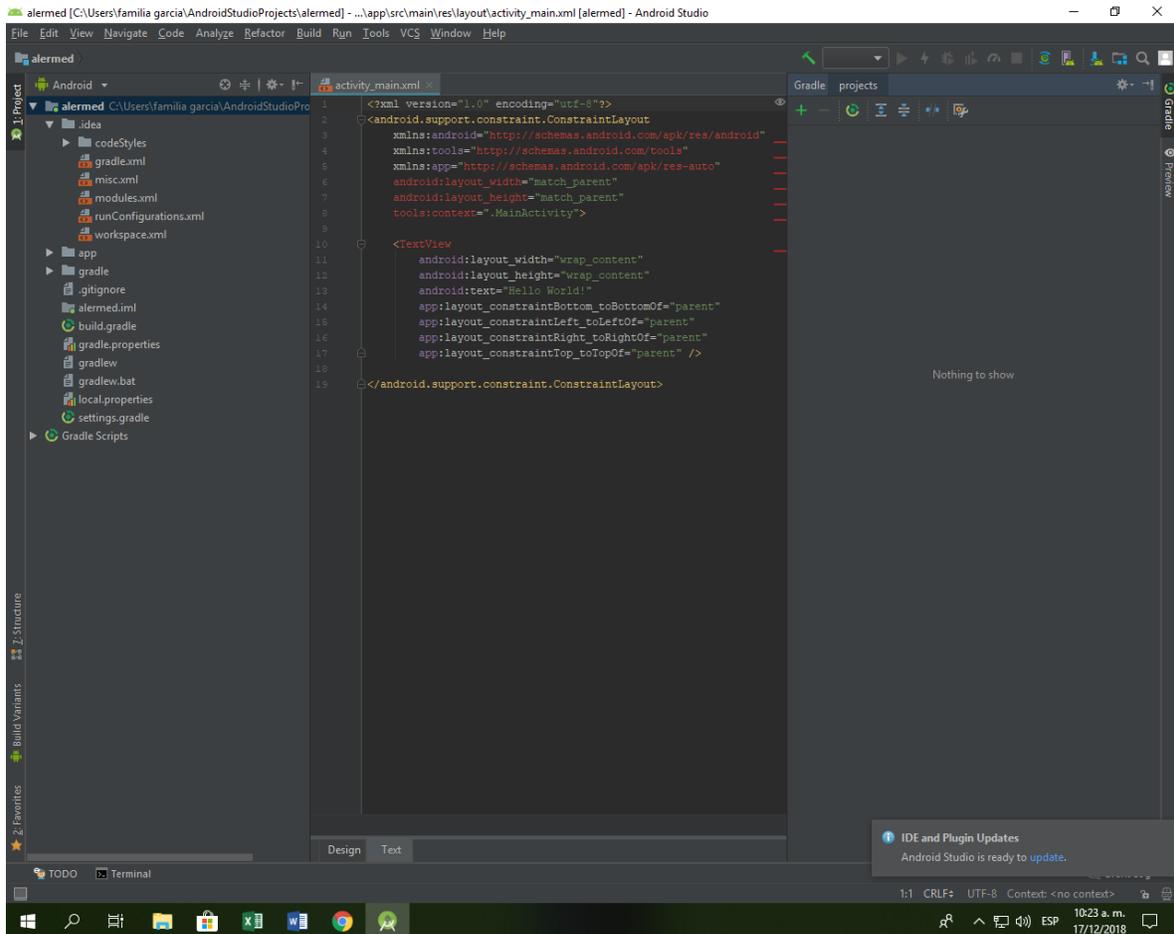
Fuente [34]

Composer es un manejador de dependencias para la compatibilidad de librerías en laravel.

2.4.9 ANDROID STUDIO

Android studio es un entorno de desarrollo para la plataforma Android. Posee una licencia gratuita a través de la licencia apache 2.0 y está disponible para cualquier sistema operativo.

Figura 18. Entorno Android Studio.



Fuente [Propia]

2.5.0 SCRUM PÓKER / PLANNING POKER

Esta técnica es la que permite al inicio del proyecto hacer una estimación fiable de tiempos en cada sprint. En este proceso participan todos los miembros del proyecto para dar a conocer ideas y aprender de los otros compañeros de equipo, finalizado el planning póker dará como resultado que junto a cada historia de usuario debe aparecer un número que indique el esfuerzo para terminar esa tarea. [35]

Figura 19. Planning póker.



Fuente [35]

2.6 Glosario

- **Inception:** técnicas que se pueden usar para calificar estrategia (misión, visión y propuesta de valor) del proyecto de empresa de cualquier compañía o bien para reorientar proyectos de cualquier índole. [26].
- **Elevator Pitch:** definir el proyecto en un tiempo determinado.
- **Not List:** describir lo que se hará, lo que se no se hará, y lo que se puede discutir para luego hacerlo.
- **Simplicidad:** Realizar diseños sencillos y amigables con el usuario, disminuyendo tiempo y costos.
- **Retroalimentación:** es un proceso por el que una cierta proporción de la señal de salida de un sistema se redirige de nuevo a la entrada. Esto es frecuentemente en el control del comportamiento dinámico del sistema.
- **XP:** “Extreme Programming” / Programación extrema
- **Planificación:** Es una fase de la metodología ágil extreme programming, en la cual se determina el objetivo del proyecto.
- **Iteración:** Es un conjunto de periodos de tiempo dentro de un proyecto.
- **Codificación:** Es una fase de la metodología ágil extreme programming, Se divide las funcionalidades del sistema, para facilitar el desarrollo.
- **Scrum:** Es un marco de trabajo, en el que se aplican las buenas prácticas, las cuales se pueden trabajar en equipo, para así obtener mejores resultados.
- **Framework:** En general, el término framework, hace referencia a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.
- **Product Owner:** Es aquel que se encarga de priorizar mediante el product backlog, lo que se realizara en los sprints.
- **Development Team:** Son los encargados de desarrollar e implementar el software, y son auto-organizados.
- **Scrum Master:** Es el encargado de apoyar al development team y al product owner, durante el desarrollo del proyecto.
- **Daily:** Son reuniones que se hacen a diario para, ver el progreso de los sprints.
- **CRUD:** Create, Read, Update y Delete. (Crear, Leer, Actualizar y borrar).

- **Not List:** describir lo que se hará, lo que se no se hará, y lo que se puede discutir para luego hacerlo.
- **EPS:** Entidad promotora de salud.
- **IPS:** Institución prestadora de servicios.
- **CIE-10:** Clasificación internacional de enfermedades 10 edición.
- **Scripting:** Archivo de procesamiento por lotes. Es un documento que contiene instrucciones, escritas en códigos de programación.
- **Morbilidad:** Cantidad de personas que enferman en un lugar y un período de tiempo determinados en relación con el total de la población.

3. METODOLOGÍA

Para la elaboración de este proyecto se optó por emplear el marco de trabajo SCRUM ya que permite tener un mayor control sobre el proyecto, una visión más clara de principio a fin con una buena documentación y entregas parciales en cada uno de los sprints del proyecto. Gracias a los pilares que nos brinda SCRUM (transparencia, inspección y adaptación) y su enfoque (iterativo, incremental) garantiza poder realizar cambios, revisar fallas y corregirlas si es necesario.

También se decide hacer uso de la metodología XP (Extreme Programming) que gracias a sus valores (simplicidad, comunicación, coraje, respeto y retroalimentación) y su énfasis de tener un equipo de trabajo y una programación en parejas, brinda la posibilidad de elaborar código y ser revisado al tiempo. Igualmente permite a los programadores estar en un ambiente adaptable a cambios.

3.1 IMPLEMENTACIÓN DE SCRUM

Basados en la metodología SCRUM, en el sprint planning se concluyeron una serie de sprints para la elaboración del proyecto

3.1.1 PLANIFICACIÓN SPRINT 0

En el primer paso se estableció los roles SCRUM TEAM, artefactos, actividades y las reuniones por sprint, para así poder guiar a todos los miembros del equipo hacia un mismo objetivo.

ROLES SCRUM

Tabla 2. SCRUM master

NOMBRE	Ing. Julio Andrés Mosquera
ROL	SCRUM master
RESPONSABILIDAD	facilitar acompañamiento del proyecto

Tabla 3. Development team

NOMBRE	Julián David García
ROL	Investigador y desarrollador
RESPONSABILIDAD	Investigar y desarrollar el producto

Tabla 4. Development team

NOMBRE	Wilson Alejandro Restrepo
ROL	Investigador y desarrollador
RESPONSABILIDAD	Investigar y desarrollar el producto

Tabla 5. Producto owner

NOMBRE	Julio Mosquera
ROL	Representante de la ciudadanía
RESPONSABILIDAD	Representa las necesidades de la comunidad

Se perpetró una lista priorizada de actividades a implementar con su debido orden (product backlog), para así poder desplegar lo primordial en el desarrollo.

3.1.2 PLANIFICACIÓN SPRINT 1

Está conformado por las siguientes historias de usuario, las cuales se estimaron para desarrollar en un tiempo aproximado de 20 días hábiles con 7 horas de trabajo individual.

- Autenticar usuario.
- Registrar Usuario.
- Actualizar Usuario.
- Eliminar Usuario.
- Recuperar Contraseña.

3.1.3 PLANIFICACIÓN SPRINT 2

Está conformado por las siguientes historias de usuario, las cuales se estimaron para desarrollar en un tiempo aproximado de 30 días hábiles con 7 horas de trabajo individual.

- Registrar Síntoma.
- Actualizar Síntoma.
- Eliminar Síntoma.
- Generar nivel de urgencia.

3.1.4 PLANIFICACIÓN DEL SPRINT 3

Está conformado por las siguientes historias de usuario, las cuales se estimaron para desarrollar en un tiempo estimado de 20 días hábiles con 8 horas de trabajo individual.

- Consultar Síntoma
- Registrar Hospital
- Actualizar Hospital
- Eliminar Hospital
- Consultar información de Hospital
- Registrar calificación de Hospital

3.1.5 PLANIFICACIÓN DAILY SCRUM.

Durante el desarrollo, se realizaron reuniones constantes en el mismo lugar con una duración estimada de 1 hora, con el fin de dar a conocer los avances trabajados por cada miembro del equipo y los problemas encontrados en el transcurso del desarrollo de la actividad. En cada una de las reuniones se hizo retroalimentación de requerimientos en cada sprint, durante las primeras dos semanas de trabajo, esta reunión se hace con el SCRUM master para posteriormente delegar actividades directamente al equipo de desarrollo y el SCRUM master participa de alguna de ellas.

3.1.6 SPRINT REVIEW

El equipo de desarrollo con el SCRUM master, comparan los resultados con lo esperado del producto (incremento).

3.1.8 RETROSPECTIVA

Entre el equipo de desarrollo y el SCRUM master para poder cumplir los objetivos planteados en cada sprint y en el proyecto, se apoyó sobre la retrospectiva ágil “bote de velocidad”, la cual se basa en 4 segmentos a evaluar en el desarrollo (fortalezas, debilidades, amenazas y objetivos), parecido al análisis FODA.

- Qué necesitamos para que el viento sople y sigamos moviéndonos. (Fortalezas).
- Qué necesitamos o anclas tenemos, que nos hacen detenernos. (Debilidades).
- Qué bloqueantes potenciales podemos tener o no depende de nuestro equipo interno. (Amenazas).
- A dónde queremos llegar. (Objetivos). [36]

3.2 XP (XTREME PROGRAMMING)

De la mano con la programación extrema y haciendo uso de sus herramientas. Se pudo dar orden desde el inicio del proyecto hasta el final de este. Teniendo retroalimentación continúa entre el equipo de trabajo y el cliente.

XP, gracias a sus valores y principios apoyó en el desarrollo para que la comunicación entre todos fuera fluida y lo más importante que el equipo de desarrolladores tuviera el coraje para enfrentar los cambios.

Se hace uso de las cuatro fases de desarrollo en XP “Planeación, Diseño, Codificación y pruebas”

3.2.1 PLANIFICACIÓN

Con el fin de tener conocimiento de cuáles son las enfermedades más comunes o recurrentes en los servicios de urgencias se hizo un trabajo de investigación en cada uno de los centros de urgencias de la ciudad de Popayán, en la cual se solicitó información referente a los motivos de consulta de los pacientes de los últimos 4 años. Se obtuvo información de dos de los centros que prestan este servicio los cuales fueron el Hospital San José y el Hospital Toribio Maya de nivel 3 y nivel 1 respectivamente; Además de ello se obtiene la morbilidad del municipio de Popayán en los últimos 4 años por medio de la secretaria departamental de salud del Cauca [37], donde se puede evidenciar los diagnósticos más tratados en dichos servicios y una tendencia en algunos de ellos como son las enfermedades intestinales y los dolores de cabeza, también evidenciamos que los motivos de consulta en su mayoría son servicios que se ofrecen en planes de PYP (promoción y prevención) y también las consultas por dolores dentales, los cuales son atendidos en su mayoría en consultorios odontológicos particulares y en los niveles de menor complejidad, también podemos encontrar la diferencia en cuanto a complejidad en atención entre los dos centros hospitalarios de los cuales se obtuvieron los datos a pesar de que este estudio no sea obligatorio para el desarrollo de la plataforma, se realiza esto con el fin de dar relevancia frente a otras enfermedades que no son comunes en el municipio; en las siguientes graficas se presenta un resumen de la morbilidad por año con los CIE10 más tratados en el municipio en y en dos centros hospitalarios.

Gráfica 1. Morbilidad Municipio De Popayán 2015



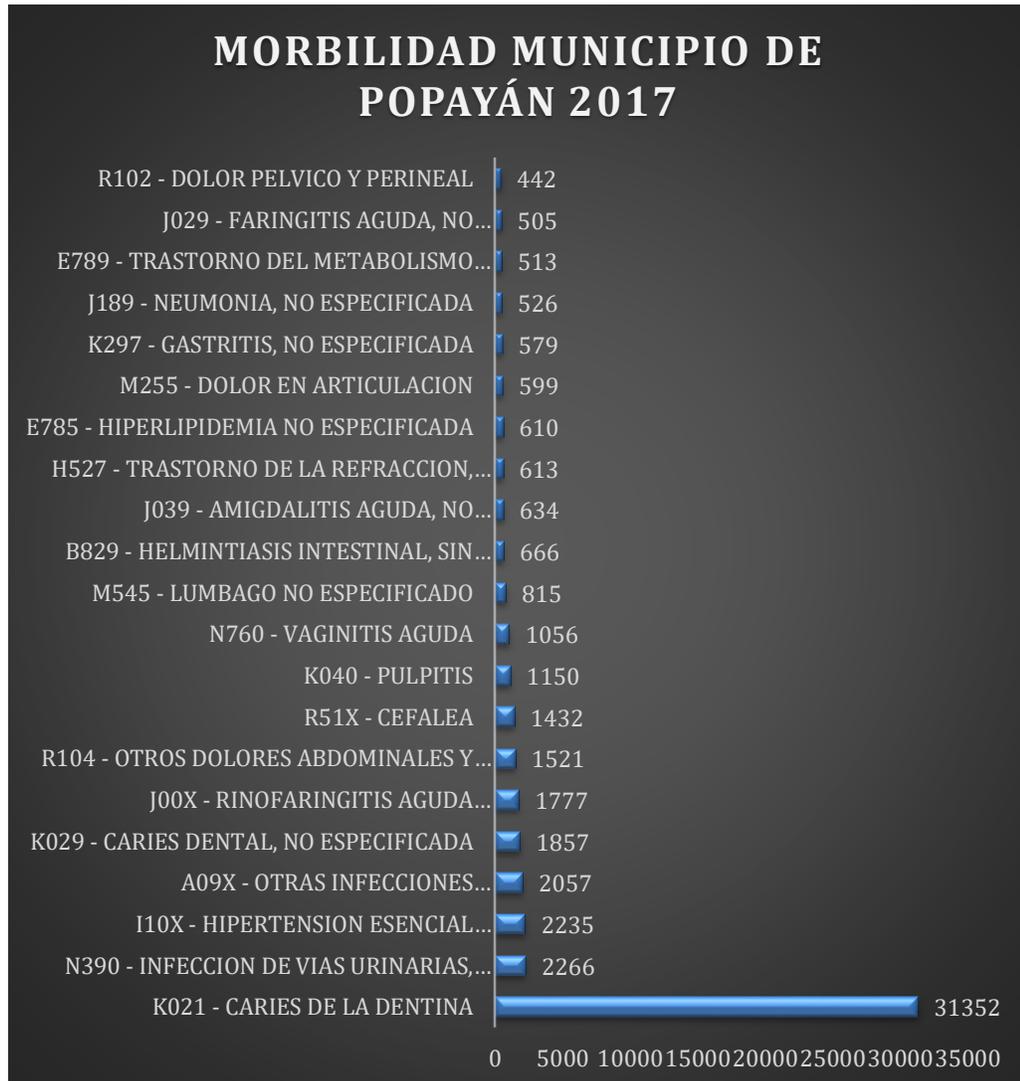
Fuente [Propia]

Gráfica 2. Morbilidad Municipio De Popayán 2016.



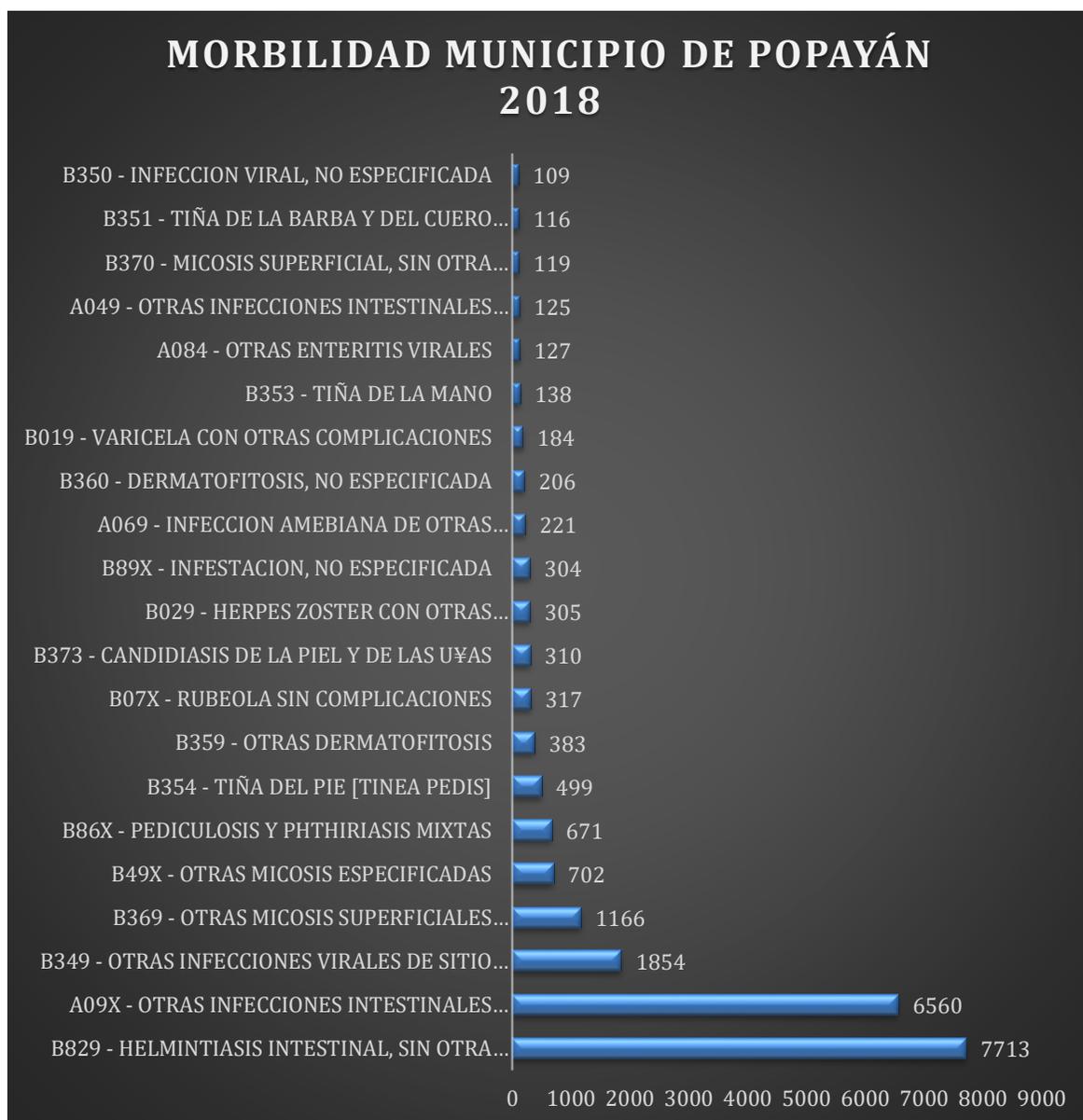
Fuente [Propia]

Gráfica 3. Morbilidad Municipio De Popayán 2017.



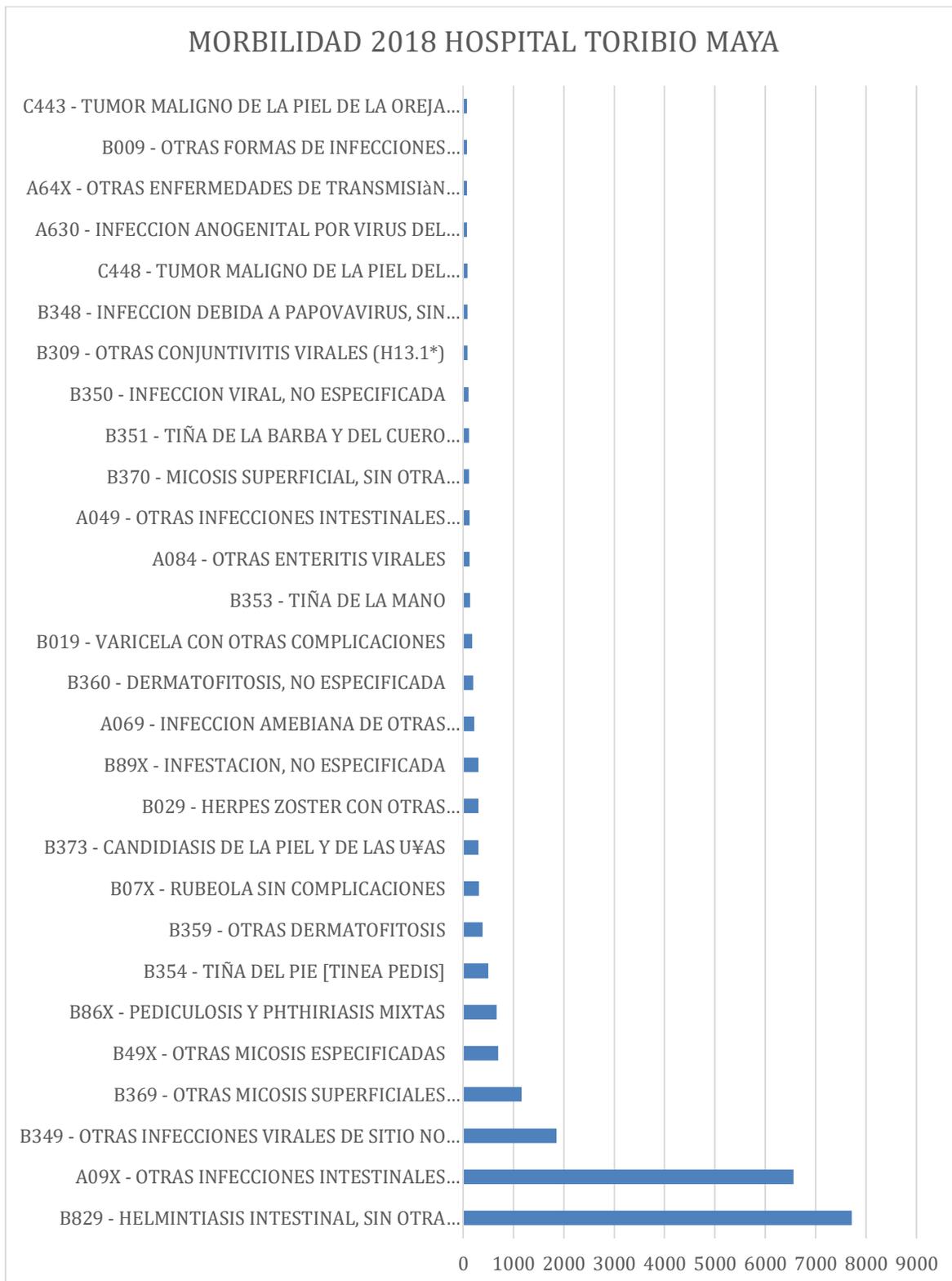
Fuente [Propia]

Gráfica 4. Morbilidad Municipio De Popayán 2018.



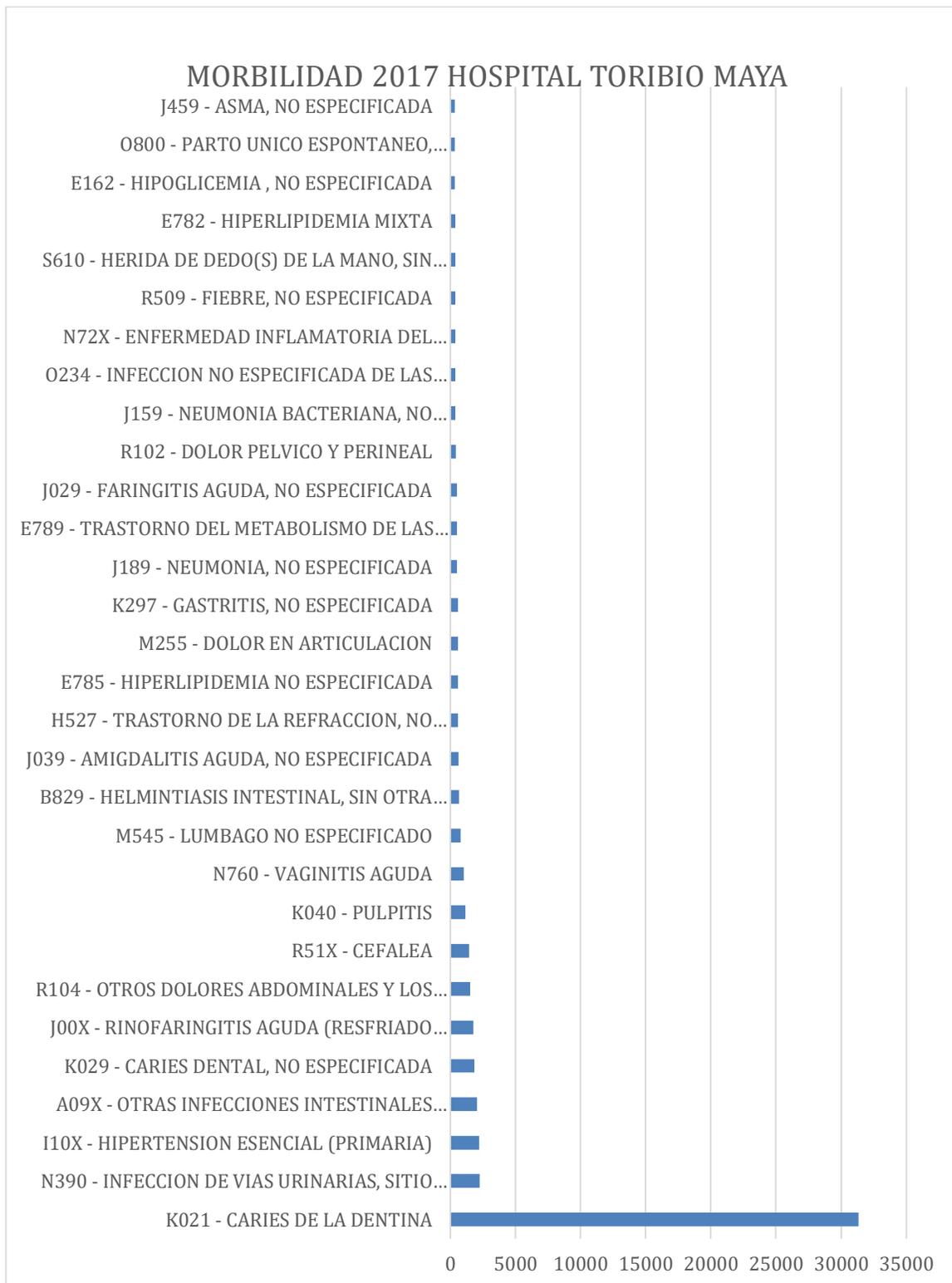
Fuente [Propia]

Gráfica 5. Morbilidad año 2018 hospital Toribio maya



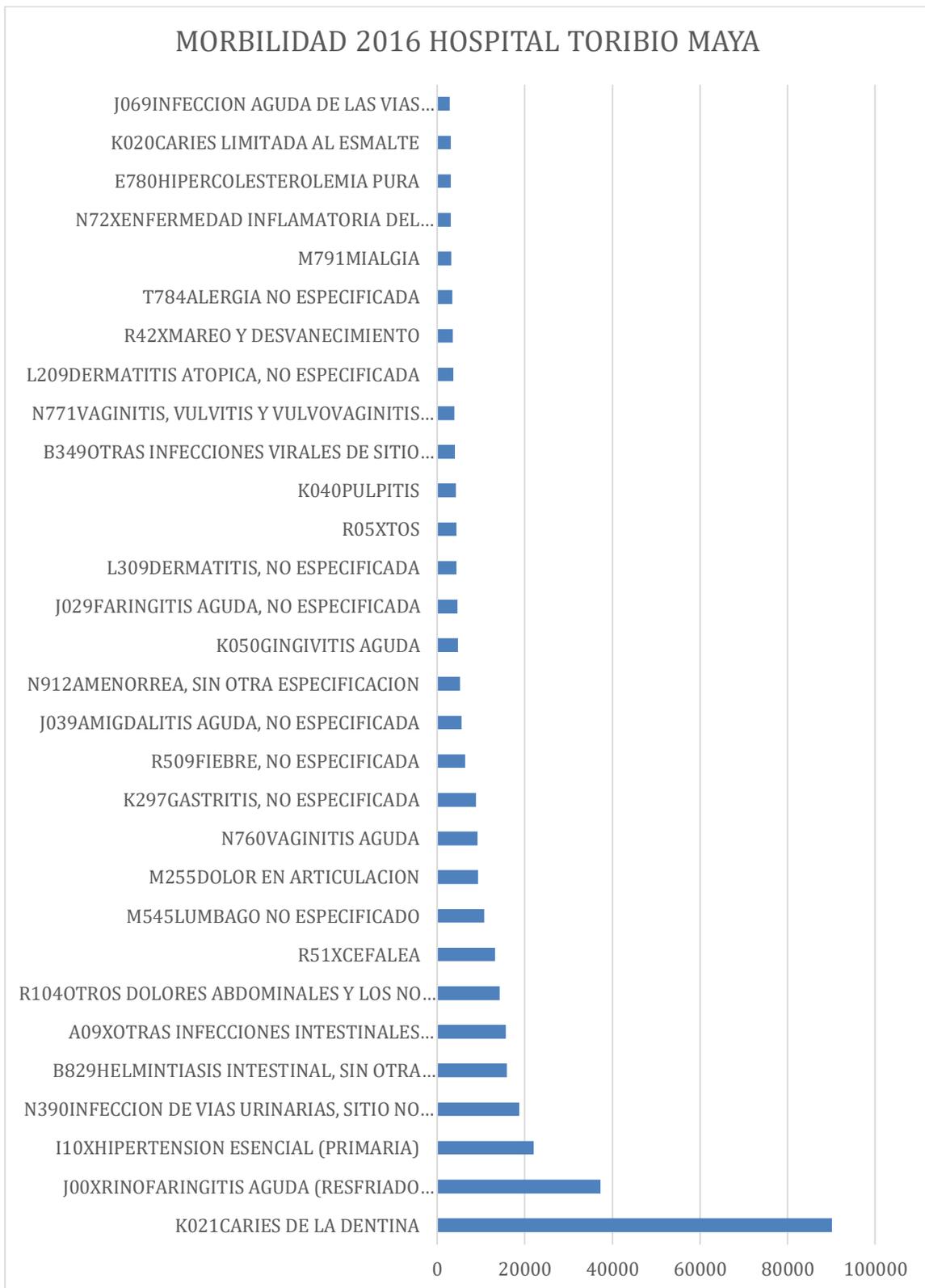
Fuente [Propia]

Gráfica 6. Morbilidad año 2017 hospital Toribio maya



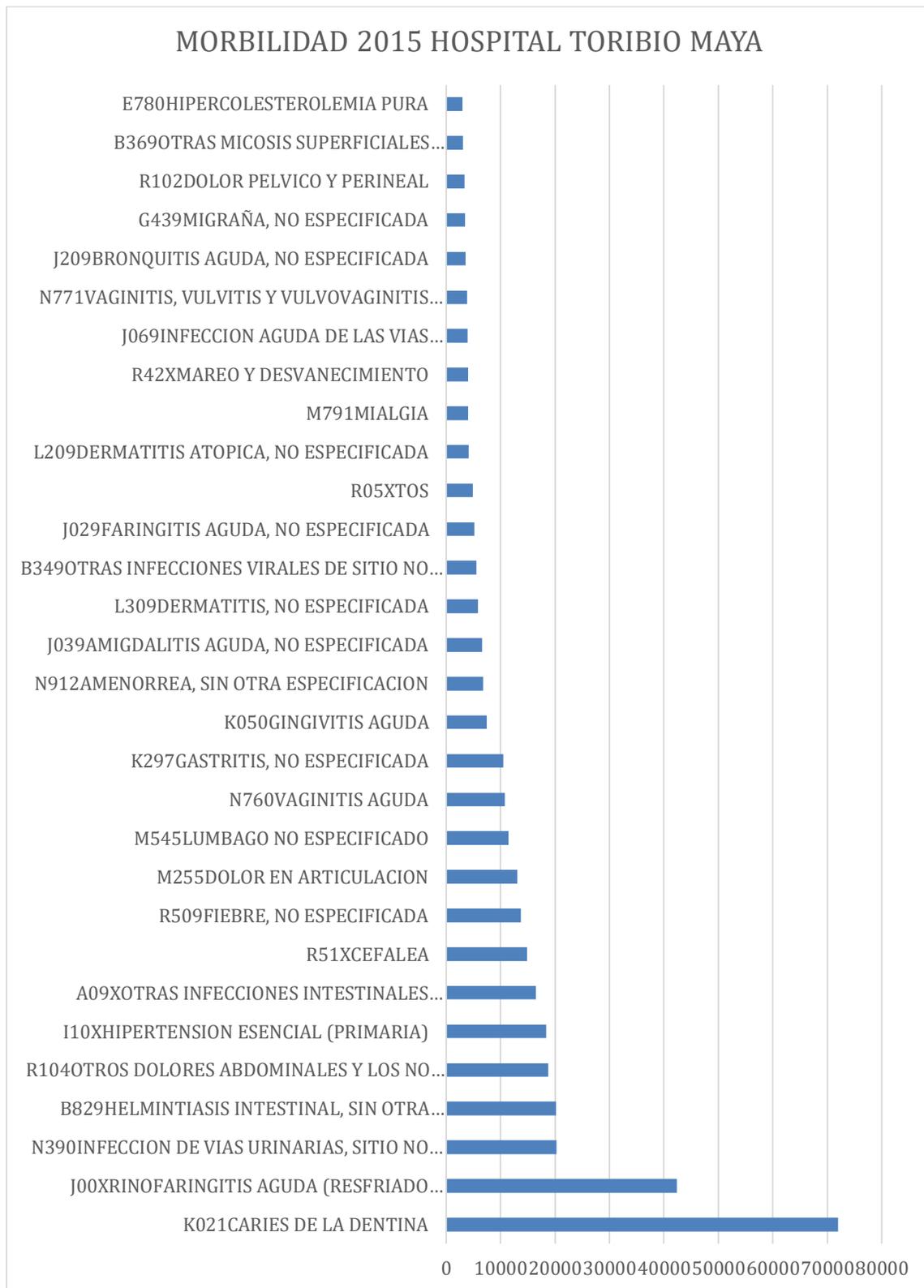
Fuente [Propia]

Gráfica 7. Morbilidad año 2016 hospital Toribio maya



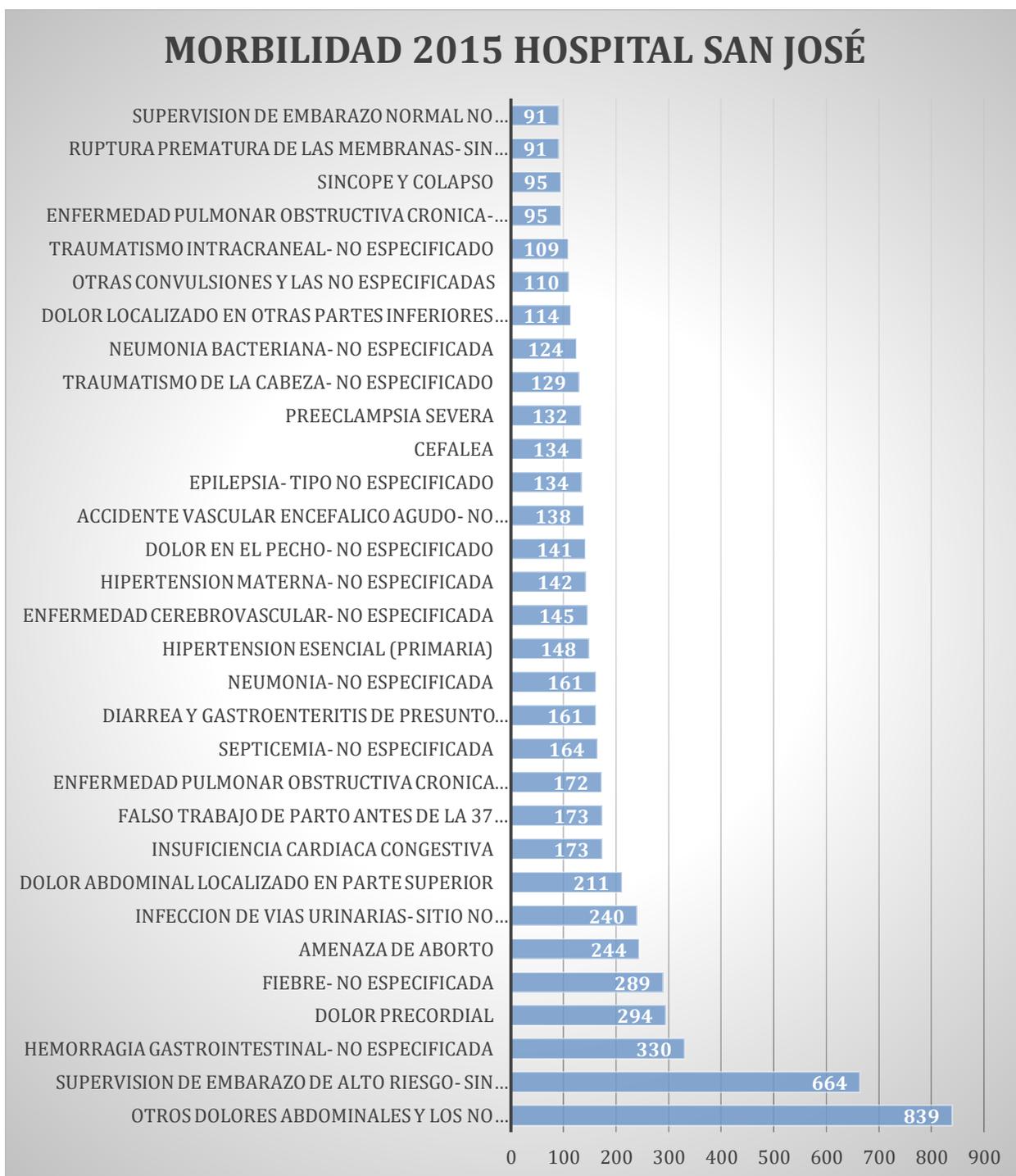
Fuente [Propia]

Gráfica 8. Morbilidad año 2015 hospital Toribio maya



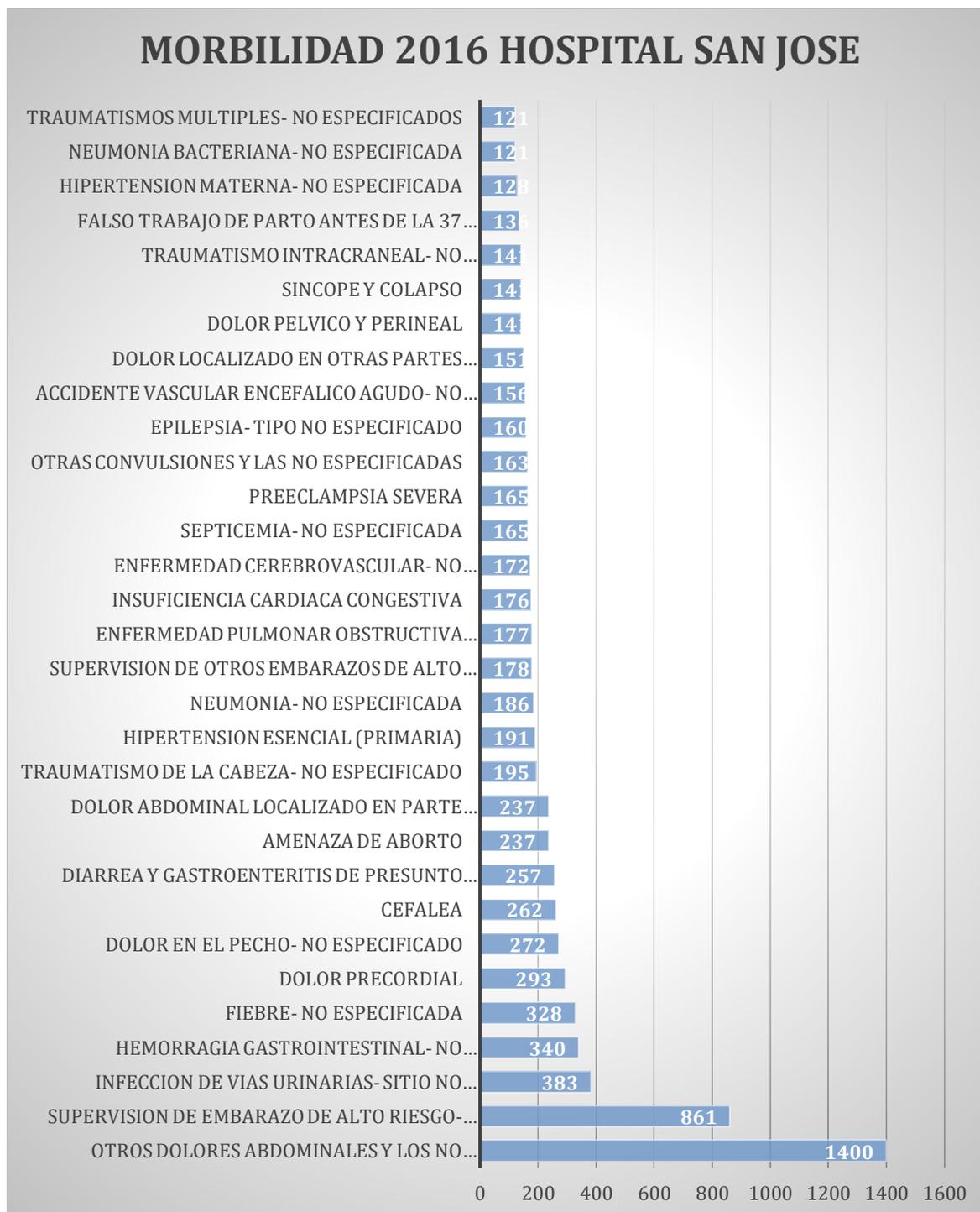
Fuente [Propia]

Gráfica 9. Morbilidad año 2015 hospital San José



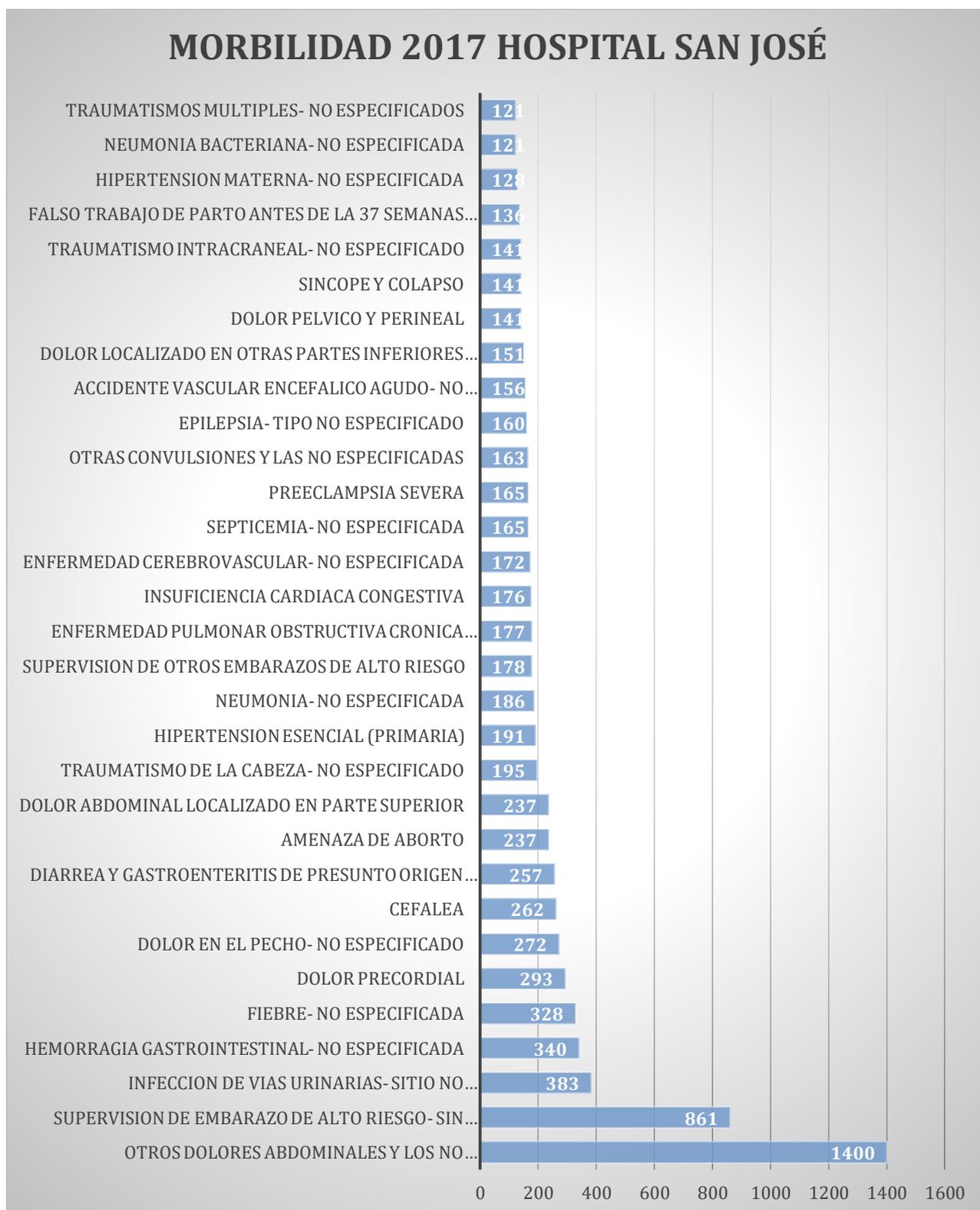
Fuente [Propia]

Gráfica 10. Morbilidad año 2016 hospital San José



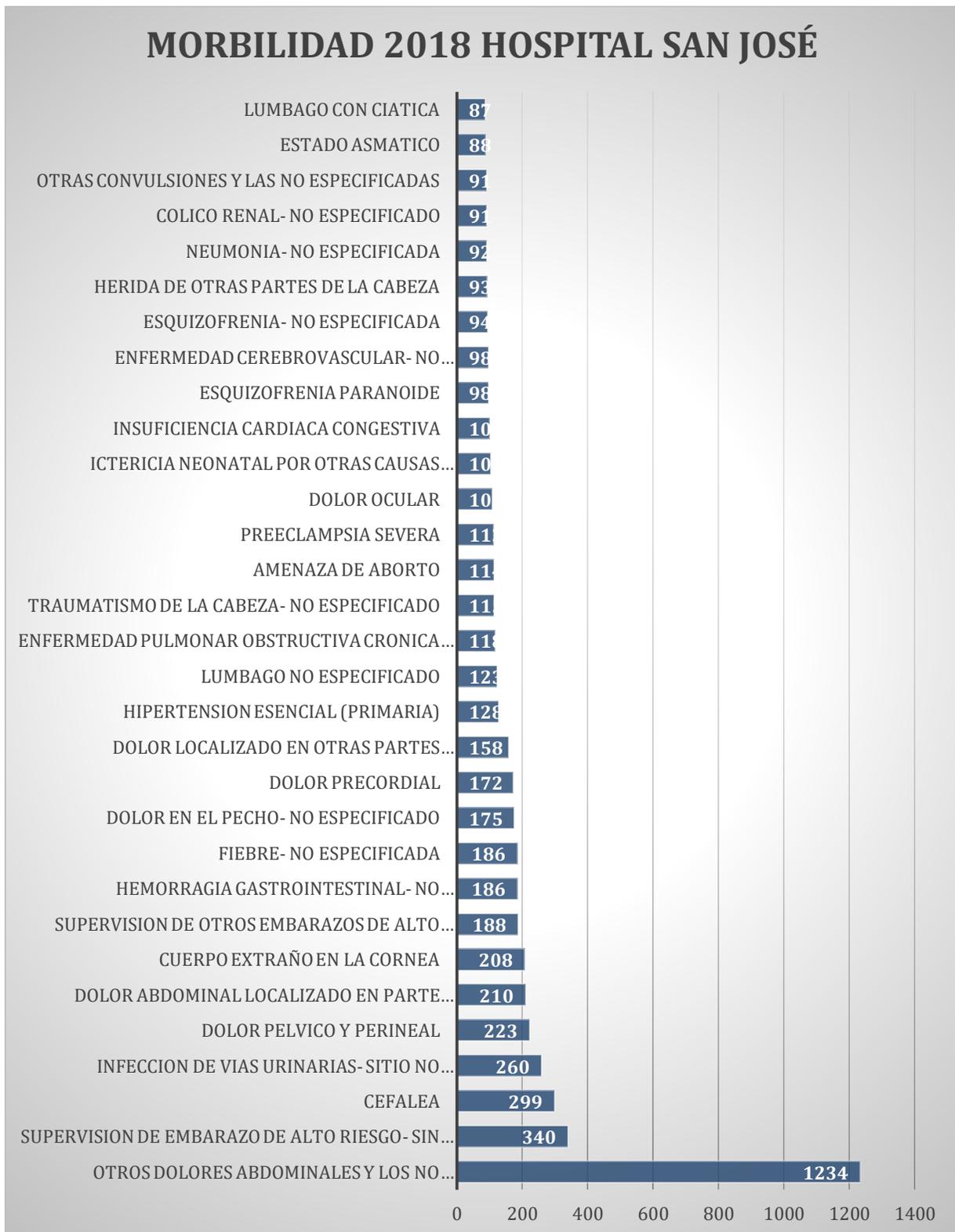
Fuente [Propia]

Gráfica 11. Morbilidad año 2017 hospital San José



Fuente [Propia]

Gráfica 12. Morbilidad año 2018 hospital San José



Fuente [Propia]

En acompañamiento de personal médico calificado se realizan los cuestionarios para cada enfermedad consultada, por ejemplo: se hace uso de la regla mnemotécnica A.L.I.C.I.A (**A**parición, **L**ocalización, **I**ntensidad, **C**aracterísticas, **I**rradiación y **A**gravantes y **A**tenuantes) [38]. Para diagnósticos relacionados con dolor, entre otras técnicas que son de pleno conocimiento de dicho personal médico.

Posterior a la realización de cada cuestionario y en base a las respuestas se asigna el nivel de complejidad que requiere el usuario.

Fueron estimadas las historias de usuario en reuniones en las cuales participaron todos los miembros del equipo. La estimación es una labor de equipo: todos los miembros del equipo deben involucrarse en estimar cada historia. [39]

Se hizo uso de la técnica “planning póker” para poder así dar una estimación de tiempos a las historias de usuario

Como parte fundamental del desarrollo y en base a la metodología XP, se realizaron reuniones en las cuales asistían todos los miembros del desarrollo para así poder exponer avances, dudas y planes a desarrollar. Estas reuniones permitieron dar retroalimentación continua durante todo el proceso de desarrollo.

3.2.2 DISEÑO

En esta fase, el equipo de desarrollo optó por un diseño sencillo, entendible y agradable para el usuario, para facilitar el uso y estimar tiempos en el momento del desarrollo.

3.2.3 CODIFICACIÓN

Para el desarrollo de la plataforma web se seleccionó el lenguaje de programación PHP dado que este lenguaje es creado para el desarrollo web. PHP facilita el acceso a las bases de datos teniendo la gran ventaja de que su código va a ser ejecutado en el lado del servidor además que cuenta con software de libre uso.

Para darle el entorno móvil se optó por usar “Android studio”, por sus diversas herramientas de desarrollo móvil como son su integración de plantillas de código, compatibilidad con herramientas de desarrollo para el proyecto (github) lo que permite llevar control de versiones del desarrollo en tiempo real.

Para la gestión de la base de datos se optó por usar MYSQL y para la administración se usó la herramienta XAMPP, ya que esta herramienta brinda la posibilidad de usar MYSQL, PHPMyAdmin y el servidor APACHE. LARAVEL como framework, este último por sus múltiples funcionalidades de adaptarse a cambios, además de llevar el proceso MVC de una manera simple y flexible.

Para la codificación e implementación del lenguaje de programación se utilizó el editor de texto SUBLIME TEXT, ya que su interfaz es de fácil uso además de dar soporte a varios lenguajes de programación entre ellos (HTML, SQL, PHP.). Sublime no cuenta con un software libre pero la versión de evaluación es plenamente funcional y no tiene fecha de caducidad.

Para el desarrollo de las vistas para el usuario final, se manejó la biblioteca multiplataforma BOOTSTRAP y el lenguaje de maquetado HTML y para darle estilo, organizar la presentación y aspecto a la plataforma web se usó CSS.

3.2.4 PRUEBAS

Basándose en el marco de trabajo SCRUM, sus estrictas estrategias constantes y funciones, se encaminó el desarrollo a obtener pruebas en cada uno de los módulos del código, por esto cada historia de usuario se elaboró con su previa prueba de aceptación para así poder dar por correcta la realización de cada una de ellas.

Para poder conocer la funcionalidad de la plataforma y sus interfaces, se hicieron varias pruebas con distintos usuarios con el fin de saber si es plenamente entendible y fácil de comprender.

4. INGENIERÍA

4.1 IMPLEMENTACIÓN DEL SPRINT 0

4.1.1 ARQUITECTURA DEL SISTEMA

Para el desarrollo de la plataforma web se implementó la estructura cliente/servidor ya que esta gracias a su ventaja de repartir tareas va a permitir tener una concurrencia de datos a la hora de la solicitud de un usuario. Y primordialmente por su escalabilidad y fácil adaptación a los cambios.

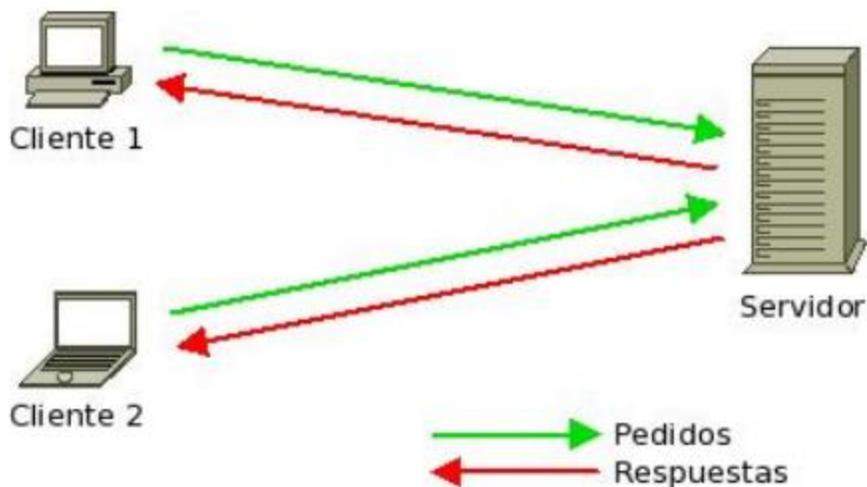
4.1.1.2 CLIENTE

En el desarrollo, partiendo de los principios de la estructura cliente/servidor, el cliente son todas las personas del común con disponibilidad de acceso a internet, ya sea desde cualquier dispositivo móvil o laptops. Para poder cumplir a pleno con la funcionalidad de la plataforma, es puesta a prueba por una persona especializada en medicina general.

4.1.1.3 SERVIDOR

Partiendo con los principios de la estructura cliente/servidor. El servidor se encargará de procesar las peticiones de todos los clientes para posteriormente dar respuesta a la solicitud.

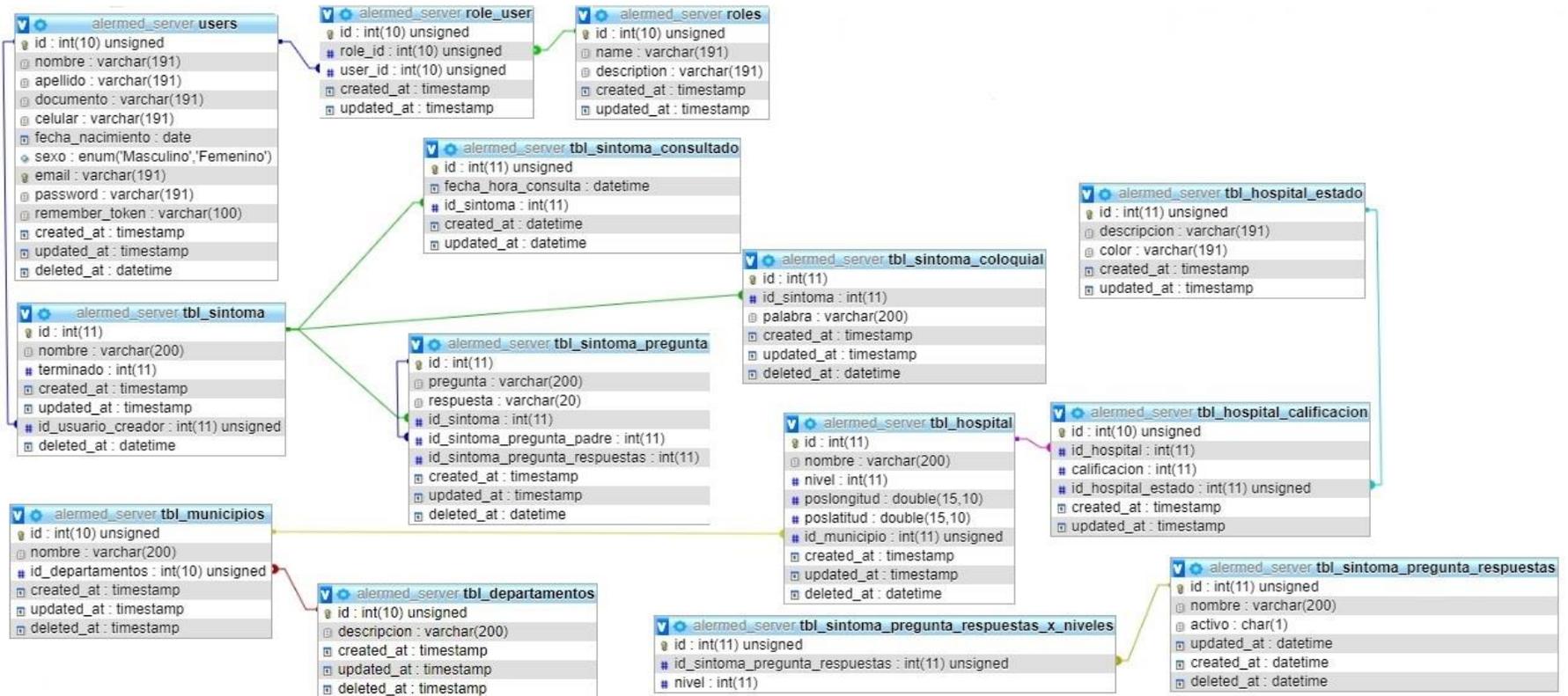
Figura 20. Estructura cliente / servidor.



Fuente [40]

4.1.2 DIAGRAMA DE BASE DE DATOS

Figura 21. Diagrama Base de Datos.



Fuente [Propia]

4.1.3 DESARROLLO DE AGILE INCEPTION

La técnica agile inception brinda dinámicas orientadas a enfocar a todos los miembros de un proyecto hacia un mismo objetivo. Fue parte esencial en el desarrollo puesto que no se tiene un product owner definido, esto a razón de que el desarrollo de este proyecto está dirigido hacia la comunidad y no hacia una empresa o persona como tal. Por ende, para poder conocer las necesidades de la colectividad se estableció un product owner como el interesado del producto final esperado dando respuestas a las siguientes preguntas:

- ¿Por qué estamos aquí?

Conociendo el sistema de salud colombiano podemos darnos cuenta que es defectuoso en muchos aspectos y a su vez está corrompido por ciertos factores en los cuales difícilmente podemos tener incidencia, es por ello que dentro de nuestras experiencias con diferentes herramientas tecnológicas hemos encontrado una forma de ayudar a la comunidad en el aspecto informativo, ya que es básicamente el desconocimiento al momento de proceder en caso de un padecimiento de salud una causa principal del colapso de los servicios de urgencias

- ✓ ¿Por qué estamos haciendo este producto?

Razón 1: brindar una guía oportuna a quien esté padeciendo una dolencia médica, evitando traslados innecesarios haciendo una correcta escala de los niveles de complejidad en la atención de salud de los servicios de urgencias.

Razón 2: aportar una posible solución al colapso de los servicios de urgencias o en su defecto reducir la asistencia innecesaria a los mismos, para que éstos puedan atender pacientes que realmente estén clasificados para su respectiva complejidad.

Razón principal: todo el esfuerzo y motivación en el desarrollo de este proyecto es brindar una ayuda a la comunidad, que puedan tener una atención de calidad y oportuna, incluyendo la comunidad que presta la atención como son el personal médico y de enfermería y de todos los actores relacionados en un servicio de atención de urgencias médicas.

- **ELEVATOR PITCH**

Es una aplicación para toda la comunidad de Popayán que permite brindar una guía de hacia dónde dirigirse en caso de una urgencia médica, basta con tener un dispositivo con acceso a internet para ingresar a la misma, responder unas preguntas acerca del estado de salud de quien consulta y la aplicación analizará estos datos, su ubicación y le sugerirá un centro hospitalario donde pueda ser atendido, adicional a esto le informará si el hospital o clínica hacia dónde se dirige se encuentra o no colapsado.

- **Crea tu Not List.**

- ✓ **IN**

- Gestionar Usuarios.
- Gestionar Síntomas.
- Gestionar Hospitales.
- Guardar historial.
- Generar nivel de urgencia.
- Generar ruta para pacientes.
- Generar Reportes de estado de los hospitales.
- Tener seguridad de la información.

- ✓ **OUT**

- Generar diagnósticos a pacientes.

- ✓ **DISCUTIR**

- Toma de signos vitales.
- Dar tratamientos médicos.

- **¿Qué te quita el sueño?**

- ✓ ¿Cuáles son tus miedos?

- Que los tiempos no alcancen para el estudio de síntomas.
- Que la plataforma no cuente con la suficiente información idónea para poder dirigir las personas.
- Que la plataforma tenga problemas de conectividad.

- ✓ ¿Cómo evitarlos?

- Dedicándole el tiempo pertinente al estudio de síntomas.

- Brindándole a la base de datos información validada por gente especialista en el tema.
 - Implementando una buena red para la plataforma contando con un servidor que brinde las técnicas de TI para el servicio.
- ✓ ¿Cómo minimizar el riesgo?
- Siguiendo el marco de trabajo con sus tiempos estimados y planes a desarrollar.
 - Realizando pruebas constantes a las funcionalidades de la plataforma.
- ✓ ¿Qué nos mantiene despiertos por las noches?
- Riesgo 1: Los usuarios finales no entiendan las funcionalidades de la plataforma.
 - Riesgo 2: El desarrollo no se termine en el tiempo establecido.
 - Impedimento 1: La interfaz sea confusa para los usuarios.
 - Impedimento 2: Estudio de todos los síntomas establecidos para una urgencia, desglosando preguntas claves para poder saber el nivel de urgencia de cada consultor.

4.1.3.1 SPRINT REVIEW

Para poder cumplir con los objetivos tanto en tiempo como en funcionalidad, se priorizaron las historias de usuario de modo que permitiera empezar en un orden específico para cada sprint socializando cada funcionalidad que tiene.

4.1.3.2 RETROSPECTIVA

Se empleó la retrospectiva agile “bote de velocidad”, identificando los cuatro segmentos que trata. (Fortalezas, debilidades, amenazas y objetivos), Brindando también retrospectiva en cada interacción.

Tabla 6. Retrospectiva del sprint 0, proyecto en general.

<p>Fortalezas</p> <ul style="list-style-type: none"> ● Uso adecuado de las herramientas de desarrollo. ● Disposición de tiempos para reuniones con el equipo de trabajo. ● Acceso fácil a material y personal especializado en el área de medicina general. 	<p>Debilidades</p> <ul style="list-style-type: none"> ● Comprensión de síntomas generales. ● Poca información brindada de algunos centros hospitalarios de la ciudad de Popayán.
<p>Amenazas</p> <ul style="list-style-type: none"> ● Tiempo de desarrollo. ● Estudio de síntomas. 	<p>Objetivos</p> <ul style="list-style-type: none"> ● Empezar a documentar el proceso de desarrollo. ● Establecer reuniones con el interesado del producto. ● Compartir constantemente avances del proyecto.

4.2 IMPLEMENTACIÓN SPRINT 1.

Está conformado por 5 historias de usuario estimadas en un tiempo de ejecución de 20 días (hábiles) de 7 horas de trabajo individual para cada entrega o iteración. En cada una de estas se realizó su respectiva planeación, diseño, desarrollo y pruebas.

- Autenticar usuario.
- Registrar Usuario.
- Actualizar Usuario.
- Eliminar Usuario.
- Recuperar contraseña.

4.2.1 PLANIFICACIÓN

A continuación, se manifiestan las historias de usuario del sprint 1.

Tabla 7. Historia de usuario autenticar usuario

Historia de usuario	
Número: HU001	
Programador Responsable: Team development	
Nombre de la historia: Autenticar Usuario	
Nivel de riesgo de desarrollo: medio	
Peso de la historia: 3 puntos	
Tiempo estimado: 8 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: autenticarse en la plataforma</p> <p>Para: Poder acceder a las funciones correspondientes.</p>	
<p>Criterios de aceptación</p> <p>1. Registro exitoso: Cuando se encuentre habilitado por el sistema y se ingresen los datos correctamente (usuario y contraseña como campos obligatorios) se permitirá el acceso a la plataforma.</p> <p>2. Registro no exitoso: cuando se ingresen los datos de manera incorrecta se mostrará un mensaje de error y no se permitirá el acceso a la plataforma.</p> <p>Restricciones:</p> <p>1. El usuario asignado debe ser único</p>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear Interfaces. • Conexión al servidor. 	<p>2</p> <p>1</p> <p>1</p>

<ul style="list-style-type: none">• Validación de la información ingresada por el usuario a través de la interfaz.• Testing• Integración	2 2
--	--------

Tabla 8. Historia de usuario Registrar Usuario.

Historia de usuario	
Número: HU002	
Programador Responsable: Team development	
Nombre de la historia: Registrar Usuario	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Registrar un nuevo usuario</p> <p>Para: Permitir acceso a todas las funcionalidades de la plataforma.</p>	
<p>Criterios de aceptación</p> <ol style="list-style-type: none"> 1. Registro de un nuevo usuario exitoso: Llenar campos obligatorios: nombre, apellido, identificación, email, rol, nombre de usuario y contraseña; la plataforma informará que el registro se realizó satisfactoriamente 2. Registro fallido: en caso de no contar con los campos requeridos la plataforma informará qué campos hacen falta para poder crear el registro 3. El Nombre de usuario y la contraseña serán asignados por el administrador. <p>Restricciones</p> <ul style="list-style-type: none"> ● los usuarios solo podrán ser creados por el usuario administrador. ● Se puede hacer uso de la plataforma, pero no con todas sus funcionalidades. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> ● Crear Interfaces. ● Conexión al servidor. ● Validación de datos ingresados por el usuario. 	<p>2</p> <p>1</p> <p>1</p>

• Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz	2
• Testing	2
• Integración	2

Tabla 9. Historia de Usuario Actualizar Usuario.

Historia de usuario	
Número: HU003	
Programador Responsable: Team development	
Nombre de la historia: Actualizar Usuario	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir Actualizar datos de usuarios registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <p>1. Actualización de usuario exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el usuario que desee y los campos que desee modificar bien sea:” nombre, apellido, identificación, email, rol, nombre de usuario y contraseña”, la plataforma informará que la actualización se realizó satisfactoriamente</p> <p>2. Actualización de usuario fallida: En caso de no contar con la autenticación del usuario, contraseña o llenar alguno de los campos a modificar erróneamente la plataforma informará qué campos son los que presentan conflictos para poder actualizar.</p> <p>Restricciones</p> <ul style="list-style-type: none"> los usuarios solo podrán ser modificados por el usuario administrador. 	
Actividades	Nº de horas de implementación
<ul style="list-style-type: none"> Crear Interfaces. 	2

• Conexión al servidor.	1
• Validación de datos ingresados por el usuario.	1
• Actualizar registro con los datos ingresados por el administrador por medio de la interfaz	2
• Testing	
• Integración	2
	2

Tabla 10. Historia de Usuario Eliminar Usuario.

Historia de usuario	
Número: HU004	
Programador Responsable: Team development	
Nombre de la historia: Eliminar Usuario	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir eliminar usuarios registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <p>3. Eliminar usuario exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el usuario que desee eliminar la plataforma informará que el usuario se eliminó satisfactoriamente.</p> <p>4. Eliminar usuario fallida: En caso de no contar con la autenticación del usuario y contraseña, la plataforma informará qué campos son los que presentan conflictos para poder ingresar al sistema y posteriormente eliminar usuario.</p> <p>Restricciones</p> <ul style="list-style-type: none"> los usuarios solo podrán ser eliminados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> Crear Interfaces. Conexión al servidor. Validación de datos ingresados por el usuario. Eliminar un usuario registrado por medio de la interfaz 	<p>2</p> <p>1</p> <p>1</p>

• Testing	2
• Integración	2
	2

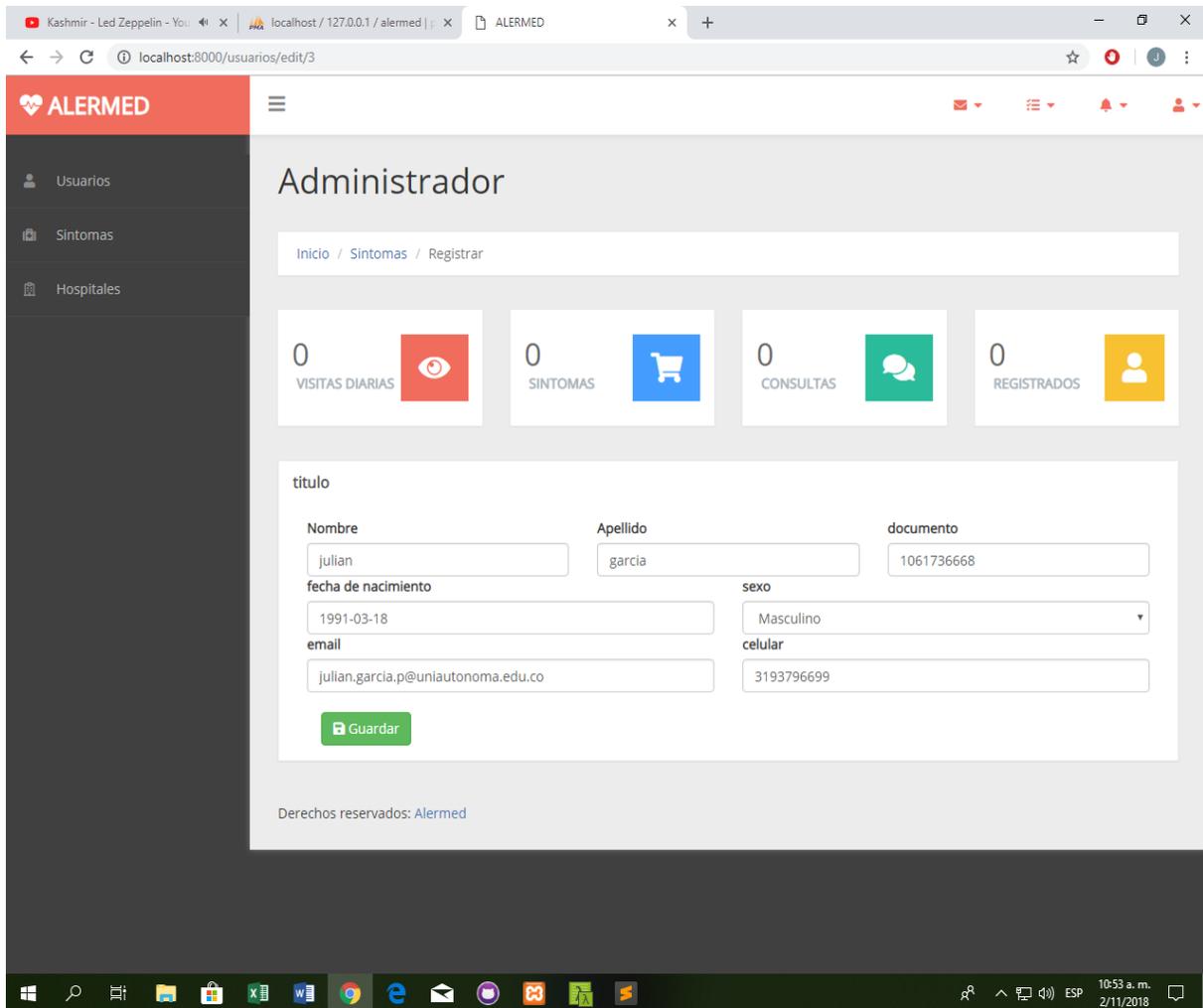
Tabla 11. Historia de Usuario Recuperar Contraseña.

Historia de usuario	
Número: HU005	
Programador Responsable: Team development	
Nombre de la historia: Recuperar Contraseña	
Nivel de riesgo de desarrollo: media	
Peso de la historia: 5 puntos	
Tiempo estimado: 9 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Recuperar la contraseña</p> <p>Para: Poder tener acceso a la plataforma en caso de no recordar la contraseña</p>	
<p>Criterios de aceptación</p> <p>1. Recuperar Contraseña: requiere que el usuario esté registrado en la plataforma. Cuando el usuario necesite recuperar su contraseña, ingresa su dirección de correo electrónico ahí se enviará los datos para su recuperación.</p> <p>2. Recuperar contraseña exitosa: Si el usuario está registrado en la plataforma y su correo es correcto, se envía un mensaje informando que se ha enviado los datos de recuperación al correo electrónico.</p> <p>3. Recuperar contraseña no exitosa: Se envía un mensaje de error informando al usuario que no se puede realizar ya que el correo no es válido.</p>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear interfaces • Conexión al servidor • Crear recuperación de contraseña • Testing • Integración 	<p>2</p> <p>1</p> <p>2</p> <p>2</p> <p>2</p>

4.2.2 DISEÑO

En la Figura 22 se muestra la interfaz de registrar usuario

Figura 22. Interfaz de Usuario “Registrar Usuario”.



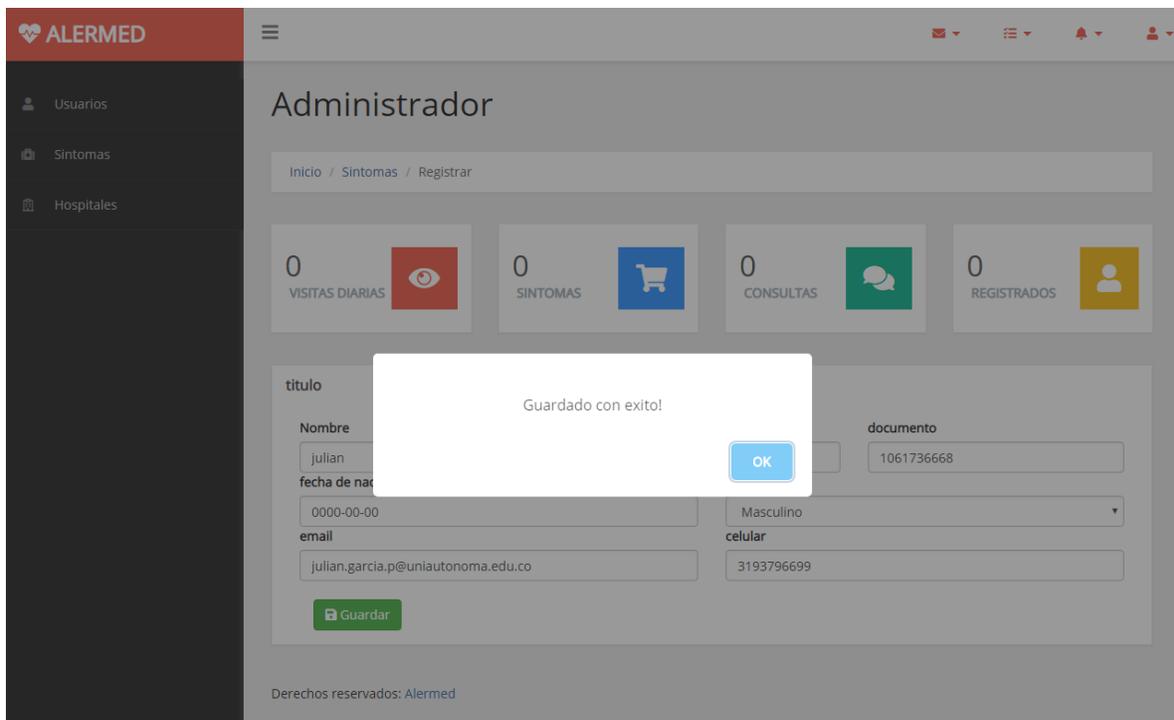
Fuente [Propia]

4.2.4 PRUEBAS

Tabla 12. Prueba funcional.

Historia de Usuario	Registrar Usuario
Propósito	Crear un usuario para asignarle un rol
Requisitos	Ser súper-administrador del sistema
Pasos	El súper administrador se autentica en la plataforma, ahí en ese momento en la parte izquierda del sistema aparecerá la opción usuarios para después agregar usuario. En este se desplegará un menú con los campos requeridos para el registro de un nuevo usuario.
Resultados Esperados	un mensaje que verifique el registro

Figura 23. Prueba Funcional Registrar Usuario.



Fuente [Propia]

4.2.5 SPRINT REVIEW.

Se hicieron entrega de las funcionalidades (Autenticar usuario, registrar usuario, actualizar usuario, eliminar usuario y recuperar contraseña), todas estas con su respectiva prueba para así poder tener una primera entrega del software funcional.

4.2.6 RETROSPECTIVA

Para tener una retrospectiva agradable se hizo uso de la técnica “bote de velocidad” la cual permite identificar cuáles fueron las (Fortalezas, amenazas, debilidades y objetivos) del sprint para posteriormente retroalimentar sus falencias para el siguiente sprint.

Tabla 13. Retrospectiva del sprint 1.

Fortalezas <ul style="list-style-type: none">● Reuniones constantes de equipo● Retroalimentación del desarrollo● Reestructuración de las funcionalidades del sistema.	Debilidades <ul style="list-style-type: none">● Reuniones con personal médico.● Reuniones con el cliente.● No se cuenta con un cliente definido.
Amenazas <ul style="list-style-type: none">● Llegar tarde a las reuniones programadas por el equipo de desarrollo● Posponer reuniones con equipo y cliente● Entendimiento de cada rol dentro del desarrollo.	Objetivos <ul style="list-style-type: none">● Organizar funcionalidades de cada sprint.● Refinamiento de requisitos.● Pruebas constantes a cada funcionalidad del sistema.

4.3 IMPLEMENTACIÓN SPRINT 2

Está conformado por las siguientes historias de usuario, las cuales se estimaron para desarrollar en un tiempo estimado de 30 días hábiles de 7 horas de trabajo individual.

- Registrar Síntoma.
- Actualizar Síntoma.
- Eliminar Síntoma.
- Generar nivel de urgencia.

4.3.1 PLANIFICACIÓN

A continuación, se manifiestan las historias de usuario del sprint 2.

Tabla 14. Historia de usuario Registrar Síntoma.

Historia de usuario	
Número: HU006	
Programador Responsable: Team development	
Nombre de la historia: Registrar Síntoma.	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 4 puntos	
Tiempo estimado: 15 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticación en la plataforma</p> <p>Para: Ingresar Síntoma</p>	
<p>Criterios de aceptación</p> <p>1. Registro de síntoma exitoso: Después de autenticarse debidamente en la plataforma. El administrador podrá ingresar el síntoma a registrar en la plataforma. Llenar campos obligatorios: síntoma y preguntas por síntoma para posteriormente guardar. La plataforma informará que el registro se realizó satisfactoriamente</p> <p>2. Registro fallido: en caso de no contar con los campos requeridos la plataforma informará qué campos hacen falta para poder crear el registro</p> <p>Restricciones</p> <ul style="list-style-type: none"> • los síntomas solo podrán ser creados por el usuario administrador. 	
Actividades	Nº de horas de implementación

• Crear Interfaces.	2
• Conexión al servidor.	1
• Validación de datos ingresados por el usuario.	1
• Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz	7
• Testing	2
• Integración	2

Tabla 15. Historia de Usuario Actualizar Síntoma.

Historia de usuario	
Número: HU007	
Programador Responsable: Team development	
Nombre de la historia: Actualizar Síntoma	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir Actualizar datos de síntomas registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <p>1. Actualización de síntoma exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el usuario que desee y los campos que desee modificar bien sea:” nombre, apellido, identificación, email, rol, nombre de usuario y contraseña”, la plataforma informará que la actualización se realizó satisfactoriamente</p> <p>2. Actualización de síntoma fallida: En caso de no contar con la autenticación del usuario, contraseña o llenar alguno de los campos a modificar erróneamente la plataforma informará qué campos son los que presentan conflictos para poder actualizar.</p> <p>Restricciones</p> <ul style="list-style-type: none"> • los síntomas solo podrán ser modificados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear Interfaces. • Conexión al servidor. 	<p>2</p> <p>1</p>

<ul style="list-style-type: none"> • Validación de datos ingresados por el usuario. 	1
<ul style="list-style-type: none"> • Actualizar registro de síntoma con los datos ingresados por el administrador por medio de la interfaz 	2
<ul style="list-style-type: none"> • Testing 	2
<ul style="list-style-type: none"> • Integración 	2

Tabla 16. Historia de Usuario Eliminar Síntoma.

Historia de usuario	
Número: HU008	
Programador Responsable: Team development	
Nombre de la historia: Eliminar Síntoma	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir eliminar síntomas registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <ol style="list-style-type: none"> 1. Eliminar síntoma exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el síntoma que desee eliminar la plataforma informará que el síntoma se eliminó satisfactoriamente. 2. Eliminar síntoma fallida: En caso de no contar con la autenticación del usuario y contraseña, la plataforma informará qué campos son los que presentan conflictos para poder ingresar al sistema y posteriormente eliminar síntoma. <p>Restricciones</p> <ul style="list-style-type: none"> • los usuarios solo podrán ser eliminados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear Interfaces. • Conexión al servidor. • Validación de datos ingresados por el usuario. • Eliminar un síntoma registrado por medio de la interfaz • Testing 	<p>2</p> <p>1</p> <p>1</p> <p>2</p>

• Integración	2 2
---------------	--------

Tabla 17. Historia de usuario Generar nivel de urgencia.

Historia de usuario	
Número: HU009	
Programador Responsable: Team development	
Nombre de la historia: Generar nivel de urgencia	
Nivel de riesgo de desarrollo: medio	
Peso de la historia: 10 puntos	
Tiempo estimado: 15 Horas	
<p>Perspectiva del producto</p> <p>Como: Usuario</p> <p>Requiero: Ingresar síntoma, responder cuestionario sobre síntoma.</p> <p>Para: Generar nivel de urgencia</p>	
<p>Criterios de aceptación</p> <ul style="list-style-type: none"> ● Búsqueda exitosa: cuando se haya especificado el síntoma y posteriormente respondido el cuestionario sobre dicho síntoma entonces el sistema dará el nivel de urgencia con sus respectivos centros de atención. ● Búsqueda no exitosa: Si el síntoma ingresado no es válido o no se ha dado respuesta al cuestionario, entonces el sistema informara al usuario el error. <p>Restricciones:</p> <p>El sistema no tiene restricciones para este.</p>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> ● Crear Interfaces. ● Conexión al servidor. ● Generar nivel de urgencia. ● Testing. ● Integración. 	<p>2</p> <p>2</p> <p>5</p> <p>2</p> <p>2</p>

4.3.2 DISEÑO

En la Figura 24 se muestra la interfaz de usuario de registrar síntoma.

Figura 24. Interfaz de Registrar Síntoma.

welcome to the jungle - You | ALERMED | No es seguro | alermed.solucionescrv.com/sintomas/nuevo

ALERMED

Usuarios
Síntomas
Hospitales

titulo

Nombre: dolor de cabeza

Síntoma coloquial

Coloque el nombre coloquial

dolor de tuste

+ Pregunta

Pregunta

eh sufrido dolor de cabeza | 1

+ Pregunta - Borrar

Pregunta

no eh sufrido dolor de cabeza | Consulta externa

+ Pregunta - Borrar

Guardar

Derechos reservados: Alermed

11:10 a.m. 22/01/2019

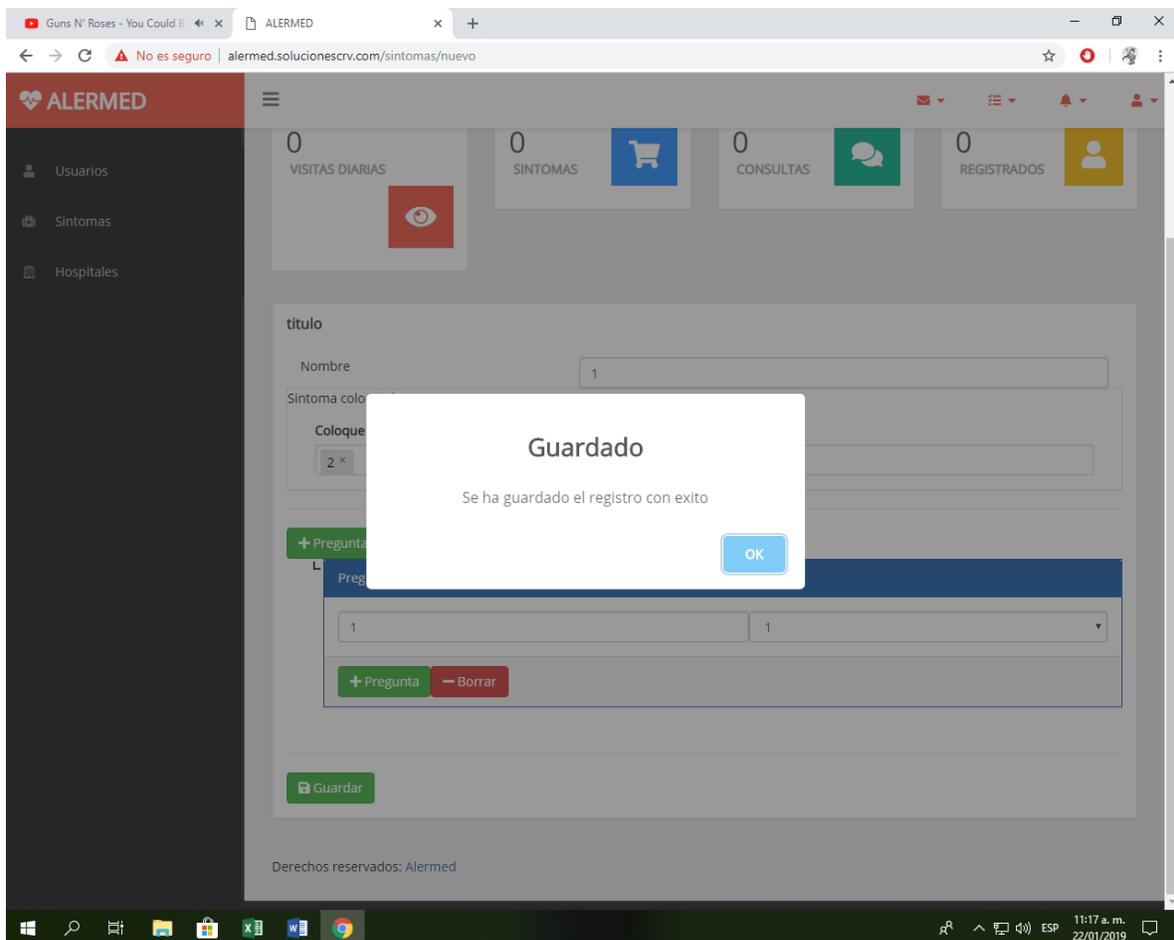
Fuente [Propia]

4.3.4 PRUEBAS

Tabla 18. Prueba funcional registrar síntoma.

Historia de Usuario	Registrar síntoma.
Propósito	Ingresar el síntoma a registrar.
Requisitos	Ser administrador o usuario con permisos para este rol.
Pasos	El administrador o usuario se autentica en la plataforma posteriormente en la pestaña síntomas para poder ingresar un síntoma. Llenar campos requeridos: “nombre, nombre coloquial o sinónimo del síntoma y sus respectivas preguntas” guardar.
Resultados Esperados	Confirmación de registro guardado.

Figura 25. Prueba Funcional Ingresar Síntoma.



Fuente [Propia]

4.3.5 SPRINT REVIEW.

Se hicieron entrega de las funcionalidades (Registrar síntoma, Actualizar síntoma, Eliminar Síntoma y generar nivel de urgencia), todas estas con su respectiva prueba para así poder tener una primera entrega del software funcional.

4.3.6 RETROSPECTIVA

Para tener una retrospectiva agradable se hizo uso de la técnica “bote de velocidad” la cual permite identificar cuáles fueron las (Fortalezas, amenazas, debilidades y objetivos) del sprint para posteriormente retroalimentar sus falencias para el siguiente sprint.

Tabla 19. Retrospectiva del sprint 2.

Fortalezas <ul style="list-style-type: none">● Reuniones con personal médico.● Trabajo en equipo● Retroalimentación continua● Documentación sobre síntomas principales	Debilidades <ul style="list-style-type: none">● Poco conocimiento sobre síntomas principales.● Tiempo prolongado para estudio de síntomas
Amenazas <ul style="list-style-type: none">● No cumplir con el tiempo previsto para el sprint● Comprensión de síntomas● Estructuración de preguntas sobre síntomas	Objetivos <ul style="list-style-type: none">● Cumplir con el tiempo estimado para el sprint● Trabajo continuo para estudio de síntomas● reuniones constantes con el equipo de desarrollo.

4.4 IMPLEMENTACIÓN SPRINT 3

Está conformado por las siguientes historias de usuario, las cuales se estimaron para desarrollar en un tiempo estimado de 20 días hábiles de 8 horas de trabajo individual.

- Consultar Síntoma
- Registrar Hospital
- Actualizar Hospital
- Eliminar Hospital
- Consultar información de Hospital
- Registrar calificación de Hospital

4.4.1 PLANIFICACIÓN

A continuación, se manifiestan las historias de usuario sprint 3.

Tabla 20. Historia de usuario consultar síntoma.

Historia de usuario
Número: HU010
Programador Responsable: Team development
Nombre de la historia: Consultar síntoma
Nivel de riesgo de desarrollo: medio
Peso de la historia: 4 puntos
Tiempo estimado: 13 Horas
Perspectiva del producto Como: Usuario Requiero: Consultar síntoma Para: poder elegir sus síntomas principales de la lista y posteriormente ver el nivel de la urgencia.
Criterios de aceptación

- Consulta de síntoma exitoso: si el síntoma ingresado está bien escrito posteriormente pasara a un cuestionario de preguntas relacionadas a este síntoma.
- Consulta de síntoma no exitoso: Si el síntoma ingresado no está bien escrito o no se reconoce en el sistema entonces se mostrará un mensaje informándoselo al usuario.

Restricciones:

El sistema no tiene restricciones para esta.

Actividades	N° de horas de implementación
<ul style="list-style-type: none"> ● Crear Interfaces. 	2
<ul style="list-style-type: none"> ● Conexión al servidor. 	2
<ul style="list-style-type: none"> ● Consultar síntoma. 	5
<ul style="list-style-type: none"> ● Testing. 	2
<ul style="list-style-type: none"> ● Integración. 	2

Tabla 21. Historia de usuario Registrar Hospital.

Historia de usuario	
Número: HU011	
Programador Responsable: Team development	
Nombre de la historia: Registrar Hospital.	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 4 puntos	
Tiempo estimado: 15 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticación en la plataforma</p> <p>Para: Registrar un nuevo hospital</p>	
<p>Criterios de aceptación</p> <p>3. Registro de hospital exitoso: Después de autenticarse debidamente en la plataforma. El administrador podrá ingresar un nuevo hospital en la plataforma. Llenar campos obligatorios: nombre, nivel y coordenadas (latitud y longitud). Para posteriormente guardar. La plataforma informará que el registro se realizó satisfactoriamente</p> <p>4. Registro de hospital fallido: en caso de no contar con los campos requeridos la plataforma informará qué campos hacen falta para poder crear el registro</p> <p>Restricciones</p> <ul style="list-style-type: none"> • los hospitales solo podrán ser creados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear Interfaces. • Conexión al servidor. • Validación de datos ingresados por el usuario. 	<p>2</p> <p>1</p> <p>1</p>

• Crear nuevo registro con los datos ingresados por el administrador por medio de la interfaz	7
• Testing	2
• Integración	2

Tabla 22. Historia de Usuario Actualizar Hospital.

Historia de usuario	
Número: HU012	
Programador Responsable: Team development	
Nombre de la historia: Actualizar Hospital	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir Actualizar datos de los hospitales registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <p>3. Actualización de hospital exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el hospital que desee y los campos que desee modificar bien sea: nombre, nivel o coordenadas (latitud y longitud). La plataforma informará que la actualización se realizó satisfactoriamente</p> <p>4. Actualización de hospital fallida: En caso de no contar con la autenticación del usuario, contraseña o llenar alguno de los campos a modificar erróneamente la plataforma informará qué campos son los que presentan conflictos para poder actualizar.</p> <p>Restricciones</p> <ul style="list-style-type: none"> los hospitales solo podrán ser modificados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> Crear Interfaces. Conexión al servidor. Validación de datos ingresados por el usuario. 	<p>2</p> <p>1</p> <p>1</p>

<ul style="list-style-type: none"> • Actualizar registro del hospital con los datos ingresados por el administrador por medio de la interfaz • Testing • Integración 	<p>2</p> <p>2</p> <p>2</p>
---	----------------------------

Tabla 23. Historia de Usuario Eliminar Hospital.

Historia de usuario	
Número: HU013	
Programador Responsable: Team development	
Nombre de la historia: Eliminar Hospital	
Nivel de riesgo de desarrollo: Media	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Administrador</p> <p>Requiero: Autenticarse en la plataforma</p> <p>Para: Permitir eliminar hospitales registrados en la plataforma.</p>	
<p>Criterios de aceptación</p> <p>3. Eliminar hospital exitoso: Se autentica en la plataforma con los campos requeridos (usuario y contraseña), selecciona el hospital que desee eliminar la plataforma informará que el hospital se eliminó satisfactoriamente.</p> <p>4. Eliminar hospital fallida: En caso de no contar con la autenticación del usuario y contraseña, la plataforma informará qué campos son los que presentan conflictos para poder ingresar al sistema y posteriormente eliminar hospital.</p> <p>Restricciones</p> <ul style="list-style-type: none"> los hospitales solo podrán ser eliminados por el usuario administrador. 	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> Crear Interfaces. Conexión al servidor. Validación de datos ingresados por el usuario. Eliminar un hospital registrado por medio de la interfaz Testing 	<p>2</p> <p>1</p> <p>1</p> <p>2</p>

• Integración	2 2
---------------	--------

Tabla 24. Historia de usuario consultar información de hospital.

Historia de usuario	
Número: HU014	
Programador Responsable: Team development	
Nombre de la historia: Consultar información de hospital	
Nivel de riesgo de desarrollo: medio	
Peso de la historia: 3 puntos	
Tiempo estimado: 10 Horas	
<p>Perspectiva del producto</p> <p>Como: Usuario</p> <p>Requiero: Consultar información del hospital.</p> <p>Para: Posteriormente tener todos los datos referentes al hospital</p>	
<p>Criterios de aceptación</p> <ul style="list-style-type: none"> ● Consultar información Exitoso: El usuario podrá consultar y visualizar la información referente al hospital (calificación). <p>Restricciones:</p> <p>El sistema no presenta restricciones para esta.</p>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> ● Crear Interfaces. 	2
<ul style="list-style-type: none"> ● Conexión al servidor. 	2
<ul style="list-style-type: none"> ● Consultar información de hospital. 	2
<ul style="list-style-type: none"> ● Testing. 	2
<ul style="list-style-type: none"> ● Integración. 	2

Tabla 25. Registrar Calificación de Hospital.

Historia de usuario	
Número: HU015	
Programador Responsable: Team development	
Nombre de la historia: Registrar calificación de hospital	
Nivel de riesgo de desarrollo: medio	
Peso de la historia: 3 puntos	
Tiempo estimado: 11 Horas	
<p>Perspectiva del producto</p> <p>Como: Usuario</p> <p>Requiero: Ingresar hospital</p> <p>Para: Posteriormente enviar un reporte de cómo se encuentra el hospital en ese momento.</p>	
<p>Criterios de aceptación</p> <p>1. Registrar calificación de hospital: El usuario podrá calificar el estado actual del hospital dependiendo de cómo se encuentre actualmente.</p> <p>Restricciones:</p> <p>El sistema no presenta restricciones para esta.</p>	
Actividades	N° de horas de implementación
<ul style="list-style-type: none"> • Crear Interfaces. • Conexión al servidor. • Generar Calificación del estado actual del hospital. • Testing. • Integración. 	<p>2</p> <p>2</p> <p>3</p> <p>2</p> <p>2</p>

4.4.2 DISEÑO

En la Figura 26 se muestra la interfaz de usuario de ver estado de hospital

Figura 26. Interfaz de Usuario ver Estado de Hospitales.



Fuente [Propia]

4.4.4 PRUEBAS

Tabla 26. Prueba funcional.

Historia de Usuario	Consultar estado de hospitales
Propósito	Ver el estado actual de algún hospital
Requisitos	El sistema no tiene requisitos para esta funcionalidad
Pasos	El usuario entra a la plataforma, en una pestaña en la parte superior encontrara la opción "ver hospitales", el sistema desplegara en un mapa los hospitales de la ciudad registrados en la plataforma con el estado en que se encuentran.
Resultados Esperados	Conocer cómo se encuentra dicho hospital

Figura 27. Prueba Funcional.



Fuente [Propia]

4.4.5 SPRINT REVIEW.

Se hizo la entrega de las funcionalidades (Consultar síntoma, Consultar información de hospital y registrar calificación de hospital), todas estas con su respectiva prueba para así poder tener una primera entrega del software funcional.

4.4.6 RETROSPECTIVA

Para tener una retrospectiva agradable se hizo uso de la técnica “bote de velocidad” la cual permite identificar cuáles fueron las (Fortalezas, amenazas, debilidades y objetivos) del sprint para posteriormente retroalimentar sus falencias para el siguiente sprint.

Tabla 27. Retrospectiva del sprint 3.

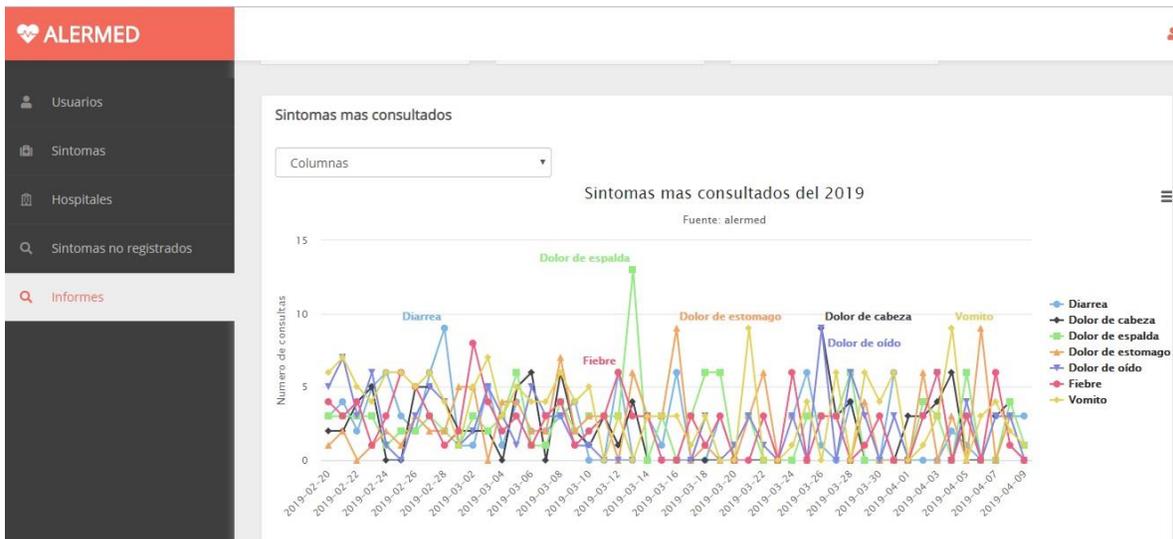
<p>Fortalezas</p> <ul style="list-style-type: none"> ● Información de hospitales. ● Trabajo en equipo 	<p>Debilidades</p> <ul style="list-style-type: none"> ● Comprensión de la funcionalidad del sprint.
<p>Amenazas</p> <ul style="list-style-type: none"> ● Postergar reuniones con el equipo de desarrollo. ● Tiempo estimado ● Cambio de diseño 	<p>Objetivos</p> <ul style="list-style-type: none"> ● Cumplir con los tiempos estimados para el sprint ● Retroalimentación continua en cada etapa del desarrollo

RESULTADOS

Debido a que la plataforma se encuentra en fase de implementación y que además de ello los resultados deben medirse por indicadores que emiten las entidades reguladoras como son la Secretaria Departamental de Salud del Cauca y el Ministerio de Salud y la Protección Social, para efectos de inmediatez se realizaron pruebas en espacios reales como las salas de urgencias, con pacientes que consultan dicho servicio.

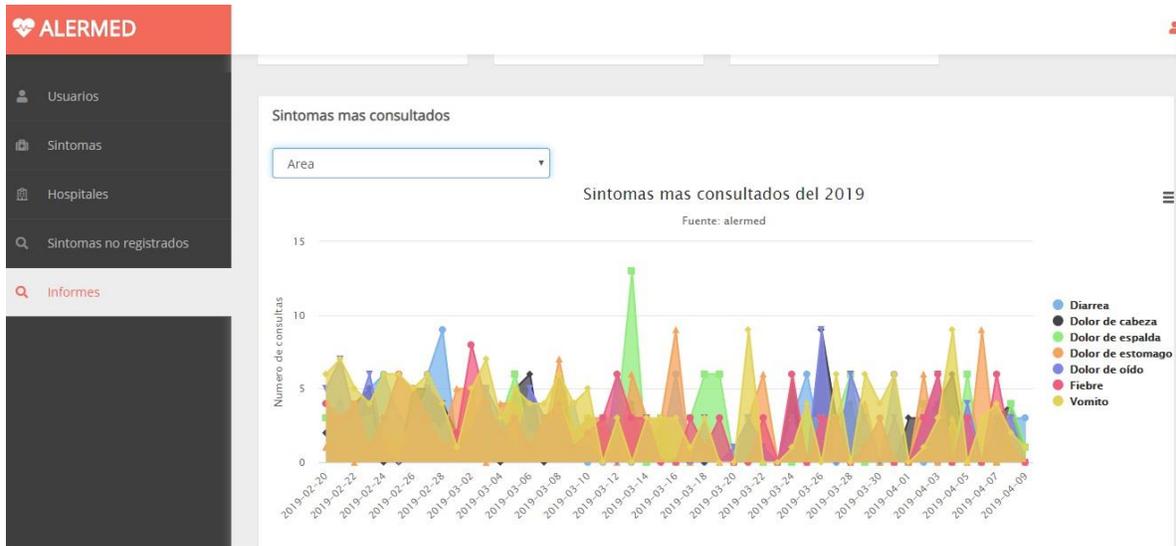
La aplicación puede generar reportes incluyendo gráficos donde se pueden evidenciar datos estadísticos de los motivos de consulta más recurrentes y el centro médico más consultado por los usuarios dando así soporte a los resultados deseados.

Gráfica 13. Reporte alermed síntomas más consultados mes a mes



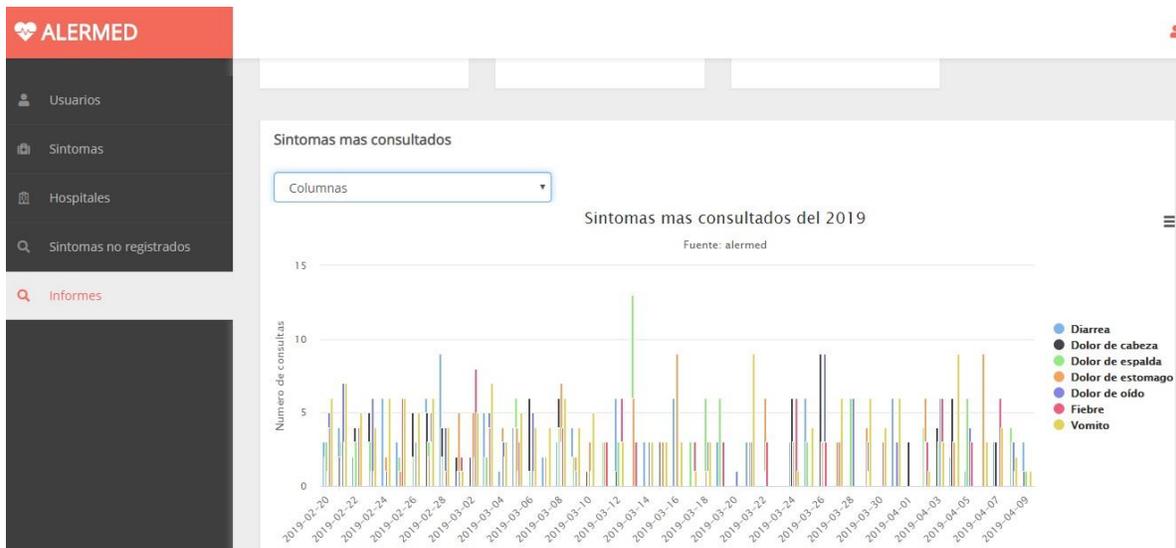
Fuente [Propia]

Gráfica 14. Reporte alermed síntomas más consultados mes a mes



Fuente [Propia]

Gráfica 15. Reporte alermed síntomas más consultados mes a mes



Fuente [Propia]

CONCLUSIONES

Se pudo evidenciar que las personas no conocen a fondo el sistema de salud, no conocen la clasificación TRIAGE ni tampoco conocen los niveles de complejidad de los centros hospitalarios entre otras muchas cosas en las que se encuentran desinformados y es ahí donde podemos dar una solución práctica de tipo tecnológico, disponible para todo aquel que cuente con un dispositivo que tenga consigo un navegador instalado, se realiza de esta manera con el fin de estar disponible para el mayor número de usuarios.

Se ejecutó un análisis en base a los motivos de consulta de los centros hospitalarios que prestan el servicio de urgencia en esta ciudad esto con el fin de centrarnos en los padecimientos más comunes de esta zona y así darles un poco más de relevancia frente a otras consultas que se llevan a cabo con menor frecuencia.

Las metodologías ágiles fueron fundamentales para el desarrollo de la plataforma ya que permite tener avances continuos y además retroalimentar el trabajo en equipo.

TRABAJO FUTURO.

Escalar este desarrollo hacia otras ciudades para beneficiar a la mayor población posible y ¿porque no? beneficiar al país, todo el proceso se puede realizar en otras áreas geográficas haciendo claro los estudios previos de datos basados en la epidemiología registrada en los hospitales y clínicas.

Bibliografía:

[1] Ministerio de Salud y Protección Social

“Ley 100 de 1993”

[Online] Disponible en:

<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/ley-100-de-1993.pdf>

[2] scielo.org.co (25 febrero 2013)

“Sistema de salud en Colombia: una revisión sistemática de literatura**”

[Online] Disponible en:

<http://www.scielo.org.co/pdf/rgps/v12n24/v12n24a05.pdf>

[3] Ministerio de Salud y Protección Social (2012)

“Motivos de los reclamos interpuestos por los usuarios del sistema de salud colombiano ante la Superintendencia Nacional de Salud en el primer semestre del año 2012”

[Online] Disponible en:

<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/IA/SSA/Motivos-de-reclamos-interpuestos-po-usuarios-sistema-salud.pdf>

[4] Hospital San José Popayán E.S.E. (2016)

“Informe de seguimiento de quejas, reclamos, sugerencias y felicitaciones tercer cuatrimestre 2016”

[Online] Disponible en:

<http://www.hospitalsanjose.gov.co/index.php/contratacion/invitaciones-publicas/item/413-informe-de-seguimiento-tercer-cuatrimestre-del-2016>

[5] Ministerio de Salud y Protección Social

“Oferta y demanda del recurso médico en Colombia”

[Online] Disponible en:

https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/COM/Enlace_MinSalud_34.pdf

[6] Razón Pública

“¿Hay déficit de especialistas médicos en Colombia?”

[Online] Disponible en:

<https://www.razonpublica.com/index.php/economia-y-sociedad/8821-¿hay-d%C3%A9ficit-de-especialistas-m%C3%A9dicos-en-colombia.html>

[7] El Tiempo (25 de octubre 2012)

“¿Por qué se muere la gente esperando ser atendida en urgencias?”

[Online] Disponible en:

<http://www.eltiempo.com/archivo/documento/CMS-12335402>

[8] Ministerio de Salud y Protección Social (24 de diciembre de 2015)

“Resolución 00005596 de 2015”

[Online] Disponible en:

<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/resolucion-5596-de-2015.pdf>

[9] “Triage”, Ministerio de salud y protección social, Resolución 5596 del 24 de diciembre del 2015 [En línea]. Disponible en:

<https://www.minsalud.gov.co/salud/PServicios/Paginas/triage.aspx> .

[Accedido: 15-01-2019]

[10] Yuly A, Martin O, Blanca M y Chávez G. (8 agosto 2013)

“Inconsistencias en la afiliación al sistema de salud colombiano: barrera administrativa para la atención integral de las urgencias médicas”

[Online] Disponible en:

<http://aprendeenlinea.udea.edu.co/revistas/index.php/fnsp/article/view/13241/20779779>

[11]Ministerio de Salud y Protección Social

“Resolución 5261 de 1994”

[Online] Disponible en:

https://www.minsalud.gov.co/Normatividad_Nuevo/RESOLUCi%C3%93N%205261%20DE%201994.pdf

[12] J. H Canos y M. C Penales Patricio (2012)

“Metodologías Ágiles en el desarrollo software”

[Online] Disponible en:

<http://roa.ult.edu.cu/handle/123456789/476>

[13] (2018-noviembre-09)

Php

[Online] Disponible en:

<http://php.net/>

[14] J.J Gutiérrez (2014)

“¿qué es un framework web?”

[Online] Disponible en:

http://www.lsi.us.es/~javier/investigacion_ficheros/Framework.pdf

[15] HostaliaWhitePappers (2016)

Laravel, un framework de php?”

[Online] Disponible en:

<https://pressroom.hostalia.com/white-papers/laravel-framework-php>

[16] “laravel-el framework php”, Página oficial de laravel [En línea]. Disponible en:

<https://laravel.com/>

[Accedido: 15-10-2018]

[17] J. Sánchez (2004)

“Mysql guía rápida (versión Windows)”

[Online] Disponible en:

<http://www.cartagena99.com/recursos/programacion/apuntes/mysql.pdf>

[18] A. Rosado Gómez, A. Quintero Duarte y C. D. Meneses Guevara (agosto 12 2012)

“Desarrollo ágil de software aplicando programación extrema”

[Online] Disponible en:

<http://revistas.ufps.edu.co/index.php/ringenio/article/view/23/10>

[19] O.A Pérez (junio 2011)

“Cuatro enfoques metodológicos para el desarrollo de software RUP-MSF-XP-SCRUM”

[Online] Disponible en:

<http://biblioteca.uniminuto.edu/ojs/index.php/Inventum/article/view/9>

[20] J. Fernández Gonzales

“Introducción a las metodologías Ágiles – otras formas de analizar y desarrollar”

[Online] Disponible en:

[https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingeneria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_\(Modulo_3\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnicas_avanzadas_de_ingeneria_de_software/Tecnicas_avanzadas_de_ingenieria_de_software_(Modulo_3).pdf)

[21] J.M. Fernández y S. Cardelli (2014)

“Convivencia de metodologías: Scrum y RUP en un proyecto de gran escala”

[Online] Disponible en:

<http://sedici.unlp.edu.ar/handle/10915/47082>

[22] M. Trigas Gallego

“Metodología scrum”

[Online] Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

[23] “Metodología SCRUM”, publicado 22/dic/2016 [En línea]. Disponible en:

<https://laravel.com/>

[Accedido: 20-10-2018]

[24] S. G. Rivadeneira Molina

“Metodologías Ágiles enfocadas al modelado de requerimientos”

[Online] Disponible en:

<http://ict.unpa.edu.ar/journal/index.php/ICTUNPA/article/view/ICT-UNPA-57-2013/57>

[25] revista softeng

[Online] Disponible en:

<https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>

[26] Re-inventa, innovación y mejora continua

Consultado 26/09/2018

[Online] Disponible en:

<http://www.re-inventa.com/agile-inception-una-buena-practica-en-la-iniciacion-y-reorientacion-de-proyectos-y-productos/>

[27] Y. Fernández Romero, Y. Díaz Gonzales (2012)

“Patrón Modelo – Vista - Controlador”

[Online] Disponible en:

<http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10>

[28] Atlassian bitbucket

“Bitbucket”

[Online] Disponible en:

<https://es.atlassian.com/software/bitbucket>

[29] GitHub

“GitHub”

[Online] Disponible en:

<https://github.com/>

[30] Sqlmanager.net

“Sql Manager para MySQL”

[Online] Disponible en:

<https://www.sqlmanager.net/de/products/mysql/manager>

[31] trayendo mysql a la web

“PhpMyadmin”

[Online] Disponible en:

<https://www.phpmyadmin.net/>

[32] Xampp apache + Mariadb + Php + Perl

“Xampp”

[Online] Disponible en:

<https://www.apachefriends.org/es/index.html>

[33] S. Agudo (2015-04-24)

“Cmder, un útil emulador de terminal portable para Windows”

[Online] Disponible en:

<https://rootear.com/windows/cmdr>

[34] “Que es composer y cómo usarlo”, publicado 11/nov/2014 [En línea]. Disponible en: <https://styde.net/que-es-composer-y-como-usarlo/>

[Accedido: 10-11-2018]

[35] “Planning Poker – ¿Cómo planificar un sprint en SCRUM?”, Publicado 9/oct/2014[En línea]. Disponible en: <https://albertoromeu.com/scrum-planning-poker/>

[Accedido: 15-11-2018]

[36] J.L Lee Razuri (2017)

“Ejercicios de retrospectivas Ágiles”

[Online] Disponible en:

<https://joseleerazuri.blog/2017/01/23/ejercicios-de-retrospectivas-Ágiles/>

[37] Secretaria de salud

“Morbilidad 2011 - 2018” (25-septiembre-2018)

[Online] Disponible en:

<https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/resolucion-5596-de-2015.pdf>

[38] Universidad de salamanca (22y23 de octubre 2007).

“Evaluación y Diagnostico del Dolor – Reunión de Expertos”

http://www.catedradeldolor.com/PDFs/Docencia_Expertos/2007%20Evaluacion%20del%20Dolor.pdf#page=16

[39] K. Kniberg

“Scrum y XP desde las trincheras” (libro)

[Online] Disponible en:

<https://s3.amazonaws.com/academia.edu.documents/39710556/scrum-y-xp-desde-las-trincheras.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1537375381&Signature=F235OIGcazaCfLdUDxyv1O%2BwFms%3D&response-content-disposition=inline%3B%20filename%3DScrum-y-xp-desde-las-trincheras.pdf>

[40] E. Marini (octubre 2012)

“El modelo Cliente/Servidor”

[Online] Disponible en:

<https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>