Revisión de las actividades del marco de trabajo scrum para adaptar aquellas que sean susceptibles de ahorro energético de tal manera que se obtenga un desarrollo amigable con el medio ambiente en un proceso de factoría.

Mónica Alejandra Vargas Jaramillo

Unidad Central del Valle del Cauca
Facultad de Ingeniería
Ingeniería de Sistemas
2021

Revisión de las actividades del marco de trabajo scrum para adaptar aquellas que sean susceptibles de ahorro energético de tal manera que se obtenga un desarrollo amigable con el medio ambiente en un proceso de factoría.

Mónica Alejandra Vargas Jaramillo

Trabajo de grado presentado para optar al título de Ingeniera de Sistemas

Director
José Gabriel Pérez Canencio
Ingeniero de Sistemas

Unidad Central del Valle del Cauca
Facultad de Ingeniería
Ingeniería de Sistemas
2021

Dedicatoria

Este trabajo de grado es el esfuerzo conjunto de muchas personas, por tanto, a todas ellas por su apoyo y amabilidad, les dedico este gran logro de mi vida.

A mi madre y hermana, María Fernanda y Natalia, quienes en los momentos difíciles siempre son una luz que alumbra el camino. A mis abuelos maternos, Marlene y Bernardo, por su cariño, apoyo y enorme interés en mí y mi éxito académico y profesional, muchas gracias. Igualmente, a toda mi familia que, con su esfuerzo, cariño y fortaleza, me han permitido convertirme en la persona que soy.

A Mauricio, por ser una parte muy importante de mi vida, agradezco tu apoyo y compresión en todo momento. Gracias por todo este tiempo. A mis amigos, que saben quiénes son, por su apoyo, por su amistad, por sus risas y por la alegría que siempre han compartido conmigo, muchas gracias.

A todas aquellas personas que creyeron en mí y aún hoy lo siguen haciendo, muchas gracias.

Finalmente, dedico el fruto de este trabajo a Dios, porque siempre he sentido su apoyo y sin él, nada es posible.

Agradecimientos

Quisiera agradecer a mi director de trabajo de grado, José Gabriel Pérez Canencio, por su apoyo incondicional, entusiasmo, guía, paciencia y perseverancia durante la elaboración de este documento. Lo reconozco como un gran educador y una excelente persona. Así mismo quiero expresar mi reconocimiento a todos mis profesores por sus enseñanzas, experiencias, apoyo y dedicación durante mis años de estudio, los cuales me forjan hoy en día como una gran profesional.

Igualmente, deseo expresar mi gratitud a Colombia Software, una empresa que me acogió para iniciar a ejercer mi profesión y en la que aún me encuentro actualmente. Agradezco particularmente a mi jefe inmediato Diego Fernando Pinillo, quién siempre me brindó su apoyo y me animó a culminar esta etapa tan importante. De igual manera agradezco a la gerente Claudia Fernanda Hernández y al señor Carlos Grajales por permitirme el acceso a la información necesaria y por el apoyo incondicional para lograr este proyecto.

Por último, agradezco a mis compañeros, quienes hicieron parte de este camino e hicieron posible la ejecución de este proyecto con su entrega, conocimiento y dedicación.

Resumen

El objetivo principal de este trabajo es describir la adaptación de la metodología SCRUM que se realizó para el desarrollo de una aplicación móvil, donde el equipo desarrollador posee conocimiento del marco de trabajo.

Se diseña y se desarrolla un plan de trabajo, adaptando cada fase de la metodología a las necesidades del proyecto, manteniendo la calidad del producto final y la satisfacción del usuario.

Se concluye con un análisis frente a los ajustes realizados dentro de cada fase del marco de trabajo, resaltando sus ventajas y la gran capacidad de adaptación según el tipo de proyecto y el equipo de trabajo disponible.

En aquellos eventos del marco de trabajo SCRUM en los cuales se adaptó el proceso de desarrollo se logró disminuir los tiempos en algunas actividades. El tiempo disminuido representa ahorro en esfuerzos del personal y en los recursos que se emplean en aquellas actividades. Estos recursos que se ahorran se pueden expresar en términos de energía eléctrica ahorrada. Estos datos permiten calcular la huella de carbono de alcance 2 correspondiente a la energía que fue posible disminuir. De esta manera, se logra contribuir al medio ambiente a partir de la eficiencia energética.

Presentación

Trabajar en la industria del software es tan interesante como disfrutar de un buen viaje, claro, con todas sus diferencias, pero, en síntesis, son dos actividades que se prestan para hacer volar la imaginación y el talento en pro de un gran objetivo.

Para que ambas actividades tengan un buen final, es necesario planear con anticipación todos los pasos a seguir. Para el caso que nos ocupa en este proyecto, la industria del software cuenta con diversas formas de implementar aplicaciones, unas las conocemos como metodologías, otras como métodos y otras como marcos de trabajo, que nos proporcionan los elementos y el orden necesario para desarrollar con éxito productos de calidad.

A propósito de lo anterior, se ha decidido trabajar este proyecto guiados por el marco de trabajo scrum con el propósito de ejecutar un proceso gestionado para reducir la complejidad en algunos de las fases del desarrollo sin afectar el producto final. La idea central es analizar paso a paso, en la medida que se avanza en la implementación, para decidir cuales esfuerzos se pueden ahorrar para llegar al mismo fin con menos trabajo.

En este propósito buscamos ser fieles a la filosofía de los procesos ágiles distribuyendo en forma adecuada los esfuerzos con iteraciones rápidas cuidando la calidad en cada paso con tiempos justos en cada ejecución.

Somos conocedores de la importancia de ahorrar esfuerzos que se traduzcan en eficiencia en términos de consumos energéticos presentes en todos los subprocesos de desarrollo de software.

Mencionamos la eficiencia energética porque es preocupación constante que las tecnologías de la información y las comunicaciones TIC – son responsables del 2% de las emisiones de gases efecto invernadero a nivel mundial, según el informe SMART2020 de Climate Group, (Climate Group, 2020) entonces, nos corresponde hacer los esfuerzos necesarios para disminuir esta cifra y en especial aportar para que las acciones en pro del medio ambiente estén presentes en las actividades diarias con las cuales producimos aplicaciones.

Teniendo en cuenta que en todo el proceso de desarrollo de software existe la necesidad de utilizar dispositivos tecnológicos que consumen energía eléctrica y además que este servicio público es esencial para la vida diaria, entonces, es importante buscar aquellos procesos en los cuales sea posible disminuir la cantidad de energía eléctrica consumida. En las etapas de desarrollo del producto software, es posible distinguir cada subproducto y cada sub proceso que se va obteniendo, desde el inicio de la producción hasta que finalice su ciclo de vida cuando se reemplaza por versiones actualizadas o se desecha. Estas etapas de la producción del software la podemos analizar desde una mirada del análisis del ciclo de vida del producto, en adelante ACV, teniendo en cuenta el impacto ambiental que se genera a lo largo de todo el ciclo de factoría.

Todo lo anterior conduce a la realización de un proyecto que conlleva al desarrollo de una aplicación móvil guiados por el marco de trabajo scrum, pero, con una particularidad: se han ajustado los tiempos de trabajo que en teoría deberían emplearse en algunas de las fases de la implementación manteniendo la calidad del producto final y la satisfacción del usuario. Con esto se hace un ahorro de esfuerzos que es suficiente para disminuir la cantidad de energía requerida en los procesos lo cual se traduce en disminución de emisiones de gases efecto invernadero y un gran beneficio para el planeta en todo el ACV del producto resultante.

Esta práctica de eficiencia energética en un proceso de desarrollo de software se asimila a las buenas prácticas tecnológicas verdes consideradas en el tema de Green It tan importante en este momento histórico para el planeta y para la sostenibilidad de las generaciones futuras.

José Gabriel Pérez Canencio
Docente Facultad de Ingeniería Uceva
Mónica Alejandra Vargas Jaramillo
Estudiante programa Ingeniería de Sistemas. Uceva

Tabla de Contenidos

Capítulo 1. Generalidades del proyecto 1
1.1 El Problema 1
1.1.1 Descripción 1
1.1.2 Formulación 3
1.2 Justificación 4
1.3 Objetivos 6
1.3.1 General 6
1.3.2 Específicos 6
1.4 Alcance 7
Capítulo 2. Marcos de referencia 8
2.1 Marco teórico 8
2.1.1 Marco de trabajo Scrum. 8
2.1.2. Metodologías ágiles. 17
2.1.3 Eficiencia energética. 21
2.1.4. Estado del arte 24
Capítulo 3. Metodología de la investigación aplicada al proyecto 29
3.1 Tipo de investigación. 29
3.1.1 Investigación descriptiva. 29
3.1.2 Enfoque de la investigación. Enfoque cualitativo. 29
Capítulo 4. Ingeniería del proyecto. 32
4.1. Situación actual de la empresa: 32
4.2. Proceso de desarrollo basado en SCRUM 33
4.2.1 FASE 1: Creación del equipo Scrum 33
4.2.2 FASE 2: Planeación de sprints, eventos y artefactos de Scrum 35
4.3 Arquitectura del sistema 44
4.3.1 Componentes y patrón de diseño 46
4.3.2 Modelo de la base de datos 47
4.4 Diseño de las interfaces gráficas de usuario final 48
4.4.1 Diseño. Las siguientes ilustraciones muestran las interfaces gráficas de usuario final 48
4.4.2 Descripción general de las interfaces gráficas de usuario. GUI´s 54
Capítulo 5 55
5.1 Conclusiones 55
5.1.1 Generalidades en la gestión de proyectos 55
5.1.2 Resultado del trabajo con Scrum 55
5.1.3 Como adaptar Scrum a un proceso de desarrollo en la industria del software. 57
5.1.4 Procesos en los cuales se ahorra tiempo y esfuerzos 59
5.2 Recomendaciones. 62

5.2.1 Porqué recomendamos scrum. 62
5.3 Trabajo futuro. 63
6. Referencias consultadas 64

Lista de tablas

Tabla 1. Matriz bibliográfica 25
Tabla 2. Matriz analítica de contenido por tipo de documento..... 26
Tabla 3. Matriz analítica de contenido por país..... 27
Tabla 4. Cálculo de tiempos Sprint 1..... 36
Tabla 5. Cálculo de tiempos Sprint 2..... 38
Tabla 6. Cálculo de tiempos sprint 3 40
Tabla 7. Cálculo de tiempos Sprint 4..... 42
Tabla 8. Tiempo estimado por la metodología 59
Tabla 9. Tiempos reales empleados en el proyecto 59
Tabla 10. Dispositivos consumidores 60
Tabla 11. Resumen de cálculos..... 61
Tabla 12. Scrum Vs Gestión de proyectos tradicional..... 62

Lista de Ilustraciones

Ilustración 1. Marco de trabajo Scrum 9
Ilustración 2. Diseño metodológico del proyecto 30
Ilustración 3. Diagrama de arquitectura 45
Ilustración 4. Patrón de diseño Factory 46
Ilustración 5. Diagrama de componentes 46
Ilustración 6. Modelo de la base de datos 47
Ilustración 7. Interfaz gráfica 1 49
Ilustración 8. Interfaz gráfica 2 49
Ilustración 9. Interfaz gráfica 3 50
Ilustración 10. Interfaz gráfica 4 50
Ilustración 11. Interfaz gráfica 5 51
Ilustración 12. Interfaz gráfica 6 51
Ilustración 13. Interfaz gráfica 7 52
Ilustración 14. Interfaz gráfica 8 52
Ilustración 15. Interfaz gráfica 10 53
Ilustración 16. Interfaz gráfica 9 53
Ilustración 17. Adaptación del proceso Scrum 58

Lista de anexos

Anexo 1. Base de datos para el estado del arte (hoja electrónica Excel). 27
Anexo 2. Product Backlog. Historias de usuario. 36
Anexo 3. Pruebas unitarias 43

Capítulo 1. Generalidades del proyecto

1.1 El Problema

1.1.1 Descripción

En el mundo entero nos movemos en entornos cambiantes, ya no vivimos una época de cambios sino un cambio de época, una nueva revolución tanto a nivel industrial como tecnológica. La información cada vez está más accesible a todas las personas y las opciones para comunicarnos aumentan a través del tiempo y las circunstancias. Esto entorno altamente competitivo, provoca que aspectos como la satisfacción del cliente sea un reto más grande para las empresas. Además, los clientes están cada vez más informados y su nivel de exigencia es mayor.

La experiencia del cliente es el factor más importante que se encuentra en la mayoría de empresas ya que el cliente actual no se conforma con un buen producto/precio, el cliente compra una emoción**,** una experiencia**,** sensaciones**;** los clientes cuando compran, están en contacto con sus emociones, son las precursoras de cualquier toma de decisión racional que el cliente toma en pocos segundos y tal como comenta Mónica Deza en uno de sus artículos de Neuromarketing "sin emoción, no hay atención; sin atención, no hay memoria, y lo que no está registrado en el disco duro de las personas no existe"[1].

Colombia software es una casa de software especializada en el desarrollo de soluciones y herramientas avanzadas de gestión para empresas del sector transporte – logística, aporta soluciones en la gestión de la cadena de suministro completas para el cliente final, es por ello que su visión se basa en construir relaciones cercanas con los clientes, brindando soluciones tecnológicas que les faciliten sus operaciones para ser más eficaces y eficientes, siendo un aspecto muy importante la tranquilidad que se pueda brindar a sus clientes y a su vez a los clientes de ellos. El producto principal de esta casa de software se llama "Silogtran", el cual, es

---

1 Fuente: Sistemas de Información. Tomada de: http://yeux.com.mx/ColumnaUniversitaria/neuromarketing-ciencia-y-marketing/

un sistema web que permite gestionar todo el ciclo del transporte de mercancía y logística, la cual comprende procesos como despacho, financiero, mantenimiento, inventario y contabilidad.

Para Colombia Software es de gran importancia el bienestar y la satisfacción de sus clientes, es por ello que cada vez se brindan soluciones de acuerdo al avance tecnológico que se va presentando en la actualidad, pero la mayoría de estas soluciones son para uso exclusivo del personal operativo que conforman estos procesos y el alcance al usuario final del cliente que a su vez es su cliente, es muy limitado. Esto hace que la información que es importante para estos usuarios finales, no llegue de manera adecuada y afecte la satisfacción y estabilidad del cliente principal ya que incrementa la gestión que estos deben hacer para satisfacerlos.

Por esto, es importante que se desarrolle un producto que permita consolidar la información de los diferentes procesos que se cumplen en el transporte de mercancía para que sea accesible por los usuarios de los clientes de Colombia Software, donde les permita hacer un seguimiento más cercano y preciso de su mercancía, en cuanto al detalle del viaje, puestos de control, novedades, visualizar, descargar facturas y remesas generadas y obtener reportes que les permita generar la información en un documento para uso interno.

Para el desarrollo de esta aplicación es importante implementar una metodología que permita al equipo de desarrollo adaptarse de forma natural a las condiciones cambiantes y a los requisitos del cliente, para ello se decidió trabajar con la metodología SCRUM el cual es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto.

De acuerdo (Shcwaber & Sutherland, 2020), scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Scrum se compone de eventos con los cuales se aborda el proceso de desarrollo.
Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para habilitar la transparencia requerida.

No operar cualquier evento según lo prescrito resulta en la pérdida de oportunidades para inspeccionar y adaptarse. Los eventos se utilizan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Lo óptimo es que todos los eventos se celebren al mismo tiempo y en el mismo lugar para reducir la complejidad.

Por lo anterior se hace importante revisar cada evento y analizar las oportunidades para adaptarlos y reducir en lo posible el consumo de recursos que sean susceptibles de optimizar.

1.1.2 Formulación

¿Cómo adaptar algunos eventos del marco de trabajo scrum para optimizar el uso de recursos en el proceso de desarrollo de la aplicación móvil para el seguimiento de la trazabilidad de la operación en los clientes de Unión Andina en la empresa Colombia Software?

1.2 Justificación

Silogtran es un sistema web multiplataforma que permite gestionar cada proceso que compone un despacho de mercancía masiva o paqueteo. Este proceso inicia con la negociación con el cliente frente a las recogidas periódicas la cual se plantea en el software en un documento llamado “Plan de recogida”, posterior, se genera una solicitud de transporte donde se consolida la información inicial del posible despacho, detallando el cliente, el remitente, el destinatario, el tipo de tarifa y flete a utilizar. Una vez se apruebe dicho documento, se procede a crear el despacho, confirmando dicha solicitud de transporte donde se define el vehículo encargado del viaje y se generan los documentos que soportan dicho viaje, estos documentos se denominan “manifiesto” y “remesa”, donde el primero sustenta la información propia del viaje, como el vehículo, la ruta y el flete; por otra parte, la remesa soporta la mercancía que se está transportando, indicando información relevante como el cliente, el destinatario, el tipo de mercancía y la tarifa aplicada.

Durante el viaje se lleva registro del recorrido del vehículo mediante el módulo de tráfico, donde se reportan los puestos de control y las novedades que se puedan generar. Al finalizar el viaje y la entrega de la mercancía, se registra dicha información en el módulo de cumplido manifiesto, donde se especifican las fechas y horas de cada entrega, además de los kilómetros recorridos y los gastos por parte del conductor.

Una vez la parte operativa ha finalizado, se inicia el proceso de facturación de dichas remesas gestionadas y la legalización del viaje, donde se verifica los gastos realizados por el conductor. El sistema también permite subir documentos e imágenes que soportan ciertos procesos hacia el cliente, los cuales se comunican mediante correos electrónicos a los clientes finales cuando el viaje ha culminado.

Por ello es de gran importancia desarrollar una herramienta donde estos clientes los cuales son los interesados del manejo de su mercancía, puedan verificar de manera más real el proceso que se realiza y la trazabilidad del mismo, agregando inmediatez e independencia al acceso de dicha información, implementando el marco de trabajo scrum para su desarrollo.

Implementar un proceso de factoría bajo el marco de trabajo scrum representa para Colombia Software la posibilidad de entregar al cliente un desarrollo de alta calidad, sin embargo, se hace importante incluir los aspectos relacionados con el cuidado del medio ambiente desde las etapas del proceso de construcción del producto y el ACV.

Teniendo en cuenta que estos procesos de factoría hacen parte de los mecanismos de desarrollo que incluyen tecnologías de información, TI y de acuerdo con (Climate Group, 2020), las TI son responsables del 2% de las emisiones de gases efecto invernadero del mundo, se hace imprescindible atender estas cifras y buscar alternativas que conlleven a la disminución de GEI mediante estrategias tales como la eficiencia energética. Esto obedece a la gran importancia que tiene en Colombia el ahorro energético que fue anunciado por (ANDESCO, 2019) afirmando que la ineficiencia energética le cuesta a Colombia 21 billones de pesos anuales.

En consideración a lo anterior hemos decidido incluir en la aplicación del marco de trabajo scrum una fase importante que relaciona el consumo energético de cada etapa con el fin de calcular los ahorros que se puedan lograr para reducir emisiones de gases efecto invernadero de tal forma que se logre la optimización de algunos eventos y se contribuya con un impacto positivo al medio ambiente

1.3 Objetivos

1.3.1 General

Adaptar eventos del marco de trabajo scrum con el propósito de contribuir a un desarrollo más eficiente durante el proceso de construcción del proyecto aplicación móvil para el seguimiento de la trazabilidad de la operación en los clientes de Unión Andina en la empresa Colombia Software.

1.3.2 Específicos

- Revisar las actividades de cada iteración de la metodología scrum.
- Analizar la situación actual de la empresa de factoría Colombia Software en cuanto a los recursos destinados para el desarrollo de la aplicación.
- Formular un proceso de adaptación de las actividades de scrum para el desarrollo de la aplicación de acuerdo con los recursos, disponibilidades del personal y tiempos de desarrollo de Colombia Software.

1.4 Alcance

La aplicación incorpora las siguientes funcionalidades:

- Trazabilidad de cada viaje en diferentes estados.
- Seguimiento de puestos de control.
- Consulta de novedades de tráfico.
- Consulta de facturas realizadas.
- Consulta de remesas facturadas.
- Generación del formato de impresión de remesas.
- Módulo para generación de reportes.

- Análisis de los eventos de scrum para determinar en cuales es posible la optimización de recursos, especialmente en consumo energético.
- Se realizará una actividad para socializar las experiencias adquiridas en el proceso de desarrollo que permita transferir conocimiento a los estudiantes del programa ingeniería de sistemas de la Uceva.

Capítulo 2. Marcos de referencia

2.1 Marco teórico

En este capítulo se van a tratar bases teóricas de las principales temáticas que fundamentan el objetivo del presente trabajo de grado. Lo anterior se realiza bajo los aportes teóricos de varias fuentes sobre scrum, metodologías ágiles, eficiencia energética y estado del arte.

2.1.1 Marco de trabajo Scrum.

Con el fin de contextualizar a cerca del método para el desarrollo de software que se empleó en este trabajo, se hace referencia a la fuente principal del método llamado scrum y que se ha recuperado del portal principal presentado en la web (Francia Huambachano, 2017) que textualmente dice:

Scrum es un proceso de gestión que reduce la complejidad en el desarrollo de productos para satisfacer las necesidades de los clientes. La gerencia y los equipos de Scrum trabajan juntos alrededor de requisitos y tecnologías para entregar productos funcionando de manera incremental usando el empirismo.

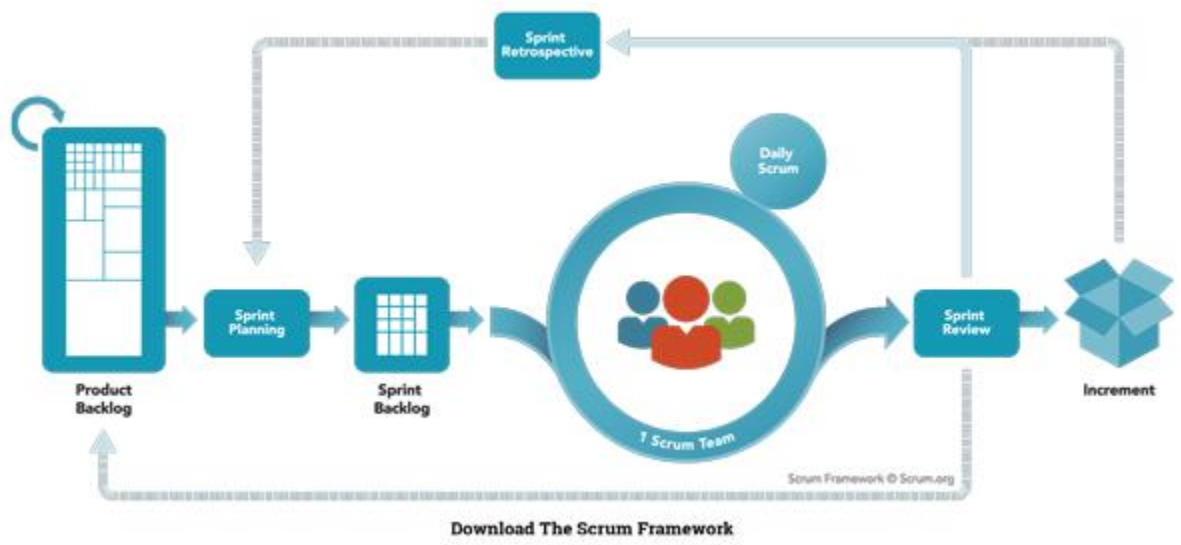
Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum y sus roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum.

De acuerdo (Shcwaber & Sutherland, 2020), scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Scrum es simple, no es una gran colección de partes y componentes obligatorios definidos de manera prescriptiva. Scrum no es una metodología, Scrum está basado en un modelo de proceso empírico con respeto a las personas y basado en la auto organización de los equipos para lidiar con lo imprevisible y resolver problemas complejos inspeccionando y adaptando continuamente. El siguiente gráfico representa Scrum como lo describen Ken Schwaber y Jeff Sutherland

Ilustración 1. Marco de trabajo Scrum



2.1.1.1 Componentes de scrum.

Scrum Team. La unidad fundamental de Scrum es un pequeño equipo de personas, un Scrum Team. El Scrum team consta de un scrum master, un product owner y developers.

Dentro de un Scrum Team, no hay sub equipos ni jerarquías. Es una unidad cohesionada de profesionales enfocados en un objetivo a la vez: el Objetivo del Producto.

Los Scrum Teams son multifuncionales, lo que significa que los miembros tienen todas las habilidades necesarias para crear valor en cada Sprint. También se auto gestionan, lo que significa que deciden internamente quién hace qué, cuándo y cómo.

El Scrum Team es lo suficientemente pequeño como para seguir siendo ágil y lo suficientemente grande como para completar un trabajo significativo dentro de un Sprint, generalmente 10 personas o menos. En general, los equipos más pequeños se comunican mejor y son más productivos. Si los Scrum Teams se vuelven demasiado grandes, deberían considerar reorganizarse en múltiples Scrum Teams cohesivos, cada uno enfocado en el mismo producto. Por lo tanto, deben compartir el mismo objetivo del producto, el product backlog y el product owner.

El Scrum Team es responsable de todas las actividades relacionadas con el producto, desde la colaboración de los interesados, la verificación, el mantenimiento, la operación, la experimentación, la investigación y el desarrollo, y cualquier otra cosa que pueda ser necesaria.

Están estructurados y empoderados por la organización para gestionar su propio trabajo.

Trabajar en Sprints a un ritmo sostenible mejora el enfoque y la consistencia del Scrum Team.

Todo el Scrum Team es responsable de crear un incremento valioso y útil en cada Sprint. Scrum define tres responsabilidades específicas dentro del Scrum Team: los Developers, el Product Owner y el Scrum Master.

Developers. Las personas del Scrum team que se comprometen a crear cualquier aspecto de un Incremento utilizable en cada Sprint son Developers. Las habilidades específicas que necesitan los Developers suelen ser amplias y variarán según el ámbito de trabajo.

Sin embargo, los Developers siempre son responsables de:

- Crear un plan para el Sprint, el Sprint Backlog
- Inculcar calidad al adherirse a una definición de terminado
- Adaptar su plan cada día hacia el objetivo del Sprint; y
- Responsabilizarse mutuamente como profesionales.

Product Owner. El Product Owner es responsable de maximizar el valor del producto resultante del trabajo del Scrum Team. La forma en que esto se hace puede variar ampliamente entre organizaciones, Scrum Teams e individuos.

El Product Owner también es responsable de la gestión efectiva del Product Backlog, lo que incluye:

● Desarrollar y comunicar explícitamente el Objetivo del Producto
● Crear y comunicar claramente los elementos del Product Backlog
● Ordenar los elementos del Product Backlog; y,
● Asegurarse de que el Product Backlog sea transparente, visible y se entienda.

El Product Owner puede realizar el trabajo anterior o puede delegar la responsabilidad en otros. Independientemente de ello, el Product Owner sigue siendo el responsable de que el trabajo se realice.

Para que los Product Owners tengan éxito, toda la organización debe respetar sus decisiones. Estas decisiones son visibles en el contenido y el orden del Product Backlog, y a través del Incremento inspeccionable en la Sprint Review.

El Product Owner es una persona, no un comité. El Product Owner puede representar las necesidades de muchos interesados en el Product Backlog. Aquellos que quieran cambiar el Product Backlog pueden hacerlo intentando convencer al Product Owner.

Scrum Master. El Scrum Master ayuda a todos a comprender la teoría y la práctica de Scrum, tanto dentro del Scrum Team como de la organización.

El Scrum Master es responsable de lograr la efectividad del Scrum Team. Lo hace apoyando al Scrum Team en la mejora de sus prácticas, dentro del marco de trabajo de Scrum.

Los Scrum Masters son verdaderos líderes que sirven al Scrum Team y a la organización en general.

El Scrum Master sirve al Scrum Team de varias maneras, que incluyen:

● Guiar a los miembros del equipo en ser auto gestionados y multifuncionales.
● Ayudar al Scrum Team a enfocarse en crear Incrementos de alto valor que cumplan con la Definición de Terminado.
● Procurar la eliminación de impedimentos para el progreso del Scrum Team; y,
● Asegurarse de que todos los eventos de Scrum se lleven a cabo y sean positivos, productivos y se mantengan dentro de los límites de tiempo recomendados.

El Scrum Master sirve al Product Owner de varias maneras, que incluyen:

● Ayudar a encontrar técnicas para una definición efectiva de Objetivos del Producto y la gestión del Product Backlog.
● Ayudar al Scrum Team a comprender la necesidad de tener elementos del Product Backlog claros y concisos.
● Ayudar a establecer una planificación empírica de productos para un entorno complejo; y,
● Facilitar la colaboración de los interesados según se solicite o necesite.

El Scrum Master sirve a la organización de varias maneras, que incluyen:

● Liderar, capacitar y guiar a la organización en su adopción de Scrum
● Planificar y asesorar implementaciones de Scrum dentro de la organización
● Ayudar a los empleados y los interesados a comprender y aplicar un enfoque empírico para el trabajo complejo; y,
● Eliminar las barreras entre los interesados y los Scrum Team.

Eventos de Scrum. El Sprint es un contenedor para todos los demás eventos.

Cada evento en Scrum es una oportunidad formal para inspeccionar y adaptar los artefactos Scrum. Estos eventos están diseñados específicamente para habilitar la transparencia requerida.

No operar cualquier evento según lo prescrito resulta en la pérdida de oportunidades para inspeccionar y adaptarse. Los eventos se utilizan en Scrum para crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Lo óptimo es que todos los eventos se celebren al mismo tiempo y en el mismo lugar para reducir la complejidad.

El Sprint. Los Sprints son el corazón de Scrum, donde las ideas se convierten en valor. Son eventos de duración fija de un mes o menos para crear consistencia. Un nuevo Sprint comienza inmediatamente después de la conclusión del Sprint anterior.

Todo el trabajo necesario para lograr el Objetivo del Producto, incluido la Sprint Planning, Daily Scrums, Sprint Review y Sprint Retrospective, ocurre dentro de los Sprints.

Durante el Sprint:

- No se realizan cambios que pongan en peligro el Objetivo del Sprint.
- La calidad no disminuye.
- El Product Backlog se refina según sea necesario; y,
- El alcance se puede aclarar y renegociar con el Product Owner a medida que se aprende más.

Los Sprints permiten la previsibilidad al garantizar la inspección y adaptación del progreso hacia un Objetivo del Producto al menos cada mes calendario. Cuando el horizonte de un Sprint es demasiado largo, el Objetivo del Sprint puede volverse inválido, la complejidad puede crecer y el riesgo puede aumentar.

Se pueden emplear Sprints más cortos para generar más ciclos de aprendizaje y limitar el riesgo de costo y esfuerzo a un período de tiempo menor. Cada Sprint puede considerarse un proyecto corto.

Existen varias prácticas para pronosticar el progreso, como el trabajo pendiente (burn-downs), trabajo completado (burn-ups) o flujos acumulativos (cumulative flows). Si bien han demostrado su utilidad, no reemplazan la importancia del empirismo. En entornos complejos, se desconoce lo que sucederá.

Solo lo que ya ha sucedido se puede utilizar para la toma de decisiones con miras al futuro. Un Sprint podría cancelarse si el Objetivo del Sprint se vuelve obsoleto. Solo el Product Owner tiene la autoridad para cancelar el Sprint.

Sprint Planning. La Sprint Planning inicia el Sprint al establecer el trabajo que se realizará para el Sprint. El Scrum Team crea este plan resultante mediante trabajo colaborativo.

El Product Owner se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Product Backlog y cómo se relacionan con el Objetivo del Producto. El Scrum Team también puede invitar a otras personas a asistir a la Sprint Planning para brindar asesoramiento.

La Sprint Planning aborda los siguientes temas:

Tema uno: ¿Por qué es valioso este Sprint? El Product Owner propone cómo el producto podría Incrementar su valor y utilidad en el Sprint actual. Luego, todo el Scrum Team colabora para definir un Objetivo del Sprint que comunica por qué el Sprint es valioso para los interesados. El Objetivo del Sprint debe completarse antes de que termine la Sprint Planning

Tema dos: ¿Qué se puede hacer en este Sprint? A través de una conversación con el Product Owner, los Developers seleccionan elementos del Product Backlog para incluirlos en el Sprint actual. El Scrum Team puede refinar estos elementos durante este proceso, lo que aumenta la comprensión y la confianza. Seleccionar cuánto se puede completar dentro de un Sprint puede ser un desafío. Sin embargo, cuanto más sepan los Developers sobre su desempeño pasado, su capacidad actual y su Definición de Terminado, más confiados estarán en sus pronósticos para el Sprint.

Tema tres: ¿Cómo se realizará el trabajo elegido? Para cada elemento del Product Backlog seleccionado, los Developers planifican el trabajo necesario para crear un Incremento que cumpla con la Definición de Terminado. A menudo, esto se hace descomponiendo los elementos del Product Backlog en elementos de trabajo más pequeños de un día o menos. La forma de

hacerlo queda a criterio exclusivo de los Developers. Nadie más les dice cómo convertir los elementos del Product Backlog en Incrementos de valor.

El Objetivo del Sprint, los elementos del Product Backlog seleccionados para el Sprint, más el plan para entregarlos se denominan juntos Sprint Backlog. La Sprint Planning tiene un límite de tiempo de máximo ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

Daily Scrum. El propósito de la Daily Scrum es inspeccionar el progreso hacia el Objetivo del Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el trabajo planificado entrante.

La Daily Scrum es un evento de 15 minutos para los Developers del Scrum Team. Para reducir la complejidad, se lleva a cabo a la misma hora y en el mismo lugar todos los días hábiles del Sprint. Si el Product Owner o Scrum Master están trabajando activamente en elementos del Sprint Backlog, participan como Developers.

Los Developers pueden seleccionar la estructura y las técnicas que deseen, siempre que su Daily Scrum se centre en el progreso hacia el Objetivo del Sprint y produzca un plan viable para el siguiente día de trabajo. Esto crea enfoque y mejora la autogestión.

Las Daily Scrums mejoran la comunicación, identifican impedimentos, promueven la toma rápida de decisiones y, en consecuencia, eliminan la necesidad de otras reuniones. La Daily Scrum no es el único momento en el que los Developers pueden ajustar su plan. A menudo se reúnen durante el día para discusiones más detalladas sobre cómo adaptar o volver a planificar el resto del trabajo del Sprint.

Sprint Review. El propósito de la Sprint Review es inspeccionar el resultado del Sprint y determinar futuras adaptaciones. El Scrum Team presenta los resultados de su trabajo a los interesados clave y se discute el progreso hacia el Objetivo del Producto. Durante el evento, el Scrum Team y los interesados revisan lo que se logró en el Sprint y lo que ha cambiado en su entorno.

Con base en esta información, los asistentes colaboran sobre qué hacer a continuación. El Product Backlog también se puede ajustar para satisfacer nuevas oportunidades. La Sprint Review es una sesión de trabajo y el Scrum Team debe evitar limitarla a una presentación.

La Sprint Review es el penúltimo evento del Sprint y tiene un límite de tiempo de máximo cuatro horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

Sprint Retrospective. El propósito de la Sprint Retrospective es planificar formas de aumentar la calidad y la efectividad. El Scrum Team inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su definición de terminado. Los elementos inspeccionados suelen variar según el ámbito del trabajo. Se identifican los supuestos que los llevaron por mal camino y se exploran sus orígenes.

El Scrum Team analiza qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas. El Scrum Team identifica los cambios más útiles para mejorar su efectividad. Las mejoras más impactantes se abordan lo antes posible. Incluso se pueden agregar al Sprint Backlog para el próximo Sprint. La Sprint Retrospective concluye el Sprint.

Tiene un tiempo limitado a máximo tres horas para un Sprint de un mes. Para Sprints más cortos, el evento suele ser de menor duración.

Sprint Backlog. El Sprint Backlog se compone del Objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan de acción para entregar el Incremento (cómo).

El Sprint Backlog es un plan realizado por y para los Developers. Es una imagen muy visible y en tiempo real del trabajo que los Developers planean realizar durante el Sprint para lograr el Objetivo del Sprint. En consecuencia, el Sprint Backlog se actualiza a lo largo del Sprint a medida que se aprende más. Debe tener suficientes detalles para que puedan inspeccionar su progreso en la Daily Scrum

Increment. Un Increment o incremento es un peldaño concreto hacia el Objetivo del Producto. Cada Increment se suma a todos los Increments anteriores y se verifica minuciosamente, lo que garantiza que todos los Increments funcionen juntos.

Para proporcionar valor, el Increment debe ser utilizable. Se pueden crear múltiples Increments dentro de un Sprint. La suma de los Increments se presenta en la Sprint Review apoyando así el empirismo. Sin embargo, se puede entregar un Increment a los interesados antes del final del Sprint. La Sprint Review nunca debe considerarse una puerta para liberar valor. El trabajo no puede considerarse parte de un Increment a menos que cumpla con la Definición de Terminado.

Definición de Terminado. La Definición de Terminado es una descripción formal del estado del Incremento cuando cumple con las medidas de calidad requeridas para el producto. En el momento en que un elemento del Product Backlog cumple con la Definición de Terminado, nace un Incremento. La Definición de Terminado crea transparencia al brindar a todos un entendimiento compartido de qué trabajo se completó como parte del Incremento. Si un elemento del Product Backlog no cumple con la Definición de Terminado, no se puede publicar ni presentar en la Sprint Review. En su lugar, vuelve al Product Backlog para su consideración futura. Si la Definición de Terminado para un Incremento es parte de los estándares de la organización, todos los Scrum Teams deben seguirla como mínimo. Si no es un estándar organizacional, el Scrum Team debe crear una Definición de Terminado apropiada para el producto.

Los Developers deben adherirse a la Definición de Terminado. Si hay varios Scrum Teams trabajando juntos en un producto, deben definir y cumplir mutuamente la misma Definición de Terminado.

2.1.2. Metodologías ágiles.

Para abordar el tema de las metodologías ágiles, nos hemos basado en la presentación de los conceptos más importantes que se encuentran publicados por la firma Red Hat en su portal. Aquí se transcriben textualmente estos conceptos. (Red Hat, 2000)

El enfoque ágil para el desarrollo de software busca distribuir de forma permanente sistemas de software en funcionamiento diseñados con iteraciones rápidas.

Sin embargo, la frase "metodología ágil" es engañosa porque implica que el enfoque ágil es la única forma de abordar el desarrollo de software. La metodología ágil no hace referencia a una serie de indicaciones sobre qué hacer exactamente durante el desarrollo de software. Se trata más bien de una forma de pensar en la colaboración y los flujos de trabajo, y define un conjunto de valores que guían nuestras decisiones con respecto a lo que hacemos y a la manera en que lo hacemos.

En concreto, las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo piezas pequeñas de sistemas de software en funcionamiento para mejorar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos auto organizados de desarrolladores de software y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un enfoque sencillo de la documentación de software, y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos.

2.1.2.1 Los valores de la metodología ágil

La metodología ágil como la conocemos en la actualidad nació en el año 2001. En respuesta a los enfoques en cascada de la gestión de proyectos, en que estos se organizan como series de secuencias lineales, un grupo de desarrolladores de software redactó el Manifiesto para el Desarrollo Ágil de Software. En este documento, los programadores propusieron un nuevo enfoque de desarrollo de software y describieron cuatro características fundamentales que se deberían priorizar por encima de otras cuestiones. De acuerdo con lo que establecieron, los equipos de desarrollo ágil de software deberían valorar:

- **Las personas y las interacciones** antes que los procesos y las herramientas
- **El software en funcionamiento** antes que la documentación exhaustiva
- **La colaboración con el cliente** antes que la negociación contractual
- **La respuesta ante el cambio** antes que el apego a un plan

Los autores aclaran que todos los puntos de la lista anterior tienen cierto valor inherente. Sin embargo, proponen que valorar los puntos de la izquierda (en negrita) antes que los de la derecha puede dar lugar a mejores resultados en el desarrollo del producto. El manifiesto no busca imponer un conjunto de prácticas, sino ser una guía que permita pensar en el desarrollo de software de otra manera.

Gracias a este manifiesto, se han obtenido varios resultados prácticos. Por ejemplo, en lugar de desarrollar sistemas de software en una secuencia que va de una fase a la siguiente (que es como el método cascada garantiza la calidad de un producto), el método ágil promueve que los procesos de desarrollo y prueba sean simultáneos y constantes. Dicho de otra forma, en el desarrollo en cascada, una fase debe finalizarse por completo antes de poder pasar a la siguiente; el desarrollo ágil, por otro lado, permite que varias secuencias sucedan al mismo tiempo.

2.1.2.2 ¿Dónde se originó el enfoque ágil?

Los enfoques ágiles de trabajo se crearon para abordar las limitaciones del método en cascada, que se derivaba del sistema de cadena de montaje de 1913 que utilizó Henry Ford como método de fabricación, y que luego se aplicó al desarrollo de software. Desde su origen en 2001, el desarrollo ágil pisa fuerte en el sector de los sistemas de software y la gestión de proyectos, a pesar de sus múltiples variantes.

El enfoque ágil surgió cuando muchos desarrolladores de software notaron que los ciclos de producción y los métodos de colaboración del modelo en cascada no daban los resultados esperados. Para comienzos de la década de los 90, este problema se había generalizado, ya que era común que transcurrieran varios años entre la aparición de una necesidad empresarial justificada y la distribución de una aplicación en funcionamiento. Eran tantos los cambios que podían sufrir las demandas empresariales y los mercados durante esos años, que posiblemente se cancelarían partes importantes de los proyectos de software antes de poder distribuirse. El desperdicio de tiempo y recursos llevó a que varios desarrolladores de software buscaran una alternativa.

Frente a la amenaza de interrupciones, las empresas adoptan cada vez con mayor frecuencia estrategias de transformación digital que permitan seguir sus ritmos acelerados. Sin embargo, cuando esto ocurre, el desarrollo ágil de software suele adquirir un papel protagónico.

El desarrollo ágil constituye las bases de muchos de los flujos de trabajo actuales. El cloud computing, con su infraestructura de TI flexible y escalable, ha crecido en paralelo a las demandas de desarrollo ágil de software. El desarrollo nativo de la nube adopta una noción del software similar a la del enfoque ágil, es decir, funciona como una serie de servicios interconectados que se adaptan para satisfacer las necesidades empresariales.

El concepto de DevOps derriba la antigua barrera que existía entre el desarrollo de software y las operaciones. La SRE es una implementación de DevOps que utiliza el software como una herramienta para gestionar los sistemas y automatizar las tareas operativas. Los métodos de integración y distribución continuas (CI/CD) reconocen que los sistemas de software cambian permanentemente, y proporcionan a los desarrolladores herramientas para acelerar la implementación de un código nuevo.

Posiblemente haya notado que el concepto de "metodología ágil" es una idea ágil en sí misma, porque responde a las necesidades de sus clientes (es decir, los desarrolladores de software) durante los tiempos de cambio. Es importante que tenga esto en cuenta mientras damos un vistazo rápido a las opciones de marcos ágiles, que se conocen con diferentes nombres y que, generalmente, varían de una implementación a otra.

2.1.2.3 Marcos ágiles

Los marcos ágiles para el desarrollo de software (como Scrum, kanban o programación extrema [XP]) son la base de procesos conocidos de desarrollo de software, como DevOps y la integración continua/implementación continua (CI/CD).

Scrum es probablemente el marco ágil que más se utiliza en la actualidad. Sin embargo, no es la única alternativa dentro del mundo ágil y, para ser sinceros, no todos sus aspectos pueden caracterizarse como ágiles. Este marco de gestión de trabajos se diseñó para equipos pequeños e interdisciplinarios de entre 5 y 9 personas, los cuales dividen su trabajo en acciones que se

puedan completar en un período de tiempo uniforme denominado "sprint". Los equipos de Scrum están formados por los miembros, un facilitador y el dueño de un producto. Por lo general, este marco de trabajo se implementa cuando un proyecto grande se puede dividir en sprints de dos a cuatro semanas. Además, se centra en ciclos de retroalimentación mediante reuniones donde se realiza una "retrospectiva" del sprint. Su lema no oficial bien podría ser "analice todo y adáptese".

Hay otros marcos ágiles, como kanban, que son anteriores al manifiesto. Sin embargo, se consideran ágiles porque promueven los valores que se detallan en él. Son muchos los marcos y los enfoques ágiles que permiten adaptar este tipo de metodología, y sería difícil enumerarlos a todos.

2.1.3 Eficiencia energética.

2.1.3.1 ¿Qué es la eficiencia energética?

Este término es muy conocido y además ha tomado mucha importancia en nuestro tiempo a causa de la emergencia por el cambio climático que se vive en la actualidad. Una definición aportada por Banco BBVA que literalmente expresa: La eficiencia energética puede definirse como la optimización del consumo energético para alcanzar unos niveles determinados de confort y de servicio, por ejemplo, ajustando el consumo de electricidad a las necesidades reales de los usuarios o implementando mecanismos para ahorrar energía evitando pérdidas durante el proceso.
En un país, disponer de un nivel adecuado de eficiencia energética permite, por ejemplo, aumentar la seguridad de que existirá un abastecimiento de energía suficiente para toda la población. (BBVA: Eficiencia energética, 2015)

2.1.3.2 La eficiencia energética y el cambio climático.

En una afirmación importante del Banco Mundial, la eficiencia energética se ha convertido en un factor importante para la reducción de emisiones de gases efecto invernadero. A continuación, se presenta este concepto, extraído del portal del Banco Mundial.

De acuerdo con (Banco Mundial., 2017), la eficiencia energética —que incluye el ahorro de energía a nivel residencial, industrial y municipal— reviste importancia crítica para alcanzar las metas nacionales relativas a la energía y el cambio climático que se han fijado países de todo el mundo. La eficiencia energética sigue siendo la opción de menor costo para cumplir los compromisos nacionales referidos al cambio climático. Por esta razón, la eficiencia energética se denomina a menudo “el primer combustible”, es decir, el recurso que se debe utilizar antes que cualquier otra alternativa energética.

Además de reducir las emisiones, la eficiencia energética tiene otros beneficios para el desarrollo: mayor seguridad energética, menor presión sobre los presupuestos nacionales y de los hogares, mayor confiabilidad de los sistemas de energía eléctrica, mayor competitividad, y mejora de las operaciones en sectores fundamentales, como la educación y la salud.

Sin embargo, a pesar de este enorme potencial, la eficiencia energética sigue estando subutilizada debido a barreras políticas, técnicas y financieras. Los países y las instituciones de desarrollo como el Grupo Banco Mundial han comprobado que para implementar con éxito programas de eficiencia energética a escala se requiere un compromiso a largo plazo. El financiamiento debe adecuarse a los mercados locales y estar respaldado por políticas, regulaciones y, en algunos casos, incentivos adecuados.

2.1.3.3. Análisis del ciclo de vida del producto

El Análisis de Ciclo de Vida (ACV) es un proceso objetivo que permite evaluar las cargas ambientales asociadas a un producto, proceso o actividad, identificando y cuantificando tanto el uso de materia y energía, como las emisiones al entorno, para determinar el impacto de ese uso de recursos y emisiones generadas, con el propósito de evaluar y llevar a la práctica estrategias de mejora ambiental. (Hernandez & Rieznik, 2005)

El Análisis de ciclo de vida de un producto permite la identificación de los principales impactos ambientales (vertidos, residuos, emisiones a la atmósfera, consumos de materias primas y de energía) teniendo en cuenta todas las etapas de su ciclo de vida, desde su origen, es decir, la extracción y procesado de materias primas, pasando por la producción, transporte y distribución, hasta el uso, mantenimiento, reutilización, reciclado y disposición en vertedero al final de su vida útil. Una vez identificados los principales impactos a lo largo de su ciclo de vida, permite realizar

el análisis de alternativas en procesos productivos y la implementación de criterios ambientales en estrategias. (Eurofins, 2021)

Etapas del Análisis de Ciclo de Vida (ACV)

La realización de un Análisis de ciclo de vida es un proceso laborioso y complejo que requiere de técnicos ambientales especialistas. Dada su complejidad es necesario abordar el trabajo en diferentes etapas:

- **Definición de objetivos y alcance.** Es necesario conocer los objetivos que perseguimos con el estudio antes de abordarlo. Debemos exponer los motivos por los que se lleva a cabo el análisis de ciclo de vida y establecer el alcance del mismo. ¿Qué buscamos con la realización de una ACV?

- **Inventario del Ciclo de Vida (ACV):** Se identifican y cuantifican todas las entradas (consumo de recursos y materiales) y todas las salidas (emisiones al aire, suelo, aguas y generación de residuos) que potencialmente pueden causar un impacto durante el Análisis de ciclo de vida. En esta fase debemos hacer una recopilación de datos y establecer los procedimientos de cálculo para identificar y cuantificar todos los efectos ambientales adversos asociados al proceso o producto objeto de estudio.

- **Evaluación de los Impactos del Ciclo de Vida**: Se establece una relación de entradas y salidas en el inventario con los posibles impactos sobre el medio ambiente, la salud humana y los recursos, con el objetivo de clasificar, caracterizar y evaluar cuánto de importantes son los potenciales impactos que se generan. Se establecen las categorías de impacto y se asignan los datos del inventario a cada categoría de impacto según el tipo de efecto ambiental esperado.

- **Interpretación de resultados**: Tras la realización del análisis de ciclo de vida habremos identificado en qué fases o elementos del ciclo de vida del producto se generan las principales cargas ambientales, y por tanto esos elementos serán claves a la hora de implementar las mejoras, puesto que son los causantes del mayor impacto ambiental, Según los objetivos que se establezcan, se obtienen conclusiones y recomendaciones que

contribuyen a la toma de decisiones. Esta es probablemente la parte crítica y más importante de todo el Análisis de Ciclo de Vida. En el caso de que el objetivo que persigamos sea la comparación de distintos productos, podremos saber cuál de ellos presenta un mejor comportamiento ambiental (Eurofins, 2021)

2.1.4. Estado del arte

2.1.4.1 Objetivo

Realizar una búsqueda especializada en la base de datos bibliográfica scopus de documentos con carácter científico en el tema de metodología scrum para conocer la literatura que en este tema haya sido abordada por personas comprometidas en la industria del software a nivel mundial.

¿Se han realizado investigaciones sobre la metodología SCRUM como modelo para el desarrollo de aplicaciones en el ámbito académico y amigable con el medio ambiente?

2.1.4.2 Protocolo de búsqueda

(TITLE-ABS-KEY (scrum) Y TITLE-ABS-KEY (metodología) O TITLE-ABS-KEY (ágil) Y TITLE-ABS-KEY (metodologías))

Resultado:

1.122 documentos encontrados

2.1.4.3 Filtros

(TITLE-ABS-KEY (scrum) Y TITLE-ABS-KEY (metodología) O TITLE-ABS-KEY (ágil) Y TITLE-ABS-KEY (metodologías)) Y (LIMIT-TO (OA , "todos")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (AFFILCOUNTRY , "United States"))
Resultado:

20 documentos encontrados

2.1.4.4 Palabras clave:

- Scrum
- Diseño de software
- Metodologías ágiles
- Ingeniería de software
- Gestión de proyectos

2.1.4.5 Hallazgos

En este trabajo se muestran los principales hallazgos en el tema de las metodologías ágiles y SCRUM como marco de trabajo. Se realizó una revisión bibliográfica y el análisis de textos para conformar una base de datos compuesta por información obtenida de 18 universidades del mundo en las cuales se aplicó SCRUM como modelo para el desarrollo de aplicaciones.

El diseño de la investigación se realizó como descriptivo con enfoque cualitativo entorno a las siguientes categorías: scrum como metodología de desarrollo y scrum como metodología ágil.

Matriz bibliográfica

Tabla 1. Matriz bibliográfica

Nro.	Título	Autores	Año de publicación	Tipo de documento	Páginas	Resumen	Referencia al documento
1							
2							
3							
.							

La forma como se estructuró la matriz bibliográfica para la base de datos es la siguiente:

La matriz bibliográfica para la base de datos se estructuró en 7 columnas las cuales indican el número mediante el cual se identifica el registro (Nro.); la identificación del documento (Título); las personas que escribieron el documento (Autores); el año en el que el documento fue publicado (Año); la categorización del documento, el cual puede ser un artículo, un libro o un acta de conferencia (Tipo de documento); un breve texto que permite identificar de qué se trata el documento (Resumen); y la (Referencia al documento) la cual permite obtenerlo.

Una vez que se consultaron los datos y se obtuvieron los resultados que se encuentran en la matriz, los documentos se han categorizado de acuerdo con la siguiente clasificación:

Matriz analítica de contenido por tipo de documento

Tabla 2. Matriz analítica de contenido por tipo de documento

Nro.	Documento	Artículo	Libros	Acta de conferencia	Observaciones
1					
2					
3					

En la matriz analítica de contenido se realiza una categorización de la información la cual se refiere al tipo de documento encontrado. Para este caso hemos encontrado artículos, libros y actas de conferencia.

Se plantea otra categorización para la base de datos la cual nos permite determinar la cantidad de registros por país:

Matriz analítica de contenido por país

Tabla 3. Matriz analítica de contenido por país

Pakistán	Ucrania	Estados Unidos	Alemania	España	Indonesia	Jamaica	Colombia	Ecuador	México	Chile
1	1	3	3	1	2	1	2	4	2	1

Para verificar la información que componen las tablas anteriores, Ver

Anexo 1. Base de datos para el estado del arte (hoja electrónica Excel).

Conclusiones

- La mayoría de información consultada, son documentos de tipo acta de conferencia, seguido de artículos, donde los autores expresan su percepción en la aplicación o investigación sobre la metodología Scrum.

- El país que más ha suministrado información respecto a la metodología scrum es Ecuador, seguido de Estados Unidos y Alemania.

- La gran mayoría de la información pertenece a universidades donde a través de actas de conferencia expusieron sus respectivas investigaciones y/o aplicaciones de la metodología.

- Scrum como marco de trabajo es aceptado y empleado a nivel mundial, sin embargo, no se encuentra un estudio que permita determinar aspectos en los cuales se hayan intervenido los eventos scrum de tal manera que se refute al autor acerca de la importancia de mantener tiempos fijos en cada uno.

- Se encontraron diversos trabajos con el marco scrum en los cuales se evidencia la aplicación exhaustiva de los tiempos en cada evento sin darle importancia a la optimización de los recursos que involucra la ejecución de cada uno.

- Para la industria del software se nota la importancia de trabajar con scrum como un método tradicional de factoría dándole importancia al cumplimiento de cada actividad en

los eventos. No se encontró información en la cual la industria del software haga reparos al cumplimiento de los tiempos por cada evento.

Las referencias consultadas se incluyen al final del documento.

Capítulo 3. Metodología de la investigación aplicada al proyecto

En este capítulo se expone el tipo de investigación y el enfoque con el que se realizó este trabajo, se explica el proceso que se definió utilizar y de que trata cada una de sus fases.

3.1 Tipo de investigación.

Para el presente proyecto se ha decidido trabajar con un tipo de investigación descriptivo con enfoque cualitativo, dadas las características del proceso investigativo que se debe seguir para entender el marco de trabajo scrum y los elementos con los cuales se logra un ahorro energético importante que beneficie el medio ambiente. A continuación, se mencionan las definiciones que enmarcaron metodológicamente este proyecto.

3.1.1 Investigación descriptiva.

Los estudios descriptivos se enfocan en describir a detalle fenómenos, problemáticas, contextos determinados o eventos específicos. En ese sentido, se centran en reunir y estructurar datos sobre las propiedades y características de aquello que están examinando, lo cual significa que el (la) investigador(a) ya tiene más o menos claro dónde va a colocar su atención. En los estudios descriptivos, no se establecen relaciones entre los conceptos y variables que se miden, sino que se tratan de manera independiente. (Hernández, Fernández y Baptista)

3.1.2 Enfoque de la investigación. Enfoque cualitativo.

El objetivo de la investigación cualitativa es el de proporcionar una metodología de investigación que permita comprender el complejo mundo de la experiencia vivida desde el punto de vista de las personas que la viven (Taylor y Bogdan, 1984). Las características básicas de los estudios cualitativos se pueden resumir en que son investigaciones centradas en los sujetos, que adoptan la perspectiva emic o del interior del fenómeno a estudiar de manera integral o completa.

El proceso de indagación es inductivo y el investigador interactúa con los participantes y con los datos, busca respuestas a preguntas que se centran en la experiencia social, cómo se crea y cómo da significado a la vida humana.

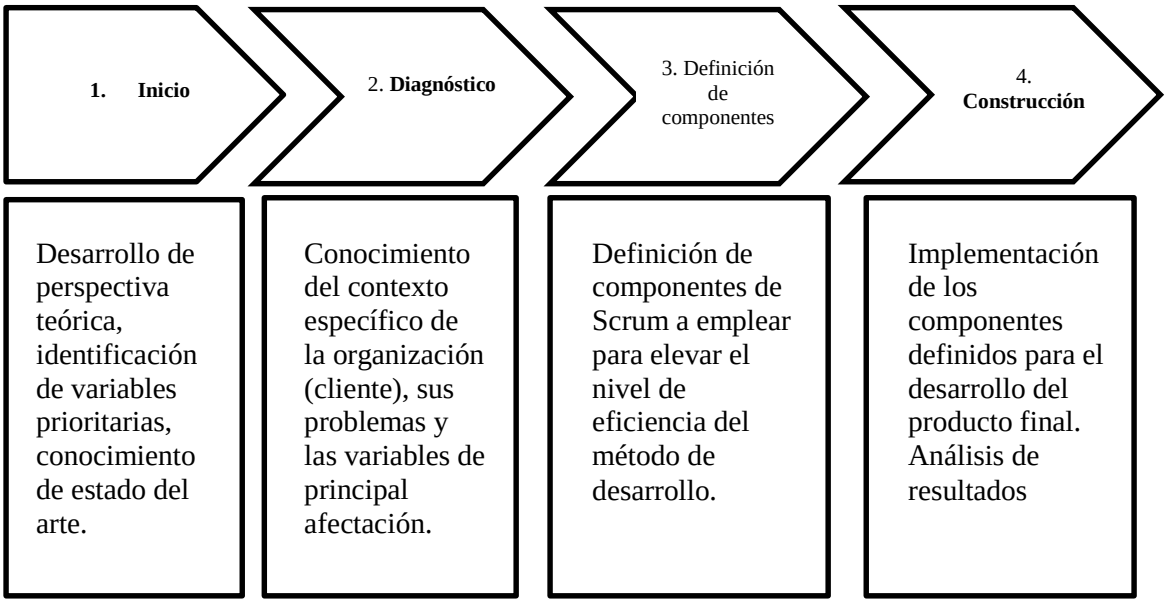
Taylor y Bogdan dicen que el investigador cualitativo pretende comprender lo que la gente dice.

Para comprender el tipo y enfoque de la investigación, se ha decidido trabajar un proceso definido en 4 fases. Esas fases son:

1. Inicio
2. Diagnóstico
3. Definición de componentes
4. Construcción y análisis de resultados

En la siguiente ilustración se muestra el método de cuatro fases aplicado al proyecto.

Ilustración 2. Diseño metodológico del proyecto



En la fase 1. Se incluye el estudio inicial del proyecto a partir del conocimiento del marco teórico necesario para entender los conceptos y el estado del arte que ha facilitado en conocimiento y comprensión de los avances de la ciencia en el tema del marco de trabajo scrum

En la fase 2. Se presenta un estudio que permite abordar el conocimiento del contexto específico de la organización (cliente), sus problemas y las variables de principal afectación. Aquí se incluye una descripción del sistema de información actual y sus nuevas funcionalidades solicitadas por el cliente. Para el caso del método de desarrollo de software seleccionado, se hace referencia a las historias de usuario en las cuales el cliente nos presenta una visión muy completa acerca del sistema actual y las necesidades de mejora en sus procesos. Para este caso se ha decidido trabajar con unas fichas denominadas historias de usuario en las cuales se plasma el contenido de las entrevistas que se realizan a las personas que tienen el dominio del conocimiento del sistema

En la fase 3. Al estudiar los componentes teóricos del marco de trabajo scrum, se definen los que harán parte del proyecto o aquellos que pueden ser ajustados de acuerdo con las características de cada proceso. Para ello se tiene en cuenta:

- Eventos que se definen con tiempos exactos para su ejecución
- Eventos que pueden ser optimizados en cuanto al uso de recursos
- Actividades de cada evento en las cuales sea posible una definición de tiempos y esfuerzos de personal para buscar oportunidades de optimización
- Actividades de cada evento que consuman recursos energéticos
- Actividades que se puedan optimizar de acuerdo con el ACV con el propósito de disminuir emisiones de gases efecto invernadero y de esta forma obtener productos amigables con el medio ambiente desde su fabricación.

En la fase 4. Se inicia el proceso de desarrollo de la aplicación de acuerdo con las decisiones tomadas en la fase anterior. Se espera como resultado la implementación de los componentes definidos para cada ciclo del desarrollo y se analiza el comportamiento de cada decisión ejecutada. Al finalizar esta fase, debe quedar finalizado el proceso de desarrollo y documentado el resultado final.

Capítulo 4. Ingeniería del proyecto.

Resultados de la aplicación del marco de trabajo scrum

En este capítulo se detalla la adaptación de la metodología en el desarrollo de la aplicación móvil, explicando la implementación de cada una de sus fases de acuerdo a las necesidades del equipo de desarrollo.

4.1. Situación actual de la empresa:

Colombia Software ofrece a sus clientes un producto web modular integrado que permite la administración de la operación de transporte desde la cotización del servicio de un cliente hasta la facturación, cobro de cartera y contabilización. Además, brinda un servicio personalizado de acuerdo a las necesidades y directrices que tenga el cliente, por ello la empresa cuenta con un área comercial donde los gestores de proyectos se hacen cargo de un número de clientes a los cuales visitan y escuchan a diario. De estas visitas surgen nuevos requerimientos o ajustes al software según la necesidad de cada cliente, por esto es de gran importancia contar con un equipo robusto de soporte y desarrollo, los cuales constantemente atienden dudas, y realizan los desarrollos necesarios indicados por los gestores de proyectos.

Debido a que cada vez es más demandante el avance tecnológico, ha sido necesario incursionar en aplicaciones móviles que extiendan algunas funcionalidades del software web para la comodidad del usuario. Estas aplicaciones despiertan el interés de otros posibles clientes y facilitan la gestión de los que ya están vinculados, generando así, nuevas ideas y herramientas para que la información llegue de manera más oportuna al usuario final del proceso. Este proceso, requiere personal capacitado para el desarrollo de dichas aplicaciones, además de tener en cuenta que el equipo actual tiene sus tareas definidas y sus horas comprometidas en el software web.

Por esta razón se hizo necesario crear un pequeño grupo dentro de la empresa que involucra las personas necesarias para el desarrollo de esta aplicación móvil. Teniendo en cuenta la limitante del tiempo disponible de cada una de ellas y el tiempo de entrega de la aplicación, se tomó la

decisión de apoyarnos en una metodología ágil que nos permitiera mantener el proyecto organizado y en marcha constante. De esta manera se implementaron los procesos más importantes de esta metodología y se logró mantener contacto con el cliente y los miembros del equipo hasta la entrega final.

4.2. Proceso de desarrollo basado en SCRUM

4.2.1 FASE 1: Creación del equipo Scrum

En esta fase se seleccionaron aquellas personas necesarias y capacitadas para desarrollar dicho proyecto:

- **Product Owner**: de este rol se encargó el coordinador de desarrollo de la empresa ya que nos representaba la voz del cliente, comprendía las necesidades del cliente debido a su experiencia en el tema de transporte y tenía la capacidad de asegurar los recursos financieros para el proyecto. Es importante resaltar que en esta labor también participó la gestora de proyectos, ya que tiene comunicación constante con el cliente y lidera sus requerimientos generales.

- **Scrum Master**: de este rol se encargó un desarrollador de la empresa con habilidades para liderar, planificar y comunicarse constantemente con el equipo para implementar de manera adecuada la metodología.

- **Development Team**: este equipo se conformó por tres personas. dos de ellas capacitadas para el desarrollo de la aplicación y una más el cual es el encargado del área TIC de la empresa y por ende tiene el manejo total de los servidores y accesos necesarios para el desarrollo. Este equipo fue el encargado de crear el entorno de desarrollo y la aplicación final entregable al cliente.

- **Perfil de los miembros del equipo**

Nombre	Diego Fernando Pinillo Solís
Rol	Product Owner

Cargo	Coordinador de desarrollo
Perfil	Ingeniero de Sistemas, se encuentra vinculado a Colombia Software hace 15 años, se desempeña en el cargo de Coordinador de Desarrollo hace 10 años. Tiene conocimientos en el área del transporte y está capacitado en el proceso de Scrum.

Nombre	Carlos Edward Grajales Marmolejo
Rol	Development Team
Cargo	Coordinador TIC
Perfil	Ingeniero de Sistemas, se encuentra vinculado a Colombia Software hace 15 años desempeñando el cargo de Coordinador TIC. Tiene conocimientos en el área del transporte, redes informáticas, servidores y está capacitado en el proceso de Scrum.

Nombre	Mónica Alejandra Vargas Jaramillo
Rol	Scrum master - Development Team
Cargo	Líder de calidad de producto
Perfil	Ingeniera de Sistemas, se encuentra vinculada a Colombia Software hace 8 años, en los cuales 6 años desempeñó el cargo Gestor de Desarrollo y actualmente se desempeña en el cargo Líder de Calidad de Producto. Tiene conocimientos en diferentes lenguajes de programación, análisis de requerimientos y está capacitada en el proceso de Scrum.

Nombre	Pedro Aníbal Ospina Posada
Rol	Development Team
Cargo	Gestor de desarrollo
Perfil	Tecnólogo en Sistemas con 5 años de vinculación en Colombia Software donde desempeñó el cargo de Gestor de Desarrollo. Tiene conocimientos en diferentes lenguajes de programación, análisis de requerimientos y está capacitado en el proceso de Scrum.

Nombre	Diana Marcela Zamora Cuervo
Rol	Organismo asesor
Cargo	Gestor de proyectos
Perfil	Ingeniera de Sistemas, se encuentra vinculada a Colombia Software hace 4 años desempeñándose en el cargo Gestora de proyectos. Tiene conocimientos en el área del transporte y atención al cliente.

4.2.2 FASE 2: Planeación de sprints, eventos y artefactos de Scrum

En esta fase se describe la planeación de cada sprint con sus respectivos eventos y la manera en que se implementa.

Inicialmente se hace una reunión con todo el equipo donde se analiza la historia de usuario creada por el cliente la cual ha sido previamente registrada mediante el software web Toolproject, el cual es una herramienta propia donde se lleva la trazabilidad de cada solicitud; se revisa cada requerimiento y se determinan los detalles que hacen falta para el correcto desarrollo de la aplicación, también se discuten los recursos informáticos que se necesitan para el desarrollo de la aplicación con el área TIC de la empresa, así como el framework a utilizar, que para este proyecto se determinó Ionic por su versatilidad y experiencia de los desarrolladores. Concluyendo la reunión, el scrum master queda encargado de actualizar la historia de usuario inicial con los detalles necesarios para que el requerimiento quede completo y una vez se complete la historia de usuario, se programa la próxima reunión para planificar el primer sprint del proyecto.

4.2.2.1 Sprint 1. Planificación del Sprint.

En esta reunión se revisa la historia de usuario general, se enumera cada punto y se generan historias de usuario más pequeñas de cada requerimiento, de esta manera se conforma el Product

Backlog el cual será la única fuente de requisitos para cualquier cambio a realizarse en el producto. Ver Anexo 2. Product Backlog. Historias de usuario.

- **Refinamiento del Product Backlog.**

Una vez se tiene enumeradas las historias de usuario de acuerdo a su prioridad, se revisa cada una y se agrega mayor detalle si el equipo de trabajo lo considera necesario.

- **Sprint Backlog.**

Una vez se tiene el Product Backlog refinado, se procede a tomar cada historia de usuario para definir las tareas necesarias para su desarrollo y el tiempo que se va a dedicar a cada una. Para estimar este tiempo, se toma como base, la experiencia de los desarrolladores que hayan tenido en este tipo de aplicaciones de acuerdo al framework definido y el previo conocimiento de la estructura base "Silogtran". De esta manera se seleccionan las historias de usuario que se pueden desarrollar en el sprint de acuerdo al tiempo destinado para este.

Tabla 4. Cálculo de tiempos Sprint 1

Tiempo del recurso humano	Desarrollador 1	Desarrollador 2
Actividades Scrum (actualizar el tablero de Trello)	6	2
Imprevistos	4	4
SUB TOTAL	**10**	**6**
Sprint - 2 semanas - 40 horas	30	34
Capacidad del 80%	**24**	**27**
TOTAL HORAS DESARROLLO	**51**	

Las historias de usuario a realizar en el sprint 1 según las horas disponibles para el desarrollo son las siguientes:

Anexo 2. Product Backlog. Tareas. HU01 - Inicio de sesión

Anexo 2. Product Backlog. Tareas. HU02 - Cierre de sesión

- **Ejecución del Sprint:**

Para organizar cada historia de usuario a desarrollar y sus tareas, se utilizó la aplicación Trello, la cual permite organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.

- **Scrum Diario (Daily Scrum)**

Esta reunión se llevó a cabo los días en que fue necesario el diálogo de los desarrolladores para aclarar dudas y/o planear la ejecución del requerimiento. Se debe tener en cuenta que esta aplicación se desarrolló en horarios extraoficiales, por lo tanto, los desarrolladores no estaban en el mismo lugar realizando el desarrollo.

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

En esta reunión, el equipo de scrum presenta los resultados de su trabajo a las partes interesadas clave y se discute el progreso hacia el Objetivo de Producto. En este sprint, se presenta la HU01 - Inicio de sesión y HU02 - Cierre de sesión, al product owner para que valide su funcionalidad de acuerdo con las condiciones expuestas en las respectivas historias de usuario. Para la aprobación de este sprint se acordó una revisión interna sin exponer los resultados al cliente ya que las historias de usuario desarrolladas denotan la base estructural del proyecto a nivel de servicios e interfaz y no exponen mucha información de interés para el cliente.

- **Retrospectiva del Sprint (Sprint Retrospective**)

En esta reunión el equipo de Scrum inspecciona cómo fue el último Sprint con respecto a individuos, interacciones, procesos, herramientas y su definición de hecho. Se expone cómo se estructuró el proyecto en sus dos partes principales. La primera es la API, la cual se debe comunicar con el proyecto base Silogtran para la respectiva extracción de información. La segunda parte del proyecto es la aplicación móvil, donde se expone su estructura, los colores definidos y el diseño a utilizar para las demás historias de usuario. También se comentan las dificultades que se tuvieron y cómo se resolvieron mediante las reuniones diarias realizadas

logrando cumplir con el objetivo del producto terminado y en condiciones para utilizarse. De esta manera finaliza el sprint 1 del proyecto, se actualiza el incremento y se planifica la próxima reunión para la planificación del sprint 2.

4.2.2.2 Sprint 2. Planificación Del Sprint (Sprint Planning)

En esta reunión se verifica el incremento y se compara con el Product Backlog para tener en cuenta sólo aquellas historias de usuario que hacen falta por terminar.

- **Sprint Backlog**

Una vez se tenga el Product Backlog depurado, se procede a seleccionar las historias de usuario que, de acuerdo al tiempo estimado y el tiempo dispuesto para el desarrollo, se puedan terminar.

Tabla 5. Cálculo de tiempos Sprint 2

Tiempo del recurso humano	Desarrollador 1	Desarrollador 2
Actividades Scrum (actualizar el tablero de Trello)	4	2
Imprevistos	8	8
SUB TOTAL	**12**	**10**
Sprint - 4 semanas - 80 horas	68	70
Capacidad del 80%	**54**	**56**
TOTAL HORAS DESARROLLO	**110**	

Las historias de usuario a realizar en el sprint 2 según las horas disponibles para el desarrollo son las siguientes:

Anexo 2. Product Backlog. Tareas. HU03 - Secciones principales
Anexo 2. Product Backlog. Tareas. HU04 - Sección Viajes
Anexo 2. Product Backlog. Tareas. HU05 - Filtro estado - Sección Viajes

Anexo 2. Product Backlog. Tareas. HU06 - Lista de información - Sección Viajes

Anexo 2. Product Backlog. Tareas. HU07 - Detalle viaje - Sección Viajes

Anexo 2. Product Backlog. Tareas. HU08 - Subsección Resumen - Sección Viajes

Anexo 2. Product Backlog. Tareas. HU09 - Subsección Tráfico - Sección Viajes

Anexo 2. Product Backlog. Tareas. HU10 - Sección Novedades - Subsección Tráfico

Anexo 2. Product Backlog. Tareas. HU11 - Sección Puesto de control - Subsección Tráfico

- **Ejecución del Sprint:**

Para organizar cada historia de usuario a desarrollar y sus tareas, se utilizó la aplicación Trello, la cual permite organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.

- **Scrum Diario (Daily Scrum):**

Esta reunión se llevó a cabo los días en que fue necesario el diálogo de los desarrolladores para aclarar dudas y/o planear la ejecución del requerimiento.

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

En este sprint, se presenta el desarrollo de las historias de usuario previamente seleccionadas al equipo scrum, se explica su funcionamiento y se validan las condiciones expuestas en las respectivas historias de usuario. El equipo scrum aprueba el sprint y se libera el incremento del sprint 1 y 2, para realizar la primera entrega oficial al cliente para que empiece a realizar las respectivas pruebas.

- **Retrospectiva del Sprint (Sprint Retrospective):**

En esta reunión se inspecciona cómo fue el último Sprint con respecto a individuos, interacciones, procesos, herramientas y su definición de hecho. Se expone la manera en la que se abordaron los requerimientos debido a su complejidad y los pormenores que ocurrieron durante el desarrollo. Por lo tanto, se determina dar cierre al sprint 2 y a agendar la reunión para planificar el sprint 3.

4.2.2.3 Sprint 3. Planificación Del Sprint (Sprint Planning):

En esta reunión se verifica el incremento y se compara con el Product Backlog para tener en cuenta sólo aquellas historias de usuario que hacen falta por terminar.

- **Sprint Backlog:**

Una vez se tiene el Product Backlog depurado, se procede a seleccionar las historias de usuario que de acuerdo al tiempo estimado y el tiempo dispuesto para el desarrollo, se puedan terminar.

Tabla 6. Cálculo de tiempos sprint 3

Tiempo del recurso humano	Desarrollador 1	Desarrollador 2
Actividades Scrum (actualizar el tablero de Trello)	4	2
Imprevistos	8	8
SUB TOTAL	**12**	**10**
Sprint - 4 semanas - 80 horas	68	70
Capacidad del 80%	**54**	**56**
TOTAL HORAS DESARROLLO	**110**	

Las historias de usuario a realizar en el sprint 3 según las horas disponibles para el desarrollo son las siguientes:

Anexo 2. Product Backlog. Tareas. HU12 - Subsección Documentación - Sección Viajes
Anexo 2. Product Backlog. Tareas. HU13 - Sección Facturas
Anexo 2. Product Backlog. Tareas. HU14 - Lista - Sección Facturas
Anexo 2. Product Backlog. Tareas. HU15 - Detalle factura - Sección Facturas
Anexo 2. Product Backlog. Tareas. HU16 - Detalle remesa - Detalle factura

- **Ejecución del Sprint:**

Para organizar cada historia de usuario a desarrollar y sus tareas, se utilizó la aplicación Trello, la cual permite organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.

- **Scrum Diario (Daily Scrum):**

Esta reunión se llevó a cabo los días en que fue necesario el diálogo de los desarrolladores para aclarar dudas y/o planear la ejecución del requerimiento.

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

En este sprint, se presenta el desarrollo de las historias de usuario previamente seleccionadas al equipo scrum, se explica su funcionamiento y se validan las condiciones expuestas en las respectivas historias de usuario. El equipo scrum aprueba el sprint y se libera el incremento del sprint 3, para realizar la segunda entrega oficial al cliente para que realice las respectivas pruebas.

- **Retrospectiva del Sprint (Sprint Retrospective):**

En esta reunión se inspecciona cómo fue el último Sprint con respecto a individuos, interacciones, procesos, herramientas y su definición de hecho. Se expone la manera en la que se abordaron los requerimientos debido a su complejidad y los pormenores que ocurrieron durante el desarrollo. También se recibe respuesta por parte del cliente de acuerdo con las pruebas realizadas mediante la herramienta Toolproject, generando nuevos requerimientos para perfeccionar el funcionamiento de la aplicación, por lo tanto, el scrum master queda encargado de gestionar estas nuevas solicitudes para desplegarlas en historias de usuarios con sus respectivas tareas. Al final de la reunión se determina dar cierre al sprint 3 y a agendar la reunión para planificar el sprint 4.

4.2.2.4 Sprint 4. Planificación del Sprint (Sprint Planning):

En esta reunión se verifica el incremento y se compara con el Product Backlog para tener en cuenta sólo aquellas historias de usuario que hacen falta por terminar.

- **Sprint Backlog:**

Una vez se tenga el Product Backlog depurado, se procede a seleccionar las historias de usuario que de acuerdo al tiempo estimado y el tiempo dispuesto para el desarrollo, se puedan terminar.

Tabla 7. Cálculo de tiempos Sprint 4

Tiempo del recurso humano	Desarrollador 1	Desarrollador 2
Actividades Scrum (actualizar el tablero de Trello)	4	2
Imprevistos	4	4
SUB TOTAL	**8**	**6**
Sprint - 2 semanas - 40 horas	32	34
Capacidad del 80%	**26**	**27**
TOTAL HORAS DESARROLLO	**53**	

Las historias de usuario a realizar en el sprint 3 según las horas disponibles para el desarrollo son las siguientes:

Anexo 2. Product Backlog. Tareas. HU18 - Solicitud de Cambio No. 1

Anexo 2. Product Backlog. Tareas. HU19 - Solicitud de Cambio No. 2

Anexo 2. Product Backlog. Tareas. HU20 - Solicitud de Cambio No. 3

Anexo 2. Product Backlog. Tareas. HU21 - Solicitud de Cambio No. 4

Anexo 2. Product Backlog. Tareas. HU22 - Solicitud de Cambio No. 5

Anexo 2. Product Backlog. Tareas. HU23 - Solicitud de Cambio No. 6

Anexo 2. Product Backlog. Tareas. HU24 - Solicitud de Cambio No. 7

Anexo 2. Product Backlog. Tareas. HU25 - Solicitud de Cambio No. 8

- **Ejecución del Sprint:**

Para organizar cada historia de usuario a desarrollar y sus tareas, se utilizó la aplicación Trello, la cual permite organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.

- **Scrum Diario (Daily Scrum):**

Esta reunión se llevó a cabo los días en que fue necesario el diálogo de los desarrolladores para aclarar dudas y/o planear la ejecución del requerimiento.

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

En este sprint, se presenta el desarrollo de las historias de usuario previamente seleccionadas al equipo scrum, se explica su funcionamiento y se validan las condiciones expuestas en las respectivas historias de usuario. El equipo scrum aprueba el sprint y se libera el incremento del sprint 4, para realizar la última entrega oficial al cliente para que realice las respectivas pruebas.

Anexo 3. Pruebas unitarias

- **Retrospectiva del Sprint (Sprint Retrospective):**

En esta reunión se inspecciona cómo fue el último Sprint con respecto a individuos, interacciones, procesos, herramientas y su definición de hecho. Se expone la manera en la que se abordaron los requerimientos debido a su complejidad y los pormenores que ocurrieron durante el desarrollo. Se da por terminado todo el desarrollo de la aplicación y se solicita su despliegue a la tienda de google para que los usuarios interesados puedan acceder a ella. Finalmente se determina dar cierre al sprint 4 y se actualiza el requerimiento mediante la herramienta Toolproject para que el cliente finalice el proceso con su aprobación.

4.3 Arquitectura del sistema

La parte visual (frontend) de la aplicación se desarrolló en el framework Ionic, el cual es una estructura tecnológica de código abierto que se utiliza en el desarrollo de aplicaciones móviles híbridas, es decir, se combinan el HTML5, CSS y JavaScript dando como resultado aplicaciones con una interfaz amigable e intuitiva para el usuario.

El backend de la aplicación es un API basada en la arquitectura REST la cual se desarrolló en PHP con el framework Laravel, utilizando patrones de diseño como Factory y Repository, debido a que esta API sería la base para la creación de diferentes aplicaciones para varios clientes, era necesario tener un control multi-empresa el cual lo brinda el patrón de diseño Factory, permitiendo administrar la lógica de cada cliente de manera organizada.

Por otra parte, el patrón Repository nos permite tener la lógica que recupera los datos independientemente a la lógica de negocio. De esta manera este API se comunica con la base de datos donde se encuentra la fuente de datos y envía la información al frontend para su posterior visualización al usuario.

La aplicación está basada en un modelo de 3 capas las cuales se componen del nivel de usuario el cual proporciona la interfaz visual que el usuario utilizará para ver la información y los datos.

El nivel de negocios hace el puente entre los servicios de usuario y el nivel de servicio de datos.

Los objetos de negocio proporcionan servicios que completan las tareas de negocio tales como verificar los datos enviados por el usuario, antes de llevar a cabo una transacción en la base de datos.
Este nivel también sirve para evitar que el usuario tenga acceso directo a la base de datos, lo cual proporciona mayor seguridad en la integridad de ésta.

Por último, se encuentra el nivel de datos, el cual se encarga de tareas como inserción, modificación, consulta y borrado de información.
Este nivel es el encargado de la gestión de las peticiones realizadas por un objeto de negocio.

En la siguiente ilustración se muestra la arquitectura diseñada para el sistema nuevo.

Ilustración 3. Diagrama de arquitectura

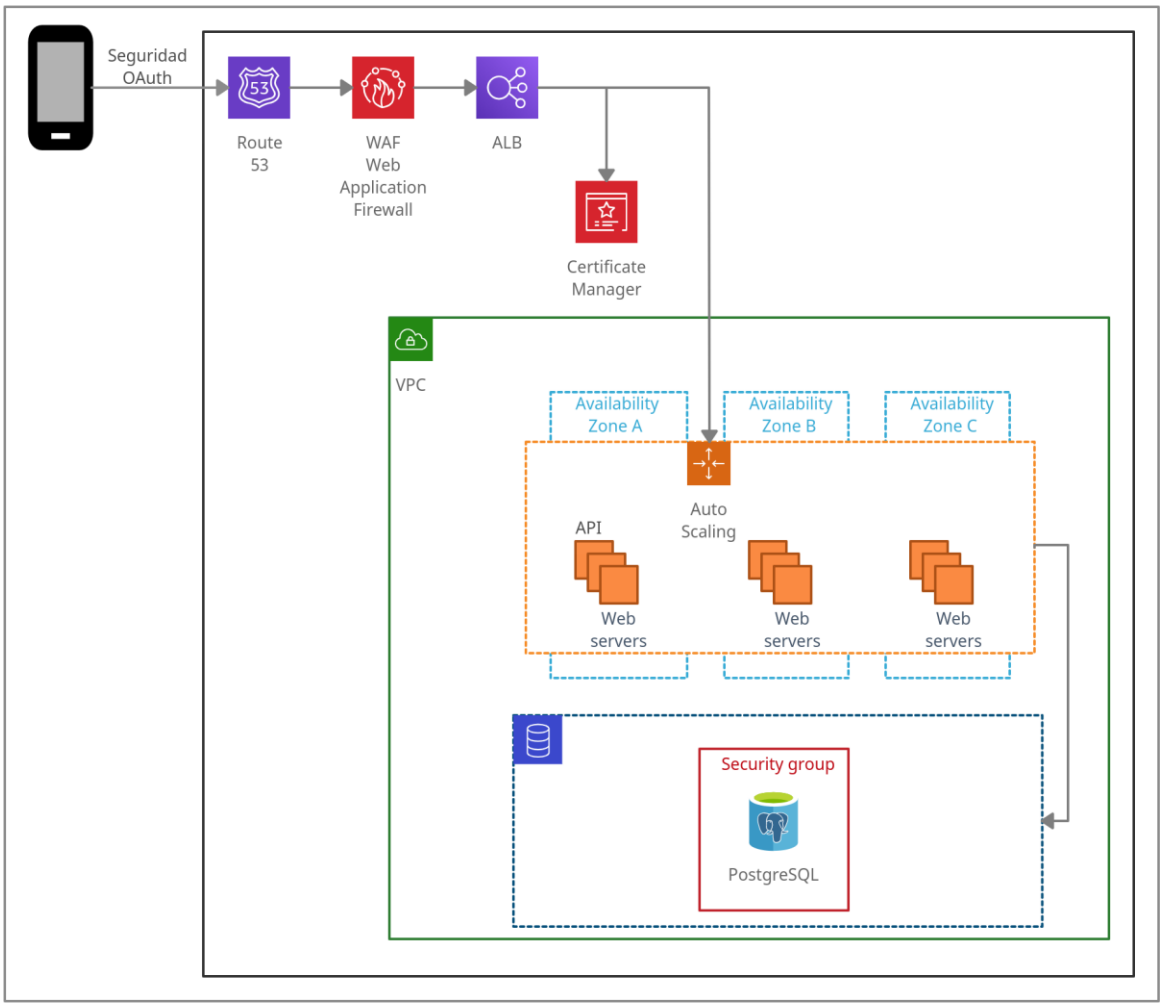


Ilustración 4. Patrón de diseño Factory

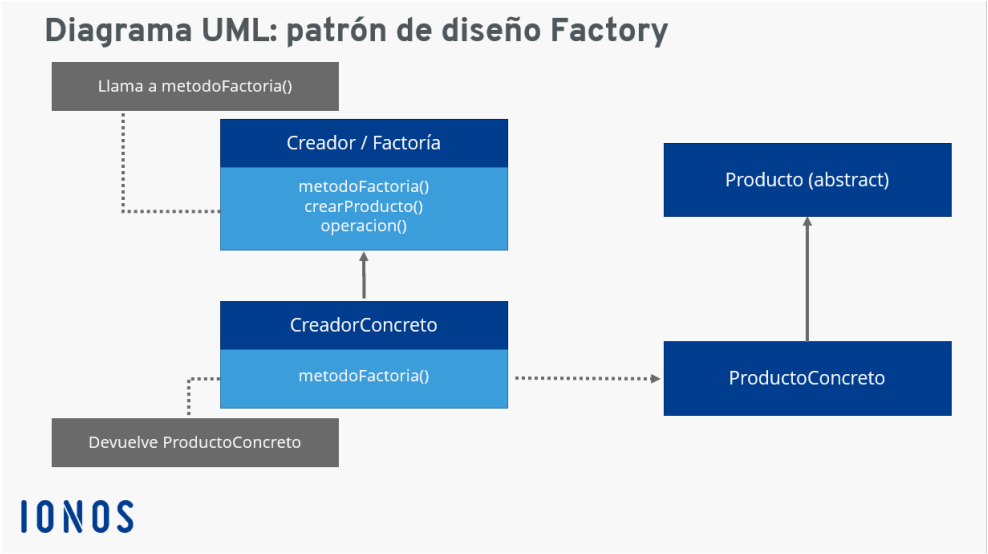
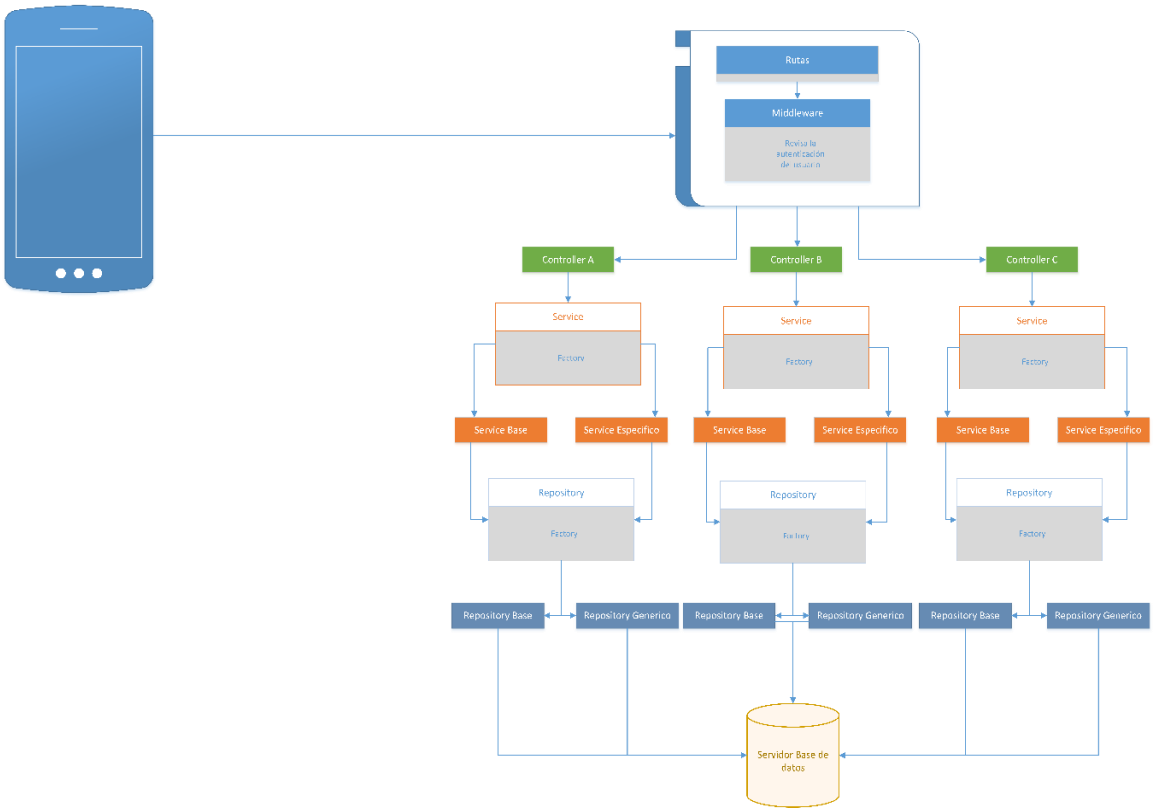


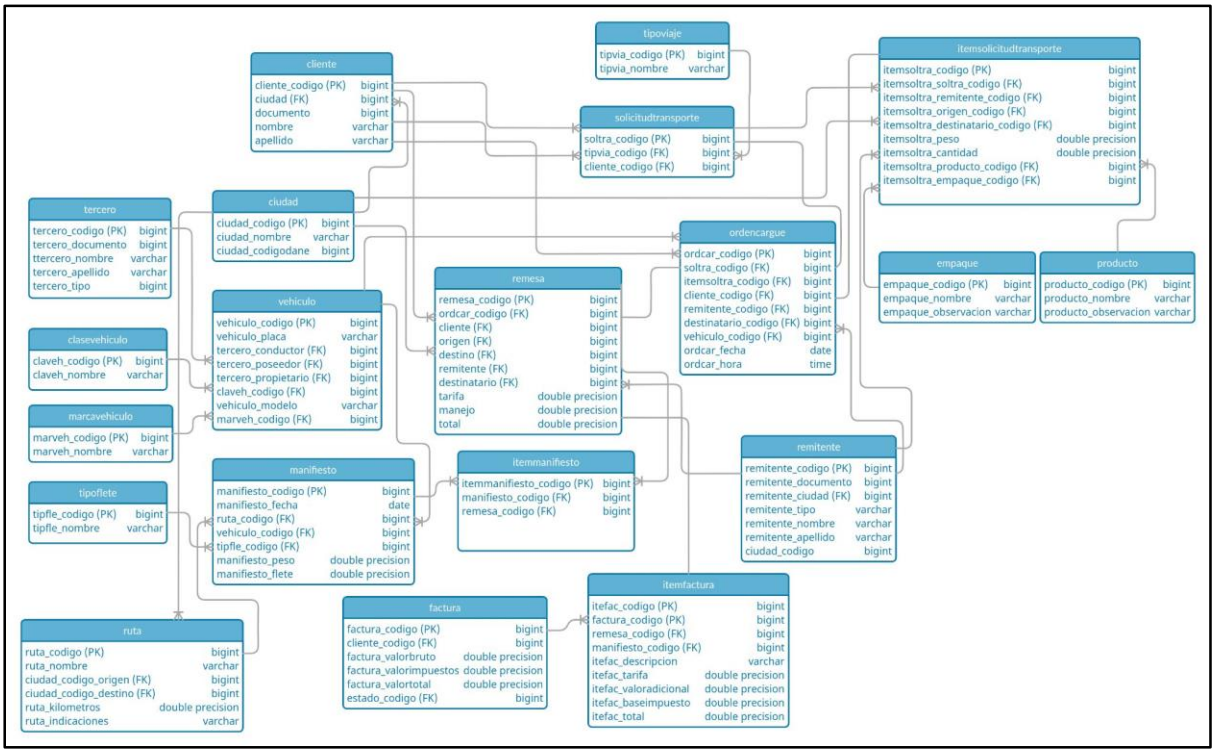
Ilustración 5. Diagrama de componentes



4.3.2 Modelo de la base de datos

La siguiente ilustra contiene el modelo relacional empleado para el aplicativo

Ilustración 6. Modelo de la base de datos



4.3.2.1 Estándares empleados

La información del sistema está estructurada en una base de datos relacional, la cual permite almacenar los datos ordenadamente bajo el siguiente estándar:

- Para las tablas: tb_nombretabla
- Para los atributos: cuando el nombre de la tabla se compone de una sola palabra después del prefijo “tb”, se toma el nombre de la tabla, guion al piso, nombre del campo.
- Cuando la tabla se compone de dos o más palabras después del prefijo “tb”, se toman las primeras tres letras de cada nombre, guion al piso, nombre del campo.

Ejemplo:

Tabla: tb_itemsolictudtransporte, tb_cliente

Campo: itesoltra_codigo, cliente_nombre

4.3.2.2 Motor de la base de datos.

El motor que se eligió para la creación de esta robusta base de datos es postgreSQL ya que es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, brinda escalabilidad, estabilidad y confiabilidad, tiene herramienta gráfica con la cual se puede administrar la base de datos de forma fácil e intuitiva, además, implementa casi todas las funcionalidades del estándar ISO/IEC 9075:2011 y brinda potencia y robustez

4.4 Diseño de las interfaces gráficas de usuario final

4.4.1 Diseño.

Las siguientes ilustraciones muestran las interfaces gráficas de usuario final:

Ilustración 7. Interfaz gráfica 1

Ilustración 8. Interfaz gráfica 2

Ilustración 9. Interfaz gráfica 3

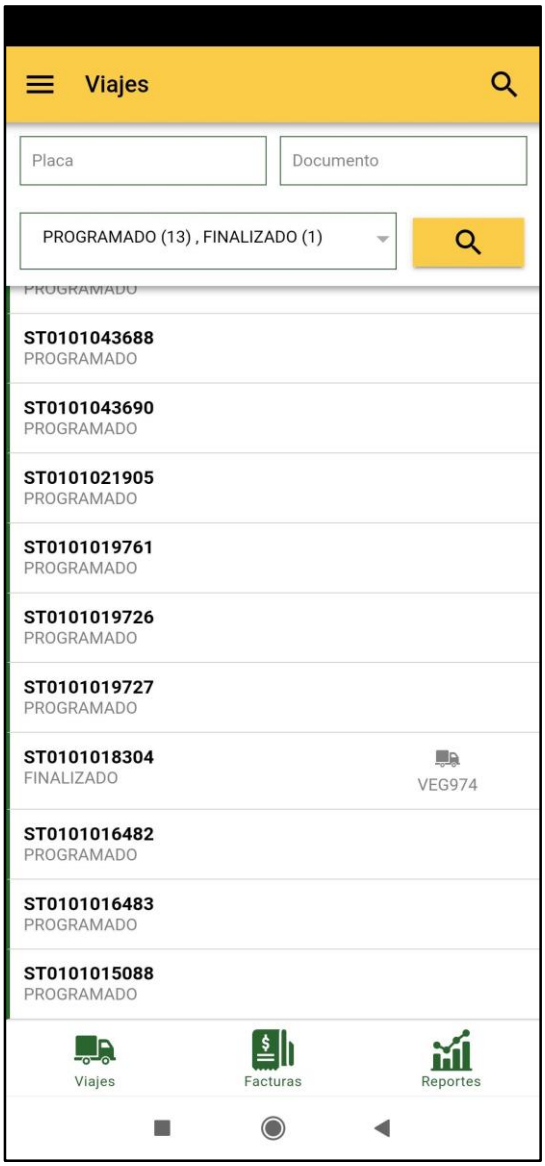


Ilustración 10. Interfaz gráfica 4

Ilustración 11. Interfaz gráfica 5

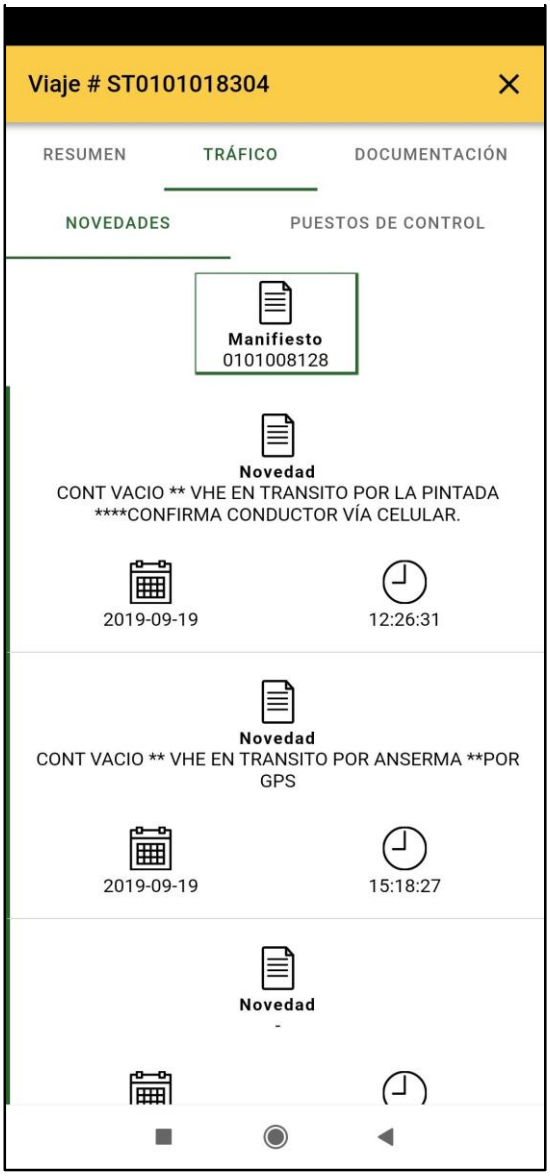


Ilustración 12. Interfaz gráfica 6

Ilustración 13. Interfaz gráfica 7

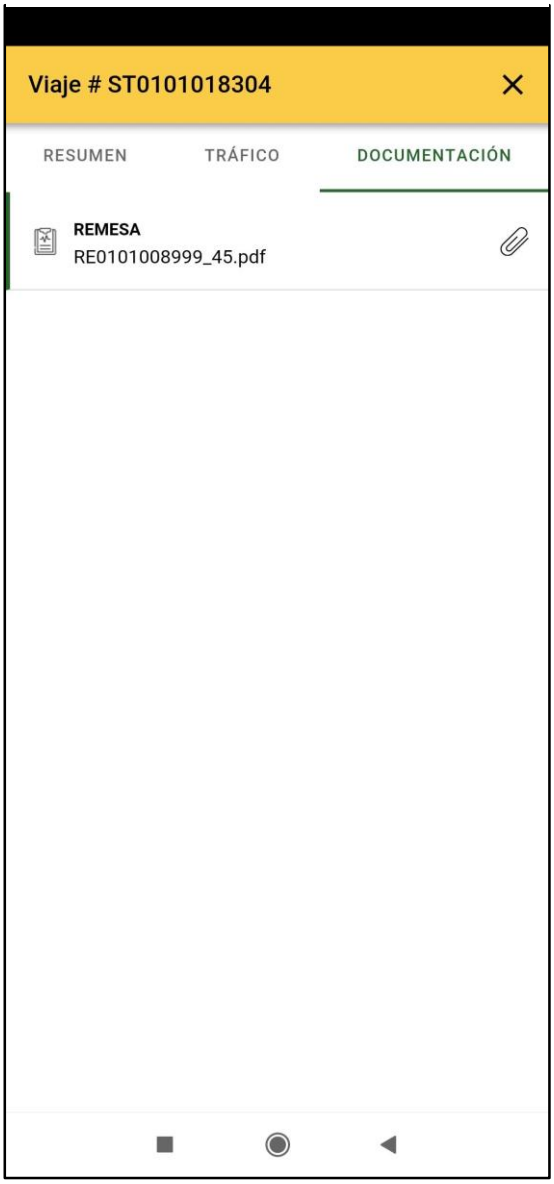


Ilustración 14. Interfaz gráfica 8

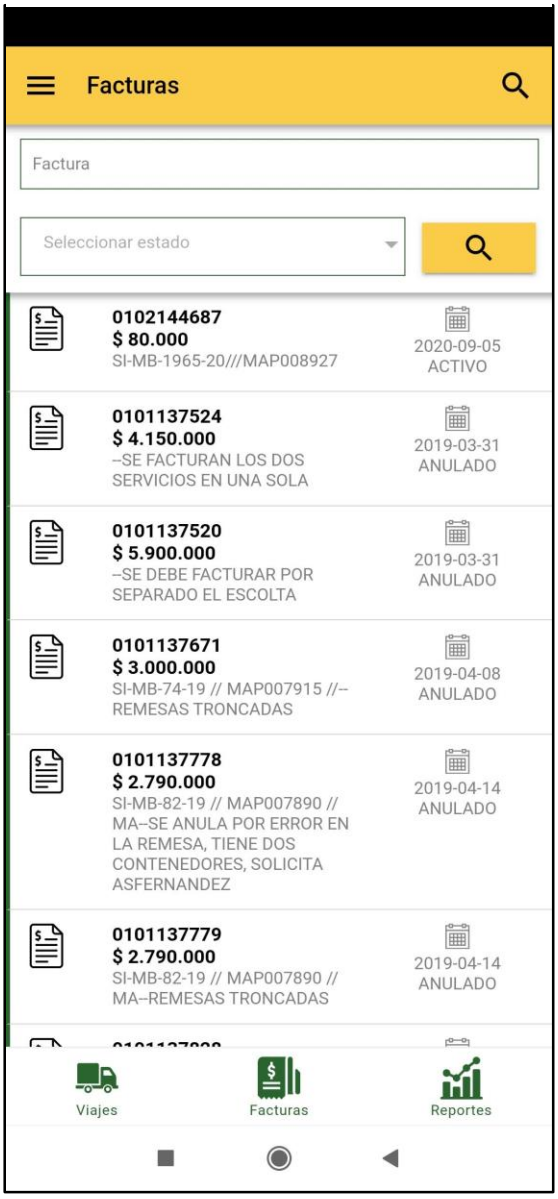


Ilustración 16. Interfaz gráfica 9

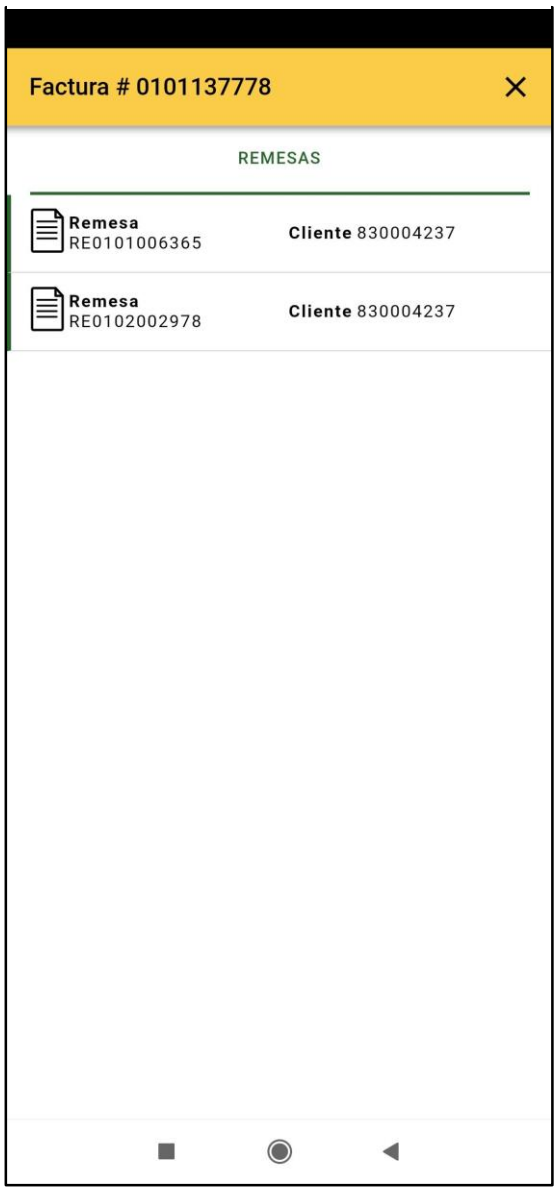
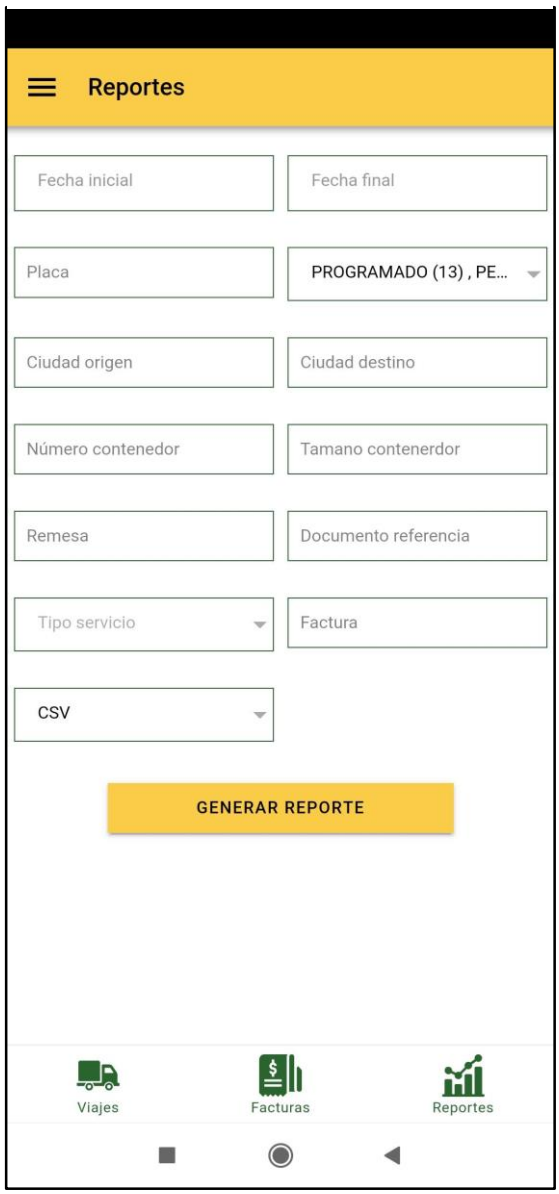


Ilustración 15. Interfaz gráfica 10



4.4.2 Descripción general de las interfaces gráficas de usuario. GUI´s

Las interfaces gráficas se desarrollaron centradas en el usuario final, enfocadas en maximizar la usabilidad y la experiencia del usuario. El objetivo final de este diseño de interfaz de usuario es hacer que la interacción entre el usuario y el sistema sea tan simple y eficiente como sea posible, es por ello que en su creación se tomaron en cuenta algunos factores como el logo de la compañía, los colores empresariales y su manejo intuitivo.

La aplicación se basa en tres módulos principales mediante los cuales pueden obtener información de acuerdo con su interés, los botones y campos de ingreso de información tienen una estructura de fácil lectura lo cual permite que la interacción con el usuario sea amigable y ágil al momento de usar filtros de búsqueda o generar información.

Capítulo 5

Conclusiones, recomendaciones y trabajo futuro

5.1 Conclusiones

5.1.1 Generalidades en la gestión de proyectos

Al iniciar el trabajo de desarrollo de un producto software, es importante tener una visión del contexto real y el ambiente tecnológico en el que se desarrollará para tomar decisiones acerca de cómo ajustar algunos procesos que se van ejecutando en el camino. De igual forma, se debe tener un conocimiento del equipo de trabajo que estará en la implementación.

En una instancia inicial, es preciso analizar los puestos de trabajo, los recursos hardware y software y otras variables que pueden alterar el rendimiento del equipo de trabajo tales como los factores ambientales (luz, ruido, temperatura) y los factores ergonómicos (diseño del puesto de trabajo, antropometría, visibilidad, iluminación) para optimizar los recursos, mejorar los rendimientos de las personas (Poelman & Keyson, 2008) y administrar eficientemente el consumo energético.

Para ello es preciso analizar en qué etapas del proceso se pueden ajustar los tiempos de desarrollo por tarea sin afectar la calidad del producto final y lograr una reducción en los tiempos de producción

5.1.2 Resultado del trabajo con Scrum

Realizar un desarrollo con el marco de trabajo scrum, haciendo un análisis minucioso a cada etapa del proceso nos lleva a las siguientes reflexiones:

Un proyecto de desarrollo se podrá realizar más fácilmente si se hace un mayor énfasis en etapas tempranas como el sprint backlog para ajustar aquellas tareas que son susceptibles de modificar.

Efectivamente, cada evento en scrum es la gran oportunidad para adaptar los artefactos y minimizar la necesidad de reuniones diarias y ayudan a reducir la complejidad.

El marco de trabajo scrum permite obtener productos de calidad favoreciendo el trabajo de los profesionales con las herramientas metodológicas que conllevan a un buen control sobre los subprocesos y mejoran el rendimiento del equipo de trabajo que se verá reflejado en la satisfacción del cliente y permiten que el producto evolucione rápidamente hacia versiones posteriores mejorando su escalabilidad.

Scrum requiere un conocimiento profundo de los requisitos del usuario para entender más rápido los problemas asociados al sistema original y diseñar fácilmente las soluciones. Para esto es muy importante escuchar bien al cliente y documentar las historias de usuario para eliminar la dependencia de las apreciaciones del desarrollador que podrían conducir a imprevistos de diseño

El rol del product owner tiene una importante responsabilidad que está asociada con el establecimiento de una efectiva comunicación con el cliente para mantener un control de los aspectos en los cuales cada subproducto debe mejorar de tal manera que las evaluaciones periódicas se reflejen en el resultado final tanto de calidad como de usabilidad del producto.

Estas evaluaciones deben mantenerse como un proceso iterativo que vaya resolviendo cada situación en el momento y no dejarse para el final.

Las compañías que trabajan en la industria del software se ven involucradas en el dinamismo propio de los acelerados avances tecnológicos de su entorno, por lo tanto, se deben adaptar a los cambios de una forma rápida para mantener su posición en el mercado. Estos cambios inclusive tienen relación con las medidas que asume el Gobierno en temas de adaptación al cambio climático y las acciones que se deben emprender para hacer frente a las consecuencias adversas que aumentan la temperatura del planeta.

En el proceso de implementación de scrum, se tomó en cuenta cada evento que compone este marco de trabajo, logrando realizar los pasos más relevantes para lograr un desarrollo ágil de alta calidad.

Debido a que el equipo de trabajo Scrum fue reducido, se dio más importancia a la planeación de cada Sprint con base en el product backlog y en el incremento que se tenía hasta determinado momento.

Los Daily scrum se realizaron cada vez que fueron necesarios, ya que las tareas estaban definidas y asignadas, esto nos permitió optimizar en tiempo.

Por otra parte, los Sprint Review en algunas ocasiones no se realizaron con el equipo completo, ya que el cliente y el integrante del área de TIC no tuvieron suficiente disponibilidad, por lo tanto, la comunicación con el cliente se estableció mediante el Toolproject para sus comentarios y el envío de la APK periódicamente para las respectivas pruebas hasta su aprobación final.

Gestión de los Riesgos: Los riesgos se evaluaron en cada sprint. El cliente revisa los resultados preliminares y hace observaciones de mejora lo cual permite hacer una gestión de los riesgos minimizando el impacto de los mismos en el proyecto.

La planeación se revisa y se reconstruye el product backlog cada vez que es necesario para mantener el control todo el tiempo.

5.1.3 Como adaptar Scrum a un proceso de desarrollo en la industria del software.

Finalizado este proyecto consideramos importante resaltar las partes del proceso scrum en las cuales fue posible adaptar el marco de trabajo con el fin de ahorrar esfuerzos y evitar la repetición de algún subproceso de manera innecesaria. Una guía importante para esta adaptación es el libro (SCRUMstudy, 2016) en su página 24 que incluimos a continuación, por supuesto con nuestros aportes.

El punto de partida es la experiencia del Team Scrum que puede ir adaptando cada proceso en la medida que se van ejecutando, implementando mejoras.

5.1.3.1 En las reuniones diarias - Daily scrum.

Los miembros del equipo con mayor experiencia guían el desarrollo de la reunión orientando la toma de decisiones para hacer más rápido estos momentos, incluso se decide acerca de la conveniencia de hacer reuniones diarias y con más tiempo entre ellas.

5.1.3.2 En los Risk identification.

La gestión de riesgos es una tarea que puede hacerse a lo largo del proyecto, priorizando aquellos que requieran atención inmediata y manteniendo una lista de pendientes para resolverlos sin interferir en el proceso de desarrollo.

5.1.3.3. En los change request.

En las solicitudes de cambio, los cambios se discuten y se aprueban durante el proceso de desarrollo creando una lista priorizada de pendientes.

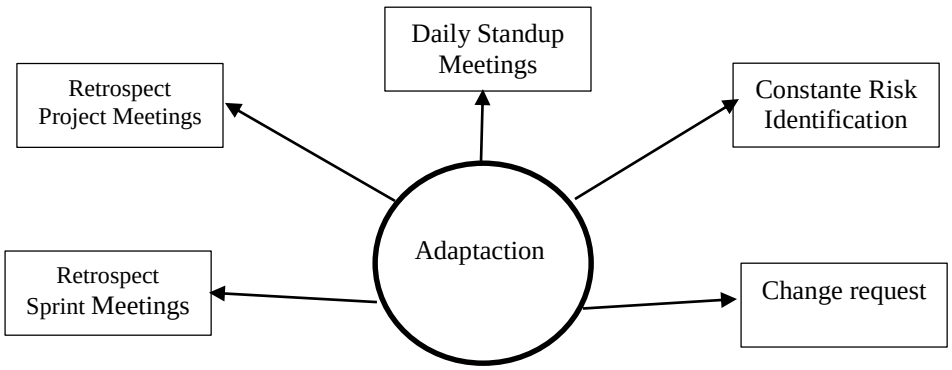
5.1.3.4. En el Sprint Retrospective.

En este proceso se pueden acordar las mejoras adicionales con base en los demos y en las validaciones de cada sprint.

5.13.5 En las reuniones de retrospectiva del proyecto.

Se documentan las lecciones aprendidas y se buscan oportunidades de mejora de procesos.

Ilustración 17. Adaptación del proceso Scrum



Fuente: Una guía para el conocimiento de Scrum. Página 25. Modificado por los autores

5.1.4 Procesos en los cuales se ahorra tiempo y esfuerzos

Las reuniones diarias no fue necesario realizarlas periódicamente, sino cuando se consideró oportuno.

Análisis de los tiempos de desarrollo

5.1.4.1 Tiempos teóricos de desarrollo de acuerdo con el marco de trabajo scrum

Tabla 8. Tiempo estimado por la metodología

Tiempo estimado por la metodología Horas				
Sprint	***Planing***	***Daily***	***Review***	***Retrospective***
1	8	5	4	3
2	8	5	4	3
3	8	5	4	3
4	8	5	4	3
Totales	32	20	16	12
		Total horas estimadas		80

5.1.4.2 Tiempos reales empleados en el proyecto

Tabla 9. Tiempos reales empleados en el proyecto

Tiempo real empleado en el proyecto Horas o Minutos				
Sprint	***Planing***	***Daily Minutos***	***Review***	***Retrospective***
1	8	105	2	2
2	4	120	3	2
3	4	120	2	2

4	4	120	2	2
Totales	20	7,75	9	8
		Total horas reales		44,75

5.1.4.3 Tiempo ahorrado en el proyecto

Tiempo ahorrado en el proyecto: 35,25
Horas de trabajo

5.1.4.4 Cálculo del CO_2eq que se disminuye con el ahorro energético

Tabla 10. Dispositivos consumidores

Factor de emisión según UPME: 0,21 $KgCO_2e/Kwh$

DISPOSITIVOS CONSUMIDORES DE ENERGÍA ELÉCTRICA	*Potencia vatios*	Tiempo Horas	*Consumo en Kw/h*	$KgCO_2$eq Mitigado
Desarrollador 1				
PORTATIL ASUS vivoBook S14 X41 1U	220	35,25	7,76	1,63
MONITOR SAMSUNG 22"	50	35,25	1,76	0,37
CELULAR SAMSUNG GALXY J5 (APK)	36	35,25	1,27	0,27
Focos LED (2 x 15w c/u)	30	35,25	1,06	0,22
Conectividad (router)	25	35,25	0,88	0,19
Desarrollador 2				
PORTATIL MAC air 2017	200	35,25	7,05	1,48
MONITOR SAMSUNG 22"	50	35,25	1,76	0,37
CELULAR SAMSUNG GALXY J5 (APK)	36	35,25	1,27	0,27
Focos LED (2 x 18w c/u)	36	35,25	1,27	0,27
Conectividad (router)	25	35,25	0,88	0,19
Totales	708		24,96	5,24

Resumen de los cálculos.

Tabla 11. Resumen de cálculos

Consumo estimado según dispositivos	708	watios
Tiempo estimado según scrum	80	horas
CO2eq Generado según scrum	11,89	$KgCO_2eq$
Consumo estimado real	708	watios
Tiempo Estimado real	44,75	horas
CO2eq Generado real	6,65	$KgCO_2eq$
Co2eq Disminuido (mitigado)	5,24	$KgCO_2eq$

Datos importantes:

Una tonelada de CO_2eq equivale al contenido de una piscina de 500m^3 de agua
5.24 $KgCO_2eq$ equivale al contenido de una piscina de 2.62 m^3 de agua

Los cálculos se realizaron con base en el protocolo de gases de efecto invernadero GHGProtocol (World Bussines Council for Sustainable Development, 2015) y la norma NTC 5947 (ICONTEC, 2012) para gases de efecto invernadero y el factor de emisión de la Unidad de Planeación Minero-Energética UPME (UPME CIAME, 2019) del Ministerio de Minas y Energía de Colombia.

Los cálculos corresponden a la huella de carbono corporativa de alcance 2.
La ecuación utilizada para el cálculo es la referida en el GHGProtocol: (World Bussines Council for Sustainable Development, 2015)

HdC=D.A. * F.E. * W.G.P.
Donde:
HdC: huella de carbono que se desea calcular
D.A.: Son los datos de actividad expresados en kw/h (kilovatios por hora)
F.E.: Es el factor de emisión publicado por la UPME, expresado en $KgCO_2eq/Kwh$
W.G.P.: Potencial de Calentamiento Global. Para el caso del CO_2 es 1

CO2eq: Dióxido de carbono equivalente (en unidades de contaminación)

Es preciso decir que en la medida en que podamos incorporar ahorros de esfuerzos y energía en los procesos industriales de desarrollo de software estaremos aportando acciones para mitigar emisiones de CO2 que tanto daño hacen al planeta. Estas acciones que se sustentan en el uso de la tecnología están consideradas como buenas prácticas de Green IT (It Green, 2018) en la organización.

5.2 Recomendaciones.

5.2.1 Porqué recomendamos scrum.

Además de las ventajas que representa trabajar con el marco scrum, en cada una de sus fases, consideramos importante mostrar algunas diferencias con los modelos tradicionales que pueden ser tenidas en cuenta para tomas decisiones cuando se trate de escoger un modelo para desarrollo.

La siguiente tabla contiene algunas de las diferencias significativas entre los modelos tradicionales de gestión de proyecto y scrum, que se encuentran en el libro referenciado como: (SCRUMstudy, 2016)

Tabla 12. Scrum Vs Gestión de proyectos tradicional

	Scrum	**Gestión de proyectos Tradicional**
El énfasis está en	Personas	Procesos
Documentación	Sólo mínima según se requiera	Exhaustivo
Estilo de procesos	Iterativo	Lineal
Planificación por adelantada	Baja	alta
Priorización de los requisitos	Según el valor del negocio y	Fijo en el plan del proyecto

	Regularmente actualizada	
Quallity Assurance	Centrada en el customer	Centrada en el proceso
Organización	Auto – organizada	Gestionada
El estilo de gestión	Descentralizado	Centralizado
Cambio	Las actualizaciones de Prioritized Product Backlog	Sistema forma de Gestión del Cambio
Liderazgo	Collborative, Servant Leadership	Mando y control
La medición del rendimiento	El valor del negocio	Plan de la conformidad
Return on Investiment ROI	Al comienzo y a lo largo del proyecto	Al fin del proyecto
Participación del customer	Alta durante todo el proyecto	Varía en función del ciclo de vida del proyecto

Fuente: Una guía para el conocimiento de Scrum. Página 20

5.3 Trabajo futuro.

Como desarrollos para el futuro enunciamos a continuación algunos módulos que podrían complementar el aplicativo:

- **Mapa**: dentro de modulo viaje exponer un mapa donde el usuario pueda visualizar la ubicación del vehículo que transporta su mercancía, indicando el tiempo estimado de entrega.

- **Módulo de recogida**: crear un módulo donde el usuario pueda solicitar la recogida de la mercancía en un punto específico detallando peso, producto, paquete y destinatario.

- **Módulo cotización**: crear un módulo donde el usuario pueda consultar el valor de un servicio de acuerdo con diferentes parámetros.

6. Referencias consultadas

Banco Mundial. (1 de Diciembre de 2017). *Banco Mundial. BIRF AIF. Eficiencia Energética.* Obtenido de https://www.bancomundial.org/es/results/2017/12/01/energy-efficiency

BBVA: Eficiencia energética. (2015). Obtenido de https://www.bbva.com/es/sostenibilidad/que-es-la-eficiencia-energetica-y-como-se-calcula/

Climate Group. (2020). *Smart 2020- Climate Group*. Obtenido de https://www.compromisorse.com/upload/estudios/000/36/smart2020.pdf

Energía & Sociedad. (2000). *Energía y Sociedad. Las claves del Sector Energético*. Obtenido de http://www.energiaysociedad.es/

Francia Huambachano, J. (25 de Septiembre de 2017). *Scrum*. Obtenido de https://www.scrum.org/resources

ICONTEC. (2012). *ICONTEC. 5947.* Obtenido de ESPECIFICACIÓN PARA EL ANÁLISIS DE EMISIONES Y REMOCIONES DE GASES DE EFECTO INVERNADERO DURANTE EL CICLO DE VIDA DE BIENES Y SERVICIOS: https://tienda.icontec.org/gp-especificacion-para-el-analisis-de-emisiones-y-remociones-de-gases-de-efecto-invernadero-durante-el-ciclo-de-vida-de-bienes-y-servicios-ntc5947-2012.html

It Green. (2018). *Efficiency & Sustainab le Technology.* Obtenido de http://www.itgreen.es/que-es-green-it

Poelman, W., & Keyson, D. (2008). *Design Processes.* Netherlands: IOS Press BV.

Red Hat. (2000). *Red Hat. ¿Qué es la metodología ágil?* Obtenido de https://www.redhat.com/es/devops/what-is-agile-methodology

SCRUMstudy. (2016). *Una guia para el conocimiento de Scrum.* Phoenix: SCRUM Study VMEdu Inc.

Shcwaber, K., & Sutherland, J. (Sedptiembre de 2020). *La Guia de Scrum.* Obtenido de https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf

UPME CIAME. (2019). *Cálculo del factor de emisión de CO2 del SIN.* Obtenido de https://www1.upme.gov.co/siame/Paginas/calculo-factor-de-emision-de-Co2-del-SIN.aspx

World Bussines Council for Sustainable Development. (2015). *Protocolo de Gases Efecto Invernadero.*

M. Adnan y M. Afzal, "Sistema de estimación de esfuerzo multiagente basado en ontología para el método Scrum Agile", en IEEE Access, vol. 5, págs.25993-26005, 2017, doi: 10.1109 / ACCESS.2017.2771257.

Babenko, V., Lomovskykh, L., Oriekhova, A., Korchynska, L., Krutko, M., & Koniaieva, Y. (2019). Features of methods and models in risk management of IT projects. Periodicals of Engineering and Natural Sciences, 7(2), 629-636. doi:10.21533/pen.v7i2.558

Bott, M., & Mesmer, B. (2020). An analysis of theories supporting agile scrum and the use of scrum in systems engineering. EMJ - Engineering Management Journal, 32(2), 76-85. doi:10.1080/10429247.2019.1659701

Douglass, B. P. (2015). Agile systems engineering. Agile systems engineering (pp. 1-429) doi: 10.1016/C2014-0-02102-8 Retrieved from www.scopus.com

Strenge, B., & Schack, T. (2020). AWOSE - A process model for incorporating ethical analyses in agile systems engineering. Science and Engineering Ethics, 26(2), 851-870. doi: 10.1007/s11948-019-00133-z

Dorantes-Jiménez, S. -. Letelier-Torres, P. O., & Pérez-Castillo, Y. J. (2020). Importance of support in the implementation of agile practices in work teams. Paper presented at the IMSCI 2020 - 14th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings, 126-129. Retrieved from www.scopus.com

Roza Albareta, A., & Mursanto, P. (2019). Design of standard operating procedure for requirement engineering in software development: Case study data processing integration