UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA

Título: **SISTEMA INFORMÁTICO PARA EL MONITOREO DE LA PRODUCCIÓN DE CILINDROS DE GAS LICUADO FABRICA CONFORMAT ´´NOEL FERNÁNDEZ´´**

.

Trabajo de Diploma para optar por el título de Ingeniero en Informáticas

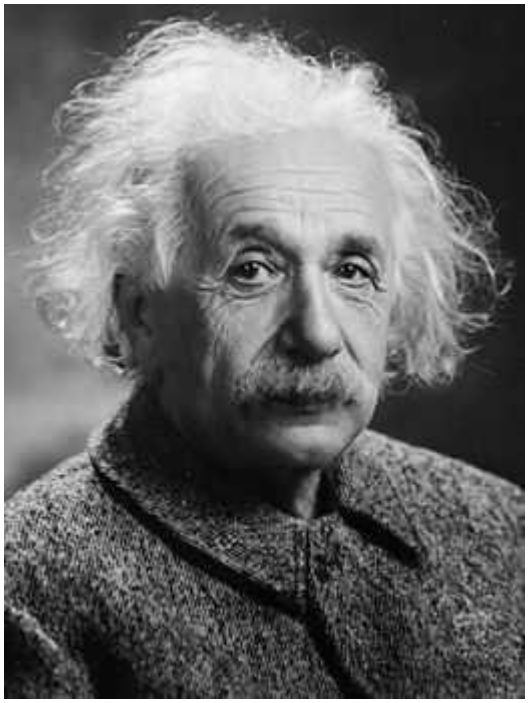
Autor: Carlos Adrian Alonso Infante

Tutores: MSc. Briseida Blanco Alfonso

MSc. Darien Menéndez Molina

Matanzas, Junio de 2019

Pensamiento



Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.

Albert Einstein

DEDICATORIA

A ustedes que me han apoyado y aconsejado en todo momento, que siempre están a mi lado sin importar mis acciones, que las dificultades las convierten solo en una palabra sin importancia; a ustedes que me aman sin pedir nada a cambio y que nunca tendré cómo saldar esa gran deuda de amor y cariño; a ustedes que hacen que mi vida sea sencilla y llena de esperanzas; a ustedes gracias por todo lo que me han dado y sé que continuarán dándome…

A ustedes; Mamá, Papá, Abuelas y familia les dedico todos mis logros, sueños, esperanzas y mi amor y cariño.

GRACIAS!

AGRADECIMIENTOS

A Dios que siempre ha estado conmigo y es el que me ha permitido llegar aquí.

A mis padres y familia por darme todo su apoyo incondicional y estar a mi lado en todo momento.

A todos los amigos, que de una manera u otra han estado ahí, cuando los he necesitado.

A mi tutor Darien Menéndez por darme su apoyo incondicional.

A todos…, GRACIAS!!!

DECLARACIÓN DE AUTORÍA

Declaro que soy autor de este trabajo y autorizo a la Universidad de Matanzas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Carlos Adrian Alonso Infante

MSc. Briseida Blanco Alfonso

_____ _____

Firma del Autor

Firma del Tutor

OPINIÓN DEL CLIENTE DEL TRABAJO DE DIPLOMA

Al iniciar el proyecto el cual tiene como título: Sistema informático para el monitoreo de la producción de cilindros de gas licuado Empresa Conformat ´´Noel Fernández´´ no se estaba seguro que se quería y que lograríamos en realidad, esa era la verdadera incertidumbre, al avanzar, fueron apareciendo nuevas ideas y desechando otras, convertimos el proyecto de algo ideal a algo objetivo. Finalizado el proyecto, estamos aún más contentos y satisfechos con el resultado obtenido que con lo que se esperaba, el trabajo superó todas nuestras expectativas porque es atractivo, sencillo de usar, adaptable a cualquier dispositivo que pueda poseer un trabajador, totalmente configurable y ligero, compatible con nuestra tecnología y totalmente funcional. En conclusión, ahora CONFORMAT cuenta con una potente aplicación Web que garantiza y agiliza el proceso productivo con mayor eficiencia y calidad como resultado de un excelente trabajo realizado por su autor, Carlos Adrian Alonso Infante que actualmente está optando por el título de Ingeniero en Informática con la presente herramienta.

Muchas felicidades y gracias por todo.
CONFORMAT

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

La tesis titulada "**SISTEMA INFORMÁTICO PARA EL MONITOREO DE LA PRODUCCIÓN DE CILINDROS DE GAS LICUADO FABRICA CONFORMAT ´´NOEL FERNÁNDEZ´´** presentada por el estudiante Carlos Adrian Alonso Infante, en opción al título de Ingeniero Informático, desarrollada en la Universidad de Matanzas, posee actualidad, contribuyendo a la informatización de nuestra sociedad.
El tutor de este trabajo de diploma considera que, durante su ejecución, el estudiante mostró las cualidades que a continuación se detallan:

Independencia, tenacidad y espíritu de investigación en el desarrollo de su trabajo. Aplicó para dar solución a su problema tecnologías novedosas, pero de probada efectividad en la solución de situaciones similares. Aportó soluciones importantes que demuestran su madurez como investigador. Fue consecuente con los aspectos tanto metodológicos como de la investigación científica propiamente. Esto le permitió una feliz culminación del método desarrollado y en la adecuada documentación.

En el trabajo se aprecia rigor, manifestado desde el tratamiento de los definiciones estudiadas y referenciadas en la bibliografía, hasta las conclusiones, lo que ha contribuido a la correcta solución de los problemas encontrados.

El trabajo tiene gran calidad puesto que se emplearon patrones de diseño acorde a la solución de la propuesta, se obtuvo una aplicación con funcionalidades útiles y con usabilidad.

Por todo lo anteriormente señalado, considero que el estudiante Carlos Adrian Alonso Infante reúne los requisitos para el título de Ingeniero Informático y espero le sea otorgada la mejor calificación de este Tribunal.

_____

MSc. Darien Menéndez Molina
Dpto. Informática, Universidad de Matanzas

Junio, 2019

RESUMEN

La industria moderna exige grandes niveles de producción como de calidad. La inspección a la calidad es de vital importancia y su debido control es de gran valor para la industria. La empresa CONFORMAT “Noel Fernández” única productora en Cuba, de cilindros de gas licuado de varias dimensiones que abaste a la población, realiza un proceso de perfeccionamiento empresarial, en la cual se desea incrementar el ritmo de producción y calidad de sus productos. Eso requiere de automatizar el proceso de inspección a la calidad que hoy se realiza manualmente y en etapas tardías del proceso de producción, de manera que se pueda detectar defectos en los productos de forma temprana y puedan ser informados al instante. Para ello se desarrolla un Sistema Informático para el monitoreo y gestión de la información referente a la inspección a la calidad de la producción que informan dispositivos insertados en la fábrica. El sistema monitorea los parámetros evaluados por otros dispositivos de inspección automática distribuidos en una red de datos, permitiendo llevar un control en tiempo real de la producción e identificar rápidamente los productos defectuosos. Facilita el proceso productivo, la emisión de reportes, el traspaso de información entre turno y almacena información referente al proceso productivo que puede ser consultada en cualquier momento. Se provee de un método de clasificación de la calidad del producto mediante colores. El sistema es implementado bajo plataforma web, de fácil despliegue y operatividad.

Abstract

Modern industry demands high levels of production as quality. The quality inspection is of vital importance and its control is urgent. The company CONFORMAT "Noel Fernández" is the only producer in Cuba of gas cylinders of various dimensions that supplies the entire population, a process of business improvement is carried out, in which the production rate and quality can be improved, This is required to automate the inspection process to the quality that is now done manually and in the later stages of the production process, in such a way that weaknesses in the products can be detected early and be informed instantly. To do this, a computer system for the monitoring and management of information must be used. The system monitors the parameters evaluated by other automatic inspection devices distributed in a data network, implemented in a real-time control of the production and quickly identify the defective products. It facilitates the production process, the issuance of reports, the transfer of information between shifts and stores information regarding the production process that can be consulted at any time. A method of classifying product quality by colors is provided. The system is implemented under a web platform, easy to deploy and operational.

Contenido

INTRODUCCIÓN

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos.

Una sociedad que aplique la informatización en todas sus esferas y procesos será más eficaz, eficiente y competitiva. Es evidente que para los países subdesarrollados resulta un reto el logro de este propósito, ya que su problemática fundamental está en lograr la supervivencia de sus pueblos.

Más de una vez en los años de la Revolución cubana, se ha reconocido la importancia estratégica de la informática para la sociedad. Más de una vez se han hecho planes estratégicos de desarrollo. Esta vez, sin embargo, ese reconocimiento y esa estrategia se ubica en un momento histórico en que un gran número de computadoras de variado tipo se encuentran instaladas en el país, en un momento en que un considerable capital humano relacionado con las diferentes especialidades informáticas como el software y el hardware y de comunicaciones, está en funciones, en un momento en que muchas entidades están basando su tratamiento de la información en soluciones informáticas y finalmente, en un momento en que la cultura general informática ha penetrado en todos los rincones de la sociedad, pues hasta el ciudadano más alejado de los centros de gestión conoce qué es una computadora y para qué sirve.

Cierto proceso de informatización llego a la Empresa de Conformación de Matanzas “Noel Fernández” en la cual la cantidad de extintores, como promedio anual, oscila alrededor de 18705 extintores por año, lo que pasa a tener un nivel de importancia considerable, no sólo por la responsabilidad del artículo elaborado, sino también por ser una producción seriada y que se exporta a varios países.

Durante el proceso de la soldadura ecuatorial de tales cilindros, se puede apreciar que este se ejecuta de una sola vez y sin refuerzo, y los valores de regímenes de soldadura que se emplean son los calculados atendiendo a tecnología muchas veces defectuosa, no siendo los más productivos. Debido a que esta producción es de gran importancia para la Empresa y para el país, es necesario monitorear los valores óptimos de los regímenes de soldadura para obtener una mayor productividad, reducir los costos y tener como resultado un producto con mejor calidad y sin errores.

En correspondencia con la situación problémica anterior se formula como **problema científico**:

¿Cómo monitorear la producción del cilindro de gas en la empresa Noel Fernández?

Según el anterior problema científico permite formular como **hipótesis**: Si se desarrolla una aplicación para el monitoreo de la producción del cilindro de gas entonces se produce un mejoramiento en la efectividad de la detección de errores y un aumento de la calidad del proceso de producción.

El **objeto de estudio** es el proceso de producción de las partes y terminación del producto final y la automatización de dicho proceso es el **campo de acción** de la investigación.

Variable independiente: software para el monitoreo y gestión de los productos y servicios de COMFORMAT ¨ Noel Fernández ¨

Variable dependiente: el monitoreo y gestión de los productos y servicios de COMFORMAT ¨ Noel Fernández ¨

Su **objetivo general** es la elaboración de una aplicación para el monitoreo de la producción del cilindro de gas y la gestión de los datos para la elaboración del reporte final.

Y tiene como **objetivos específicos**: Mejorar el control de la producción mediante la automatización del proceso, disminuir el tiempo de obtención de datos y automatizar el proceso de obtención de datos de la producción de los cilindros de gas.

Para darle solución al problema planteado y cumplir los objetivos trazados se utilizaron en la investigación los siguientes métodos:

Dentro de los **métodos teóricos**:

- Método de análisis histórico - lógico.
- Método de análisis y síntesis.
- Método inductivo - deductivo.

Los **métodos empíricos** fueron utilizados por medio de las siguientes técnicas:

- Observación científica.
- Entrevistas.

Se empleó una metodología de desarrollo de software ágil:

Un lema recurrente en Scrum es “inspección y adaptación”. Dado que el desarrollo conlleva de forma inevitable aprendizaje, innovación y sorpresas, Scrum enfatiza dar pequeños pasos en el desarrollo, inspeccionando tanto el producto resultante como la eficacia de las prácticas actuales, y a continuación

adaptar los objetivos respecto al producto y las prácticas de los procesos. *Repetir indefinidamente*. (Pete Deemer, 2012)

Se pueden señalar como **beneficios** que se garantizan un mejor control de los datos y una mejor obtención de reporte diario por turnos. También se obtienen datos más detallados del proceso de producción y al instante mejorando su velocidad. Con un software de este tipo se facilitará una herramienta de gran **aporte práctico** para la empresa, puesto que garantiza el acceso a la información actualizada a los trabajadores para contribuir al control de la producción y agilizar la misma disminuyendo costos en pruebas.

Como **resultados esperados** se quiere que la empresa CONFROMAT "Noel Fernández" cuente con una herramienta informática capaz de garantizar el acceso a la información actualizada de sus procesos y satisfacción de sus productos y servicios.

En consecuencia, con lo anteriormente planteado, el documento queda estructurado como sigue:

Capítulo I: Marco teórico-referencial, se plantean las definiciones fundamentales asociadas al campo de acción. Se hace un estudio sobre el estado del arte, las tendencias y tecnologías actuales que serán usadas. Se exponen las características fundamentales de los lenguajes de programación, los sistemas de bases de datos y las características fundamentales de las metodologías de desarrollo de software ágiles.

Capítulo II: Análisis, Diseño de la Solución Propuesta y Construcción, se argumenta la solución que se propone al problema de investigación, presentando una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software SCRUM. Se realiza además un estudio de los beneficios tangibles e intangibles como resultado de la realización del proyecto de software. Se construye la solución propuesta, presentando una planificación por iteraciones.

Capítulo III: Validación de la Solución Propuesta, se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio de los clientes y los propios de la metodología de software.

Finalmente, se presentan las **Conclusiones y Recomendaciones** de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

Asimismo, quedan recogidas las **Bibliografías** empleadas y **Anexos** que fueron necesarios para el desarrollo de todo el trabajo y un mejor entendimiento del mismo.

CAPÍTULO I: Marco teórico-referencial

Introducción

En el mundo entero son empleados software para la gestión empresarial con el objetivo de garantizar el acceso a la información actualizada.

La aplicación de la informática en la gestión empresarial es una estrategia a considerar. Además se definen también los principales conceptos relacionados con el objeto de estudio y se describen de manera precisa los diferentes elementos seleccionados para darle respuesta a la problemática planteada. Por otra parte, se realiza un análisis del estado del arte, tomándose en cuenta sistemas existentes vinculados al campo de acción, tendencias y tecnologías sobre las que se basa la propuesta a implementar.

Antecedentes del trabajo

Durante la búsqueda de software a fines con la temática, se encuentra una aplicación para el monitoreo y control de procesos industriales basada en el estándar de comunicaciones OPC. Es una aplicación de software tipo cliente, capaz de realizar actividades de monitoreo y control de procesos industriales empleando el protocolo de comunicación “OLE para control de procesos”, OPC. Se utilizó el lenguaje C# y la metodología Extreme Programming (XP). El software permite al usuario, interactuar con datos que pueden provenir de un proceso real o de una simulación, a través de algoritmos de control personalizados realizados en un lenguaje de fácil manejo (VBScript y JavaScript), y adicionalmente genera un registro histórico de variables OPC que puede ser recuperado en cualquier momento. Al fundamentarse en el estándar OPC, es posible agregar dispositivos de diferentes fabricantes a medida que estos son adquiridos e incorporados al proceso. El programa puede ser utilizado en aplicaciones de automatización de procesos e instrumentación virtual. La operación del sistema fue verificada de forma local y remota por medio de una red LAN. La aplicación desarrollada presenta interfaces amigables y una arquitectura abierta, capaz de crecer o adaptarse según las necesidades cambiantes de la empresa o instalación industrial. (Angel Villegas, 2008)

El uso de este software no es aplicable a la problemática propuesta, porque es de tipo propietario y no se adapta al modelo de negocio de la empresa. Normalmente el software a pedido sólo implementa funciones que correspondan con el modelo de negocio empresarial para el cual fueron contratados, pero su implementación sirve como referencia para la implementación propuesta.

Objeto de estudio

El objeto de estudio es el proceso de producción de las partes y terminación del producto final, por tanto, surge la necesidad de hacer una descripción más detallada de estos elementos que se pretenden transformar y perfeccionar.

Caracterización de CONFORMAT

La Empresa de Conformación de Matanzas “Noel Fernández” en lo adelante CONFORMAT, situada en Covadonga # 10: Playa, Matanzas. Cuba. Creada con el objetivo fundamental de producir y comercializar, de forma mayorista y en ambas monedas, recipientes a presión y sus componentes, artículos de uso doméstico e industrial a partir de la conformación de metales tanto para el mercado nacional como para la exportación; brindar servicios de mantenimiento y reparación de recipientes a presión y ofrecer servicios de mantenimiento industrial y maquinado. Hoy en día la empresa está inmersa en una gran producción de cilindros para la comercialización nacional y exportación lo que demanda una gran calidad en cuanto al producto que se oferta. La Empresa cuenta con varios talleres como son el de cilindro del que se sirve todo el proceso de este artículo, el de maquinado que principalmente se dedica a elaborar piezas para la reparación de otros equipos de la empresa y el de estampado, que es donde se realizan las etapas de producción de los extintores. En el taller de cilindros se realizan una serie de procesos tales como: el embutido de la tapa y el fondo que forman el cuerpo, corte de vuelos de la tapa y el fondo, marcado del fondo, conformado de patas, perforado de la tapa, pestañado de la tapa, corte de tira de presilla, conformado de presilla, soldadura del bushing a la tapa, soldadura ecuatorial, soldadura de la presilla, prueba hidráulica, cepillado y limado del cuerpo, por último pasan por el horno de recocido para eliminar las tensiones residuales, la granalladora para su limpieza y finalmente son pintados y embalados.

Análisis crítico de la ejecución actual de este proceso y las causas que originan la situación problémica, así como sus consecuencias

El proceso de control de los datos y conformación del informe se realiza de forma manual y es ejecutado por los jefes de brigada, los datos son obtenidos por los demás trabajadores, con más o menos conocimiento del tema. Todo esto conlleva a que no se informe de forma correcta y uniforme, y trae como consecuencia gastos innecesarios tanto en recursos como en tiempo.

Metodología de la investigación

Métodos teóricos empleados

- **El método de análisis histórico y el lógico**: permitió estudiar la trayectoria y desarrollo de las aplicaciones web de gestión empresarial, así como el proceso de los productos y servicios realizado por CONFORMAT.

- **Los métodos de análisis y de síntesis**: fue precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
- **Inducción-deducción**: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

Métodos empíricos empleados

Para la utilización de los métodos empíricos se ponen en práctica las siguientes técnicas:

- **La observación científica**: acompañó la investigación desde los primeros momentos, a través de la cual se conoció el estado y proceder de todo el proceso, se obtuvo la información primaria acerca de los objetos investigados.
- **La entrevista**: aportó datos esenciales a la investigación puesto que el entrevistado es la persona que propuso el desarrollo del sistema. Fue útil en distintos momentos de la investigación; fundamentalmente al inicio, cuando se realizó el modelado del negocio y la lista priorizada de requerimientos de software.

Tendencias tecnológicas

En la actualidad existe una tendencia creciente por la utilización de metodologías ágiles de desarrollo de software, así como el uso de software libre en el desarrollo de todo tipo de proyectos y cada vez más proyectos van destinados a la Web. Esta creciente tendencia de usar software libre en el desarrollo de aplicaciones Web se basa básicamente en la libertad de uso del software y los beneficios económicos que estos devengan.

Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos que deben seguirse para llevar a cabo el desarrollo de software con calidad, éstas brindan un conjunto de detalles organizativos, añadiendo rigurosidad y normas, permitiendo que los integrantes de un equipo de desarrollo puedan seguir un criterio común a la hora de realizar las tareas durante el desarrollo de un software. (Sánchez, 2003)

Actualmente existen dos grandes grupos de metodologías de desarrollo, las metodologías tradicionales y las metodologías ágiles; las primeras se centran en el uso exhaustivo de documentación durante todo

el ciclo de vida del proyecto, mientras que las segundas dan mayor importancia a la capacidad de respuesta a los cambios. A continuación, se presenta una breve comparación entre ellas.

Metodologías ágiles dan mayor valor a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar software que funcione, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan.

Metodologías tradicionales llevan un control estricto del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Centran su atención en llevar una documentación exhaustiva de todo el proyecto y en cumplir con un plan de proyecto.

SCRUM es una metodología ágil de desarrollo de software, una manera de afrontar los proyectos de creación de aplicaciones de forma iterativa, rápida y eficaz, desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. (**Ver Anexo 1**)

Scrum (n): Es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. (Beck K., 2005)

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 varias técnicas y procesos. 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 y para su uso.

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: **transparencia**, **inspección** y **adaptación**.

Los **Equipo Scrum (Scrum Team)** consiste en un **Dueño de Producto (Product Owner)**, el **Equipo de Desarrollo (Development Team)** y un **Scrum Master** que desempeñan un rol determinado dentro del equipo.

El **Dueño de Producto (Product Owner)** es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Pila de Producto (Product Backlog). La gestión de la Pila de Producto incluye:

- Expresar claramente los elementos de la Pila de Producto;
- Ordenar los elementos en la Pila de Producto para alcanzar los objetivos y misiones de la mejor manera posible;
- Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo;
- Asegurar que la Pila de Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,
- Asegurar que el Equipo de Desarrollo entiende los elementos de la Pila de Producto al nivel necesario.
- Scrum **no reconoce sub-equipos** en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla.
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero **la responsabilidad recae en el Equipo de Desarrollo como un todo**.

El **Scrum Master** es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

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

Una vez que comienza un **Sprint**, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

El **Sprint** es el corazón de Scrum, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprint es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprint contienen y consisten en:

- **Reunión de Planificación de Sprint (Sprint Planning Meeting)** es el trabajo a realizar durante el Sprint y se crea mediante el trabajo colaborativo del Equipo Scrum completo.
- La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprint más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo.
- **Scrum Diario (Daily Scrum)** es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.
- **Revisión de Sprint (Sprint Review)** es para inspeccionar el Incremento y adaptar la Pila de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.
- **Retrospectiva de Sprint (Sprint Retrospective)** es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

Pila de Producto (Product Backlog) es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Pila de Producto, incluyendo su contenido, disponibilidad y ordenación.

Pila de Sprint (Sprint Backlog) es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Pila de Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un

Incremento “Terminado”.

Seguimiento del Progreso del Sprint es posible sumar el trabajo restante total en los elementos de la Pila de Sprint. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el Objetivo del Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint, el Equipo de Desarrollo puede gestionar su progreso.

Incremento es la suma de todos los elementos de la Pila de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprint anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no. **Beneficios de SCRUM**

- **Flexibilidad a cambios.** Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.
- **Reducción del Time to Market.** El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- **Mayor calidad del software.** El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad. **Mayor productividad**. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- **Maximiza el retorno de la inversión (ROI).** Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.

- **Predicciones de tiempos.** A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos.** El hecho de llevar a cabo las funcionalidades de mayor valor en primer lugar y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada. (Kniberg, 2017)

Estas características dan motivo para ponerla en práctica en el desarrollo de la aplicación que se desarrolla en este trabajo.

Software Libre

La definición de **Software Libre**, escrita por **Richard Stallman** y publicada por la **Free Software Foundation,** define al software libre como un asunto de libertad, no de precio. (Fundation, 1986)

La palabra "libre" en nuestro nombre no se refiere al precio; se refiere a la libertad. Primero, a la libertad de copiar y redistribuir un programa a tus vecinos, para que ellos al igual que tú, lo puedan usar también. Segundo, a la libertad de cambiar un programa, así podrás controlarlo en lugar que el programa te controle a ti; para esto, el código fuente tiene que estar disponible para ti. (Fundation, 1986)

De acuerdo con la definición establecida por Richard Stallman, un software es "libre" cuando garantiza estas 4 libertades:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades.
- La libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios.
- La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Aplicaciones Web

El software o aplicación de web, también conocido como un recurso o un "App", es un programa informático diseñado para ayudar al usuario a realizar singulares o múltiples tareas específicas relacionadas. Ayuda a resolver problemas en el mundo real. Los ejemplos incluyen software empresarial, chats, foros, redes sociales, tiendas virtuales, etc.

Entre las ventajas de las aplicaciones web, destacaríamos:

- **Entorno visual enriquecido con diversidad de efectos:** Haciendo uso de las tecnologías destinadas a trabajar del lado del cliente y los navegadores que ofrecen cada vez más y mejor funcionalidades, los diseñadores pueden lograr diseños atractivos, ligeros y llenos de curiosos efectos visuales.
- **Ahorra tiempo:** Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **No hay problemas de compatibilidad:** Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- **Actualizaciones inmediatas:** Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo:** Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- **Multiplataforma:** Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- **Portables:** Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- La **disponibilidad** suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Los virus no dañan los datos** porque éstos están guardados en el servidor de la aplicación.
- **Colaboración:** Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

Lenguajes de programación para la Web

Lenguajes de programación del lado del cliente

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. El navegador es una especie de aplicación capaz de interpretar las

órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.

HTML (**H**yper**T**ext **M**arkup **L**anguage)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido, etc.). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado.

HTML es el lenguaje en el que es creada la web. La utilización de HTML como uno de los lenguajes para el desarrollo, responde a las necesidades de manipulación y maquetación de los elementos visuales de la aplicación. (Mora)

Se compone de una serie de comandos, que son interpretados por el visualizador o programa que utilizamos para navegar por la web. En última instancia es el visualizador es el que ejecuta todas las ordenes contenidas en el código HTML, de forma que un visualizador puede estar capacitado para unas prestaciones, pero no para otras. Así, podemos especificar que una página tenga una imagen de fondo, o un texto parpadeando, pero si nuestro visualizador no está capacitado para esas funciones no podremos verlas. (Murcia)

Una muy breve historia de estos lenguajes para conocimiento general:
HTML 1: fue la primera versión, creada por Tim Berners-Lee en 1991.
HTML 2: la segunda versión apareció en 1994 y se terminó en 1996 con la aparición de
HTML 3.0, esta es la versión que en realidad plantea las bases de las siguientes versiones de HTML.

Las reglas y el funcionamiento de esta versión están dadas por el W3C (mientras que la primera versión fue creada por una persona).
HTML 3: apareció en 1996, esta nueva versión de HTML, añade muchas posibilidades al lenguaje como tablas, *applets, scripts,* posicionamiento de texto alrededor de imágenes, etc.
HTML 4: Esta es la versión más común de HTML (en concreto, es HTML 4.01). Apareció por primera vez en 1998 y propone el uso de marcos (que dividen una página web en varias partes), tablas más complejas, mejoras en las formas, etc. Más importante aún, esta versión permite por primera vez utilizar hojas de estilo del famoso CSS.

HTML 5.2: es la última versión. Aún no está muy extendida, llama mucho la atención porque trae muchas mejoras como la posibilidad de incluir fácilmente vídeos, mejorar el contenido, nuevas características para los formularios, etc. Esta es la versión que vamos a describir en esta documentación. El estándar está completo desde el año 2014. (Asensio, Lenguajes de programación HTML y CSS, 2019)

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red. (Cantón, 2017)

Las barreras entre sitios webs y aplicaciones finalmente han desaparecido. Las tecnologías requeridas para el proceso de integración están listas. El futuro de la web e prometedor y la evolución y combinación de estas tres tecnologías (HTML, CSS y JavaScript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo. HTML5 indica claramente el camino. (Gauchat, 2012)

CSS (**C**ascade **S**tyle **S**heet) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Es un mecanismo simple para añadir estilos (ej., color, tipos de letras, formas) a documentos web. Esto trae como ventaja la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes haciendo que su uso sea estándar para la mayoría de los navegadores. (Llonch, 2005)

Características y ventajas de las CSS

El modo de funcionamiento de las CSS consiste en definir, mediante una sintaxis especial, la forma de presentación que le aplicaremos a:

• Un web entero, de modo que se puede definir la forma de todo el web de una sola vez.

• Un documento HTML o página, se puede definir la forma, en un pequeño trozo de código en la cabecera, a toda la página.

• Una porción del documento, aplicando estilos visibles en un trozo de la página.

• Una etiqueta en concreto, llegando incluso a poder definir varios estilos diferentes para una sola etiqueta. Esto es muy importante ya que ofrece potencia en nuestra programación. Podemos definir, por ejemplo, varios tipos de párrafos: en rojo, en azul, con márgenes. (Martos, 2018)

CSS3 sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas. (Gauchat, 2012)

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Java Script podemos crear efectos especiales en las páginas y definir interactividades con el usuario.

Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con *JavaScript* se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, legalmente JavaScript es una marca registrada de la empresa Sun Microsystems. (Pérez, 2016)
JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario (eventos), con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. Además, Java Script pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (Gauchat, 2012)

● JavaScript es un lenguaje de programación interpretado, es decir, que necesita un intérprete para ser ejecutado.
● JavaScript se utiliza principalmente en páginas web.
● Al igual que HTML, JavaScript es ejecutado por el navegador del usuario: se llama un de cliente, en comparación con el lado del servidor cuando el código es ejecutado por el servidor.
● JavaScript está normalizado por ECMA International como el nombre *ECMAScript Language Reference*.
● Hay otros lenguajes derivadas del ECMAScript como ActionScript, EX4 o JScript.NET.
● La última versión del estándar está basado en ECMAScript 5, lanzado en 2009. (Asensio, Desarrollo de Aplicaciones web)

Lenguajes de programación del lado del servidor

La Programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas

HTML dinámicamente como respuesta. Por otra parte un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

Python

Es simple, pero es un lenguaje de programación real. Ofrece más apoyo e infraestructura para programas grandes que el intérprete de órdenes. Por otra parte, también ofrece mucha más comprobación de errores que C y, al ser un *lenguaje de muy alto nivel*, tiene incluidos tipos de datos de alto nivel, como matrices flexibles y diccionarios, que llevarían días de programación en C. Dados sus tipos de datos más generales, se puede aplicar a un rango de problemas más amplio que *Awk* o incluso *Perl*, pero muchas cosas son, al menos, igual de fáciles en Python que en esos lenguajes.

Python te permite dividir su programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que puedes utilizar como base de tus programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a IGU (interfaz gráfica con el usuario). Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa de la base hacia arriba. También es una calculadora muy útil.

Python permite escribir programas muy compactos y legibles. Los programas escritos en Python son típicamente mucho más cortos que sus equivalentes en C o C++, por varios motivos:

_ Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola sentencia.

_ El agrupamiento de sentencias se realiza mediante sangrado (indentación) en lugar de begin/end o llaves.

_ No es necesario declarar los argumentos ni las variables. (Rossum, Guía de aprendizaje de Python)

Es *extensible*: si ya sabes programar en C es fácil agregar una nueva función o módulo al intérprete, ya sea para realizar operaciones críticas a velocidad máxima, o para enlazar programas Python con bibliotecas que tal vez sólo estén disponibles en forma binaria (por ejemplo bibliotecas gráficas específicas de un fabricante). Una vez que estés realmente entusiasmado, podes enlazar el intérprete Python en una aplicación hecha en C y usarlo como lenguaje de extensión o de comando para esa aplicación.

Por cierto, el lenguaje recibe su nombre del programa de televisión de la BBC "Monty Python's Flying Circus" y no tiene nada que ver con reptiles. Hacer referencias a sketches de Monty Python en la documentación no sólo está permitido, ¡sino que también está bien visto! (Rossum, El Tutorial de Python, 2017)

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. (Rossum, El tutorial de Python , 2009)

Base de Datos (BD)
Sqlite3

Las bases de datos ofrecen, un método superior de entrada y salida de datos de alto volumen sobre un archivo típico como puede ser un archivo de texto. **SQLite** es una versión “ligera” que funciona basada en la sintaxis SQL. SQL es un lenguaje de programación en sí mismo, pero es un lenguaje de base de datos muy popular.

SQLite es realmente extremadamente ligero. La configuración de una base de datos **SQLite** es casi instantánea, no hay ningún servidor que configurar, ningún usuario que definir y ningún permiso del que preocuparse. Por esta razón, a menudo se utiliza como una base de datos de desarrollo y creación de prototipos. El principal problema con SQLite es que termina siendo como cualquier otro archivo plano, por lo que la entrada/salida de alto volumen, especialmente con consultas simultáneas, puede ser problemática y lenta. Cada tabla probablemente necesita su propio archivo como si estuviera haciendo archivos planos, y en SQLite está todo en uno. SQLite también va a almacenar sus datos en un búfer. Por último, en las ediciones no se requiere que se vuelva a guardar todo el archivo, tan sólo esa parte. Esto mejora significativamente el rendimiento. (SQLite)

Características de la base de datos

– **Fichero único.** La base de datos se almacena en un único fichero, cuyo formato es multiplataforma (Es posible leer el fichero en sistemas de 32 y 64 bits o en arquitecturas big-endian y little-endian). Estas características hacen que SQLite sea popular para usarlo como formato de archivo de las aplicaciones. O dicho de otra forma: Se puede usar SQLite como sustituto de Oracle o como sustituto de fopen.

– **Manifiesto de tipado.** La mayoría de motores SQL utilizan tipado estático. Cada columna de una tabla se asocia con un tipo de datos, y solo pueden introducirse valores de un tipo particular. SQLite elimina esta restricción, y hace que el tipo de datos pueda ser una propiedad del valor en sí, y no de la columna.
– **Registros de longitud variable.** Muchos otros motores SQL, fijan una cantidad de espacio para cada todas las filas, de forma que, por ejemplo, si declaramos una columna como varchar (100), el motor reservará 100 bytes de espacio para todas las filas sin tener en cuenta la información que se guarde. SQLite, por el contrario, usa sólo la cantidad de disco que necesita para almacenar la información en una fila.
– **Seguridad de los datos.** Más de dos tercios del código están dedicados puramente a la prueba y verificación. Una aplicación automatizada ejecuta cientos de miles de pruebas empleando millones de consultas SQL. SQLite responde perfectamente a fallos de reserva de memoria, y errores de E/S de disco. (Montiel)

Herramientas de desarrollo

Las fases de desarrollo de un sistema web, así como los lenguajes de programación usados, son muy extensos y variados, y por ello necesitamos herramientas específicas para cada una de ellas.

Framework y Librerías

Un **Framework** de aplicaciones web permite el desarrollo de sitios web dinámicos, servicios web y aplicaciones web. El propósito de un framework es permitir a los desarrolladores construir aplicaciones web y centrarse en los aspectos interesantes, aliviando la típica tarea repetitiva asociada con patrones comunes de desarrollo web. La mayoría de los framework de aplicaciones web proporcionan los tipos de funcionalidad básica común, tales como sistemas de plantillas, manejo de sesiones de usuario, interfaces comunes con el disco o el almacenamiento en base de datos de contenido cacheado, y persistencia de datos. Normalmente, los framework de aplicación web además promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos.

Django

En el mejor de los casos, el desarrollo Web es un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y frustrante. Django te permite enfocarte en la parte creativa – la parte divertida de tus aplicaciones Web al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. Al mismo tiempo, Django intenta no entrometerse, dejándote trabajar fuera del ámbito del framework según sea necesario. (Kaplan, 2015)

Django fue inventado para satisfacer esas nuevas ambiciones. Django te permite construir en profundidad, de forma dinámica, sitios interesantes en un tiempo extremadamente corto. Django está diseñado para hacer foco en la diversión, en las partes interesantes de tu trabajo, al mismo tiempo que alivia el dolor de las partes repetitivas. Al hacerlo, proporciona abstracciones de alto nivel de patrones comunes del desarrollo web, atajos para tareas frecuentes de programación y claras convenciones sobre cómo resolver problemas. Al mismo tiempo, Django intenta mantenerse fuera de tu camino, dejando que trabajes fuera del alcance del framework cuando es necesario. (Kaplan-Moss)

Xhtml2pdf==0.2.3

xhtml2pdf permite a los usuarios generar documentos PDF a partir de contenido HTML fácilmente y con control de flujo automatizado tales como la paginación y mantener el texto juntos. Los
Módulo de Python puede ser utilizado en cualquier entorno Python, incluyendo Django. (Xhtml2, 2018)

Bootstrap

Bootstrap es un framework que permite montar una estructura responsive fácilmente, se dice fácilmente porque muchas clases y funciones están desarrolladas. Además, incorpora muchas librerías como normalice, jQuery, less, por lo que se puede hacer uso de muchos efectos y funciones sin tener que programar de cero. También permite conseguir un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones. (Risueño, 2015)

Algunas de las razones para el uso de Bootstrap son:

- La primera ventaja de utilizar Bootstrap literalmente salta a la vista. Se puede hacer que todo se vea realmente más agradable gracias a los estilos que provee el framework. Con sólo agregar algunas clases y el markup correcto se pueden lograr casi sin esfuerzo grupos de botones, barras de navegación, dropdowns, formularios, etc. Todo sin tener que escribir una línea de CSS.
- Es **Cross Browser**. Funciona y se ve de la misma manera en la mayoría de los navegadores de escritorio.
- Es **Mobile**. Bootstrap fue pensado no sólo para funcionar en desktops sino también en dispositivos móviles como celulares o tablets. Por ello fue desarrollado teniendo en mente un diseño **responsive** para que cada componente se pueda adaptar a diferentes resoluciones de pantalla. Y por lo que pudimos probar, lo hace muy bien.

- **Ahorra tiempo**. Al tener resuelto todo lo que mencionamos anteriormente, podemos enfocarnos en otros aspectos de nuestra aplicación. (Cochran, 2012)

Chart.js

Es el nuevo chico en el bloque para los gráficos de JavaScript. Es una excelente biblioteca para hacer gráficos, especialmente si su audiencia principal está usando navegadores más modernos. Hasta ahora este ha sido de los mejore, tanto de pago como gratuito. La mejor parte es que chart.js es gratis. (Chart.js)

Highcharts

Es una biblioteca de gráficos basada únicamente en JavaScript destinada a mejorar las aplicaciones web mediante añadiendo capacidad de gráficos interactivos. Es compatible con una amplia gama de gráficos. Los gráficos se dibujan utilizando SVG en navegadores estándar como Chrome, Firefox, Safari, Internet Explorer.

- Características
- Las siguientes son las características más destacadas de la biblioteca de Highcharts.
- Compatibilidad: funciona de forma impecable en todos los principales navegadores y plataformas móviles, como Android y iOS.
- Soporte multitáctil: admite plataformas multitáctiles en pantalla táctil como Android y iOS. Ideal para teléfonos inteligentes / tabletas basados en iPhone / iPad y Android.
- De uso gratuito: código abierto y de uso gratuito para fines no comerciales.
- Ligero: la biblioteca principal de highcharts.js con un tamaño de casi 35 KB, es extremadamente liviana biblioteca.
- Configuraciones simples: utiliza json para definir diversas configuraciones de los gráficos y muy Fácil de aprender y usar.
- Dinámico: permite modificar el gráfico incluso después de la generación del gráfico.
- Ejes múltiples - No restringido a x, eje y. Soporta múltiples ejes en los gráficos.
- Información sobre herramientas configurable: la información sobre herramientas se presenta cuando un usuario se desplaza sobre cualquier punto de un gráfico.
- Highcharts proporciona información sobre herramientas incorporada en el formateador o en el formato de devolución de llamada para controlar la información sobre herramientas programáticamente
- Soporte de Date Time - Maneje la fecha especialmente. Proporciona numerosos controles incorporados sobre fecha sabia categorías.

- Exportar - Exportar gráfico a formato PDF / PNG / JPG / SVG habilitando la función de exportación.
- Imprimir - Imprimir gráfico utilizando la página web.
- Zoomablity: admite gráficos de zoom para ver los datos con mayor precisión.
- Datos externos: admite la carga dinámica de datos desde el servidor. Proporciona control sobre los datos utilizando las funciones de devolución de llamada. (GUIDE, 2018)

JQuery

Es una **librería** que se encuentra escrita en JavaScript, un lenguaje de programación muy rico y expresivo. Esta es la librería web más popular disponible en estos días. La librería jQuery es gratuita y fue diseñada para simplificar la creación de sitios web modernos. Facilita la selección de elementos HTML, la creación de animaciones y efectos, y también controla eventos y ayuda a implementar Ajax en nuestras aplicaciones.

La librería jQuery se encuentra en un archivo pequeño que se puede descargar desde **www.jquery.com** y luego incluir en nuestros documentos usando la etiqueta **<script>** y provee una API sencilla que cualquiera puede aprender y rápidamente aplicar a sus proyectos.

Una vez que el archivo provisto por jQuery es incluido en nuestro documento, ya estamos listos para aprovechar los métodos simples incorporados por la librería y convertir nuestra web estática en una moderna y práctica aplicación. Tiene la ventaja de proveer soporte para viejos navegadores y vuelve simple tareas cotidianas. Puede ser utilizado junto con HTML5, CSS3 o como una forma simple de reemplazar funciones de HTML5 en navegadores que no están preparado para esta tecnología. (JQuery, 2015)

Entorno de Desarrollo Integrado (IDE)
Visual Studio Code

Ofrece a los desarrolladores una nueva opción de herramienta de desarrollador que combina la simplicidad y la experiencia optimizada de un editor de código con lo mejor de lo que los desarrolladores necesitan para su ciclo de depuración de código de núcleo. Visual Studio Code es el primer editor de código y la primera herramienta de desarrollo multiplataforma, compatible con OS X, Linux y Windows, en la familia Visual Studio.

En su esencia, Visual Studio Code (VS Code) cuenta con un potente y rápido editor de códigos ideal para el uso diario. La versión Beta de Code ya tiene muchas de las funciones que los desarrolladores necesitan en un editor de código y texto, incluida la navegación, el soporte de teclado con enlaces personalizables, el resaltado de sintaxis, la coincidencia de corchetes, la sangría automática y los fragmentos de código, con soporte para docenas de idiomas.

Para la codificación seria, los desarrolladores a menudo necesitan trabajar con el código como algo más que texto. El código de Visual Studio incluye soporte incorporado para la finalización del código de IntelliSense siempre activo, la mejor comprensión y navegación del código semántico y la refactorización del código. El código incluye excelentes herramientas para tecnologías web como HTML, CSS y JSON. El código también se integra con los administradores de paquetes y repositorios, y crea y otras tareas comunes para hacer que los flujos de trabajo diarios sean más rápidos.

Pero los desarrolladores no pasan todo su tiempo simplemente escribiendo código: van y vienen entre la codificación y la depuración. La depuración es la característica más popular en Visual Studio y, a menudo, la característica de un IDE que los desarrolladores desean en una experiencia de codificación más ágil. Visual Studio Code incluye una experiencia de depuración integrada y optimizada, con soporte para la depuración de Node.js y más para el futuro.

Visual Studio Code incluye un modelo de extensibilidad pública que permite a los desarrolladores crear y usar complementos, y personalizar de forma rica su experiencia de edición-creación-depuración. (Tobias Kahlert, 2016)

Herramientas de modelado

Visual Paradigm for UML 8 es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas, generar documentación, así como la generación de código inverso.

Posee varias características importantes como son (multiplataforma, interoperabilidad, modelamiento de los requisitos, colaboración de equipo, generación de documentos, editor de detalles de casos de uso, ingeniería de código, modelado de procesos del negocio y modelamiento de bases de datos. (UML, 2019)

Conclusiones parciales del capítulo.

Durante este capítulo se consolidó el basamento teórico con vistas al desarrollo del software mediante un estudio del estado del arte de las principales herramientas y tecnologías que se proponen para el proyecto. Se profundiza en el análisis de las herramientas existentes hasta el momento, para concluir con los resultados que se esperan alcanzar al final del trabajo. Se define el modelo de desarrollo a seguir, el cual describe cada artefacto a obtener en cada etapa del desarrollo de esta aplicación y se sientan las bases que dan pie a su inicio.

Queda plasmada de manera clara, la necesidad de desarrollar un software para CONFORMAT que garantice el acceso a la información actualizada de los procesos productivos, para contribuir al aumento de la rapidez del trabajo y de obtención de datos.

Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología SCRUM y se realiza un análisis detallado de las tecnologías seleccionadas para el desarrollo de la propuesta de solución.

CAPÍTULO II: Análisis, diseño y construcción de la solución propuesta

En el presente capítulo se hace una descripción de la solución propuesta. Según la metodología seleccionada se explican los roles, eventos y artefactos para la creación de la Pila del Producto y la Pila de Sprint. La implementación de la solución se basa en los principios y reglas de la metodología SCRUM utilizando las tecnologías y herramientas definidas. Además se generan otros artefactos que no son parte de la metodología usada pero se emplean para brindar una mayor comprensión del sistema a implementar, como es el caso del diagrama entidad-relación. Y el análisis de los resultados obtenidos teniendo en cuenta el análisis económico de la solución.

Modelación del negocio

Diagrama de Casos de Uso del Negocio

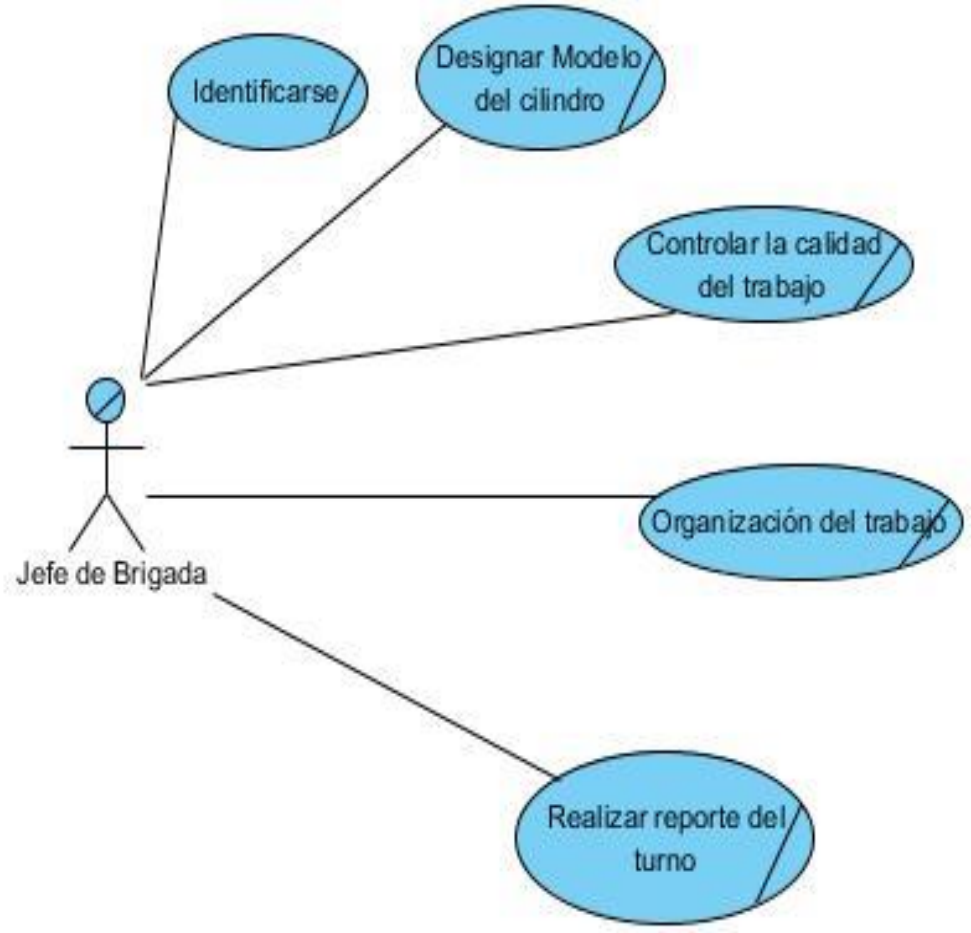


Figura 2.1 Diagrama de casos de usos del negocio

Descripción de Actores del Negocio

Actor del negocio	Descripción
Jefe de Brigada	El Jefe de Brigada es aquella persona que inicializa el negocio (Solicitar los datos de producción), pidiéndole al Operario los datos.

Descripción de Trabajadores del Negocio

Trabajador del negocio	Descripción
Operario	El (la) Operario es aquel trabajador del negocio que se encarga de realizar las operaciones en el proceso de producción.

Procesos del Negocio

Proceso: "Designar modelo del cilindro"

Descripción del Caso de Uso

Caso de uso del negocio	*<Designar modelo del cilindro>*
Actores	*Jefe de Brigada*
Resumen	*<El Caso de Uso comienza cuando el Jefe de Brigada revisa la existencia de materia prima y observa cual es el cilindro a trabajar en el día. Luego designa el modelo del cilindro al Operario. >*
Casos de uso asociados	

Acción del actor	Respuesta del proceso de negocio
1. *Revisa la existencia de materia prima.* 2. *Observa el modelo de cilindro a trabajar.* 3. *Orienta al operario.*	4. *El operario comienza jornada laboral.*
Otras secciones	
Cursos Alternos:	
3.1: En caso de no haber materia prima no se produce.	

Diagrama de Actividad del caso de uso Controlar la calidad del trabajo (ver Anexo 2)

Proceso: **"Controlar la calidad del trabajo"**

Descripción del Caso de Uso

Caso de uso del negocio	< Controlar la calidad del trabajo>
Actores	*Jefe de Brigada*
Resumen	*<El Caso de Uso comienza cuando el Jefe de Brigada se desplaza por el taller y revisa los diferentes procesos de producción. Toma nota de los procesos defectuosos. Al momento de encontrar un cilindro defectuoso informa al operario y se rechaza. >*
Casos de uso asociados	

Acción del actor	Respuesta del proceso de negocio
1. El Jefe de Brigada se desplaza por el taller. *2. Revisa los diferentes procesos de producción.* *3. Toma nota de los procesos defectuosos* *4. Informa al operario*	*5. Rechaza el cilindro*
Otras secciones	
Cursos Alternos:	
3.1 El no defectuoso pasa al siguiente proceso	

Diagrama de Actividad del caso de uso Controlar la calidad del trabajo (ver Anexo 3)

Solución propuesta

Haciendo uso de tecnologías y herramientas se propone el desarrollo de una aplicación Web para permitir el acceso a la información actualizada de los trabajadores de CONFORMAT. La herramienta cumplirá con una serie de requisitos funcionales y no funcionales que lograrán que la interacción del administrador, el jefe de brigada y el supervisor sea sencilla y eficiente, garantizándose la actualización de la información en el proceso productivo.

Desarrollo de la solución propuesta

En el proceso de desarrollo de la solución propuesta se seguirá el formato de trabajo propuesto por la metodología SCRUM que fue seleccionada para guiar el proceso de desarrollo. A continuación se especifican los roles y se crean los artefactos de la metodología con el fin de dar seguimiento al proceso de desarrollo y lograr un producto de calidad.

Roles

A continuación se relacionan los roles y las funciones del equipo Scrum en el desarrollo de la solución:

Tabla 1: Definición de los roles del equipo de Scrum

Rol	Miembro
Dueño del producto	CONFORMAT

Equipo de desarrollo	Carlos Adrian Alonso Infante
Scrum master	Darien Menéndez

Artefactos

Los artefactos de SCRUM representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por SCRUM están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

Pila de Producto (PRODUCT BACKLOG)

Debido a que la **Pila de Producto** es la única fuente de requisitos y funcionabilidades para la realización del producto; que está en constante evolución y es un artefacto vivo y dinámico; ha pasado por un proceso de cambios constantes para identificar lo que el producto necesita para ser adecuado, competitivo y útil. A continuación se presenta la Pila de Producto final, elaborada y ordenada por el Dueño de Producto atendiendo a las necesidades del producto.

Tabla 2: Pila de Producto (PRODUCT BACKLOG)

	PILA DE PRODUCTO	
PRIORIDAD	**DESCRIPCIÓN**	**ESTIMADO (hrs)**
1	Investigar la plataforma tecnológica	24
2	Analizar y diseñar de la Base de Datos	120
3	Diseñar la Interface de Usuario	56
4	Crear la vista de Datos de Producción	48
5	Crear la vista de progreso de producción del trabajador	48
6	Crear la vista de Listado de operaciones	48
7	Crear la vista del Búsqueda de Históricos	40
8	Crear las vista de información estática	30

9	Crear el mecanismo de seguridad	24
10	Gestionar organización del trabajo	40
11	Gestionar reporte final	48
12	Gestionar operarios	30
13	Gestionar operaciones	30
14	Gestionar producto nuevo	30
15	Gestionar incidencias	48
16	Gestionar usuarios	40

Pila de Sprint (SPRINT BACKLOG)

Partiendo de la Pila de Producto anteriormente elaborada se crea la Pila de Sprint que descompone las funcionalidades de la Pila de Producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. Además también se planificarán de forma independiente y detallada cada sprint, descomponiendo el trabajo en unidades de tamaño adecuado para monitorear el avance a diario, e identificar riesgos y problemas sin necesidad de procesos de gestión complejos.

Tabla 3: Pila de Sprint (SPRINT BACKLOG)

	PILA DE SPRINT				
Sprint	**Pila de Producto**	**Encargado**	**Fecha Inicial**	**Fecha Final**	**Valor**
1	Investigar la plataforma tecnológica	Carlos Alonso	18/11/18	20/12/18 (30 días)	200
	Analizar y diseñar de la Base de Datos				
	Diseñar de la Interface de Usuario				

2	Crear la vista de Datos de Soldadura	Carlos Alonso	5/01/19	5/03/19 (50 días)	549
	Crear la vista de progreso de producción				
	Crear la vista de Listado de operaciones				
	Crear la vista del Búsqueda de Históricos				
3	Crear las vista de información estática	Carlos Alonso	11/03/15	11/04/15 (30 días)	691
	Crear el mecanismo de seguridad				
	Gestionar organización del trabajo				
	Gestionar reporte final				
4	Gestionar operarios	Carlos Alonso	13/04/15	1/06/15 (45 días)	869
	Gestionar operaciones				
	Gestionar producto nuevo				
	Gestionar incidencias				
	Gestionar usuarios				

Planificación de Sprint del proyecto

Tabla4: Planificación del SPRINT 1

	SPRINT 1			
Pila del Producto	**Tareas**	**Encargado**	**Est. Inicial**	**Est. Total**

Investigar la plataforma tecnológica	- Definir las características mínimas del hardware - Seleccionar el gestor de base de datos - Seleccionar los lenguajes de programación y herramientas de desarrollo	Carlos Alonso	18	**200**
	- Realizar pruebas rendimiento al hardware instalado - Realizar pruebas de rendimiento al gestor de base de datos - Comprobar que los lenguajes de programación y las herramientas seleccionadas cumplen las exigencias del proyecto	Carlos Alonso Darien Menéndez	6	
Analizar y diseñar de la Base de Datos	- Analizar y definir los tipos de datos - Normalizar la base de datos - Crear las clases de la base de datos - Generar la base de datos	Carlos Alonso	104	
	- Cargar datos de prueba	Carlos Alonso Darien Menéndez	8	
Diseñar de la Interface de Usuario	- Seleccionar colores del diseño - Diseñar la interface de usuario - Realizar la maquetación del diseño	Carlos Alonso	48	

	- Comprobar la funcionalidad y adaptabilidad del diseño en los diferentes dispositivos y navegadores	Carlos Alonso Darien Menéndez	8	

Tabla 5: Planificación del SPRINT 2

	SPRINT 2			
Pila del Producto	**Tareas**	**Encargado**	**Est. Inicial**	**Est. Total**
Crear la vista de Datos de Producción	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista	Carlos Alonso	40	184
	- Comprobar la correcta funcionabilidad de las vistas - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso	8	
Crear la vista de progreso de producción del trabajador	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista	Carlos Alonso	40	
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso	8	

Crear la vista de Listado de operaciones	- Crear y maquetar la vista general -Recuperar la información de la base de datos y mostrarla en la vista correspondiente	Carlos Alonso	40	
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso Darien Menéndez	8	
Crear la vista del Búsqueda de Históricos	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista	Carlos Alonso	35	
	- Comprobar la correcta funcionabilidad de la vista	Carlos Alonso Darien Menéndez	5	

Tabla 6: Planificación del SPRINT 3

	SPRINT 3			
Pila del Producto	**Tareas**	**Encargado**	**Est. Inicial**	**Est. Total**
Crear las vista de información estática	- Crear y maquetar las vistas estáticas - Agregar la información estática a las vistas	Carlos Alonso	25	
	- Comprobar la correcta funcionabilidad de las vistas - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso	5	
Crear mecanismo de seguridad	- Crear y maquetar la vista - Generar el formulario de inicio de sesión - Crear la lógica de validación de usuario	Carlos Alonso	16	
	- Comprobar la correcta funcionabilidad de la vista - Hacer pruebas de inicio de sesión	Carlos Alonso	8	
Gestionar Organización del trabajo	- Crear y maquetar la vista de agregar, editar y eliminar Organización del trabajo - Crear Organización del trabajo - Editar Organización del trabajo - Eliminar Organización del trabajo	Carlos Alonso	35	
	- Comprobar la correcta funcionabilidad de la vista	Carlos Alonso	5	142

	- Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario	Darien Menéndez		
Gestionar Reporte Final	- Crear y maquetar las vistas de agregar, editar y eliminar reporte final - Agregar reporte final - Editar reporte final - Eliminar reporte final - Generar los formularios de las correspondientes vistas	Carlos Alonso	40	
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso Darien Menéndez	8	

Tabla 7: Planificación del SPRINT 4

	SPRINT 4			

Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Gestionar operarios	- Crear y maquetar las vistas de agregar, editar y eliminar operarios - Agregar operarios - Editar operarios - Eliminar operarios - Generar los formularios de las correspondientes vistas	Carlos Alonso	25	**138**
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario	Carlos Alonso Darien Menéndez	5	
Gestionar operaciones	- Crear y maquetar las vistas de agregar, editar y eliminar operaciones - Agregar operaciones - Editar operaciones - Eliminar operaciones - Generar los formularios de las correspondientes vistas	Carlos Alonso	25	

	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario	Carlos Alonso Darien Menéndez	5	
Gestionar producto nuevo	- Crear y maquetar las vistas de agregar, editar y eliminar producto nuevo - Agregar producto nuevo - Editar producto nuevo - Eliminar producto nuevo - Generar los formularios de las correspondientes vistas	Carlos Alonso	20	
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario	Carlos Alonso Darien Menéndez	10	
Gestionar incidencias	- Crear y maquetar la vista de incidencias y la de lectura de incidencias - Agregar incidencias - Mostrar incidencias	Carlos Alonso	40	

	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso Darien Menéndez	8	
Gestionar Usuarios	- Agregar usuarios - Mostrar usuarios - Eliminar usuarios	Carlos Alonso	35	
	- Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Carlos Alonso Darien Menéndez	5	

Modelo de datos

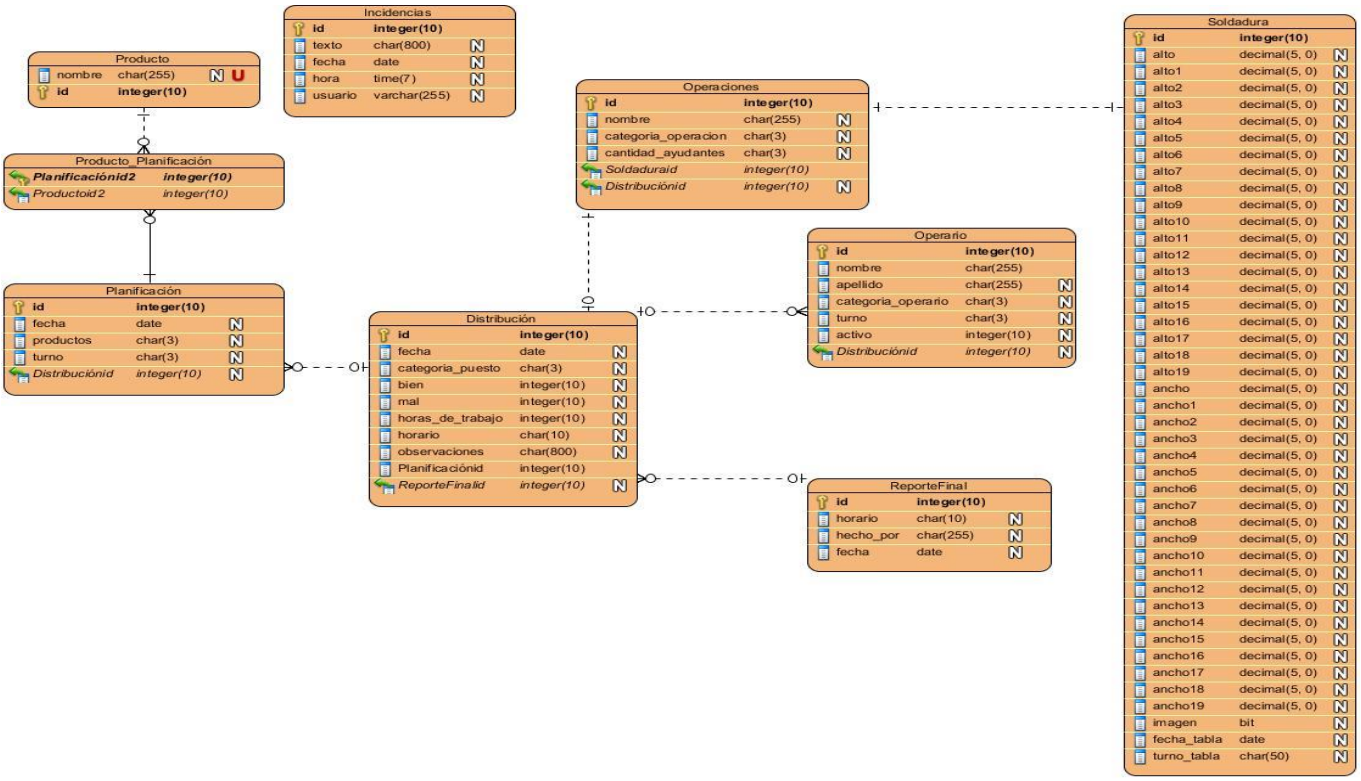


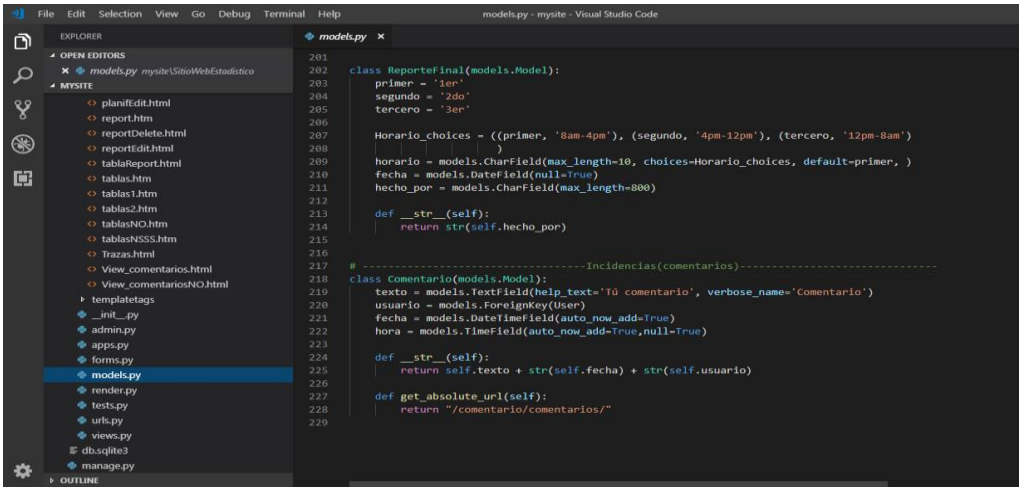
Figura 2.2 Modelo de datos

Patrón Arquitectónico

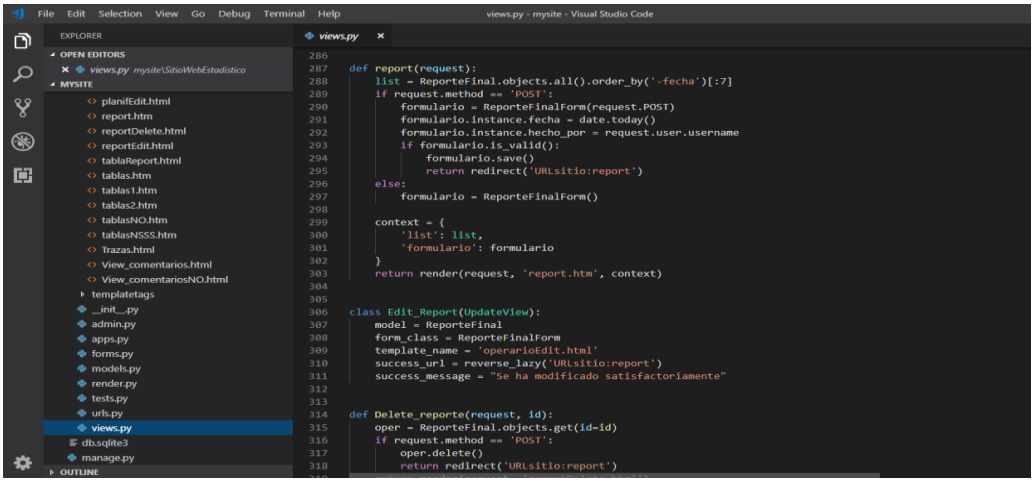
Ofrece soluciones a problemas de arquitectura de software. Expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. Posee un nivel mayor de abstracción.

Durante el proceso de desarrollo se emplea el MVT (Modelo-Vista-Plantilla), es un patrón arquitectónico de software característico de Django. Es una colección de tres componentes importantes Modelo, Vista y Plantilla. El modelo ayuda a manejar la base de datos. La plantilla es una capa de presentación que maneja la parte de la interfaz de usuario completamente. La vista se utiliza para ejecutar la lógica de negocios e interactuar con un modelo para transportar datos y renderizar una plantilla. Un ejemplo claro del uso de este patrón se encuentra en las siguientes imágenes del software:

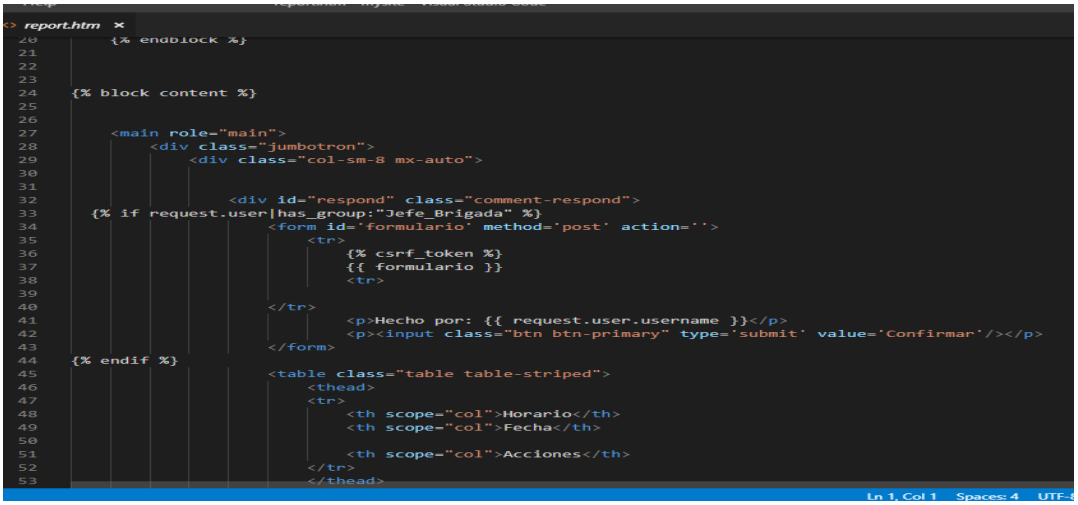
Modelo de la base de datos



Vistas



Template o plantilla



Patrones de diseño

Durante el proceso de elaboración del software, se tuvo en cuenta los patrones de diseño para hacer uso de buenas prácticas de programación. Ejemplo de esto se puede ver a continuación con los patrones empleados.

Decorador

El patrón Decorador responde a la necesidad de añadir dinámicamente funcionalidad a un Objeto. Esto nos permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Un ejemplo claro es el Login required, que es el encargado de cuando se hace una solicitud a la URL se redirija a la plantilla Login con el objetivo de autenticarse el usuario, permitiendo un control en la seguridad del software.

Ejemplo del uso del patrón en el software

urls.py

```
1   from django.conf.urls import url
2   from django.contrib.auth.decorators import login_required
3
4   from SitioWebEstadistico import views
5   from SitioWebEstadistico.views import ComentarioCreate
6
7   urlpatterns = [
8
9       url(r'^$', login_required(views.home), name='home'),
10      url(r'^index/', login_required(views.index), name='index'),
11
12      url(r'^graficas/', login_required(views.graficas), name='graficas'),
13      url(r'^tablas/', login_required(views.tablas), name='tablas'),
```

Experto en información

El patrón de experto en información es el principio básico de asignación de responsabilidades. Nos indica por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). Cada clase es encargada de una función en específico como crear, editar y eliminar.

Ejemplo del uso del patrón en el software.

```
323     def cilindro(request):
324         list = Producto.objects.all()
325         if request.method == 'POST':
326             formulario = ProductoForm(request.POST)
327             if formulario.is_valid():
328                 list = Producto.objects.all()
329                 formulario.save()
330 
331                 return redirect('URLsitio:cilindro')
332         else:
333             formulario = ProductoForm()
334             list = Producto.objects.all()
335         context = {
336             'list': list,
337             'formulario': formulario
338         }
339         return render(request, 'cilindro.htm', context)
340 
341 
342     class cilindroEdit(UpdateView):
343         model = Producto
344         form_class = ProductoForm
345         template_name = 'cilindroEdit.html'
346         success_url = reverse_lazy('URLsitio:cilindro')
347 
348 
349     def Delete_cilindro(request, id):
350         oper = Producto.objects.get(id=id)
351         if request.method == 'POST':
352             oper.delete()
353             return redirect('URLsitio:cilindro')
354         return render(request, 'cilindroDelete.html')
355 
```

Alta cohesión

Nos dice que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase.

Bajo acoplamiento

Es la idea detener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

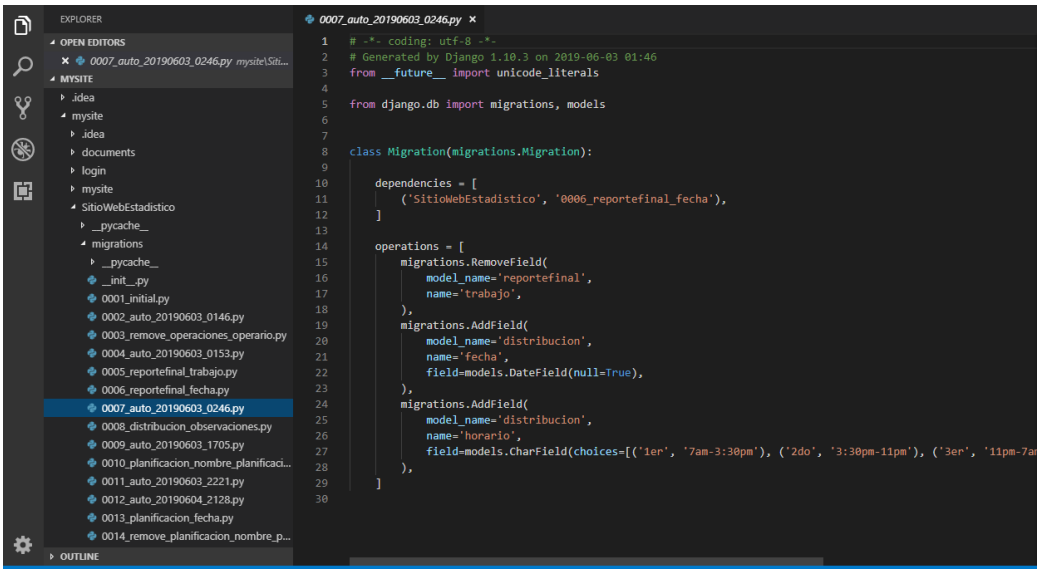
Ejemplo de alta cohesión y bajo acoplamiento en el software

```
def diezkgnuevo(request):
    list = Operaciones.objects.all().order_by('-id')[:7]
    if request.method == 'POST':
        formulario = OperacionesForm(request.POST)
        if formulario.is_valid():
            list = Operaciones.objects.all().order_by('-id')[:7]
            formulario.save()
            return redirect('URLsitio:diezkgnuevo')
    else:
        formulario = OperacionesForm()
        list = Operaciones.objects.all().order_by('-id')[:7]
    context = {
        'list': list,
        'formulario': formulario
    }
    return render(request, 'diezkgnuevo.htm', context)


class operacionesEdit(UpdateView):
    model = Operaciones
    form_class = OperacionesForm
    template_name = 'operacionesEdit.html'
    success_url = reverse_lazy('URLsitio:diezkgnuevo')


def Delete_operacion(request, id):
    oper = Operaciones.objects.get(id=id)
    if request.method == 'POST':
        oper.delete()
        return redirect('URLsitio:diezkgnuevo')
    return render(request, 'operacionDelete.html')
```

ORM

ORM procede de las siglas ORM, Object Relational Mapping. El trabajo deja de ser manual ya que el ORM lo realizará de forma independiente de la base de datos. Además, gracias al mapeo automático podrás cambiar de motor de base de datos fácilmente y cuando quieras. El ORM es un modelo de programación que transforma las tablas de una base de datos en entidades para simplificar enormemente la tarea del programador. El uso de ORM se puede apreciar mediantes las Migraciones realizadas de forma automática sin necesidad de tocar la base de datos. Es una de las facilidades que brinda Django, haciendo posible hacer cambios en el modelo de la base de datos, de manera rápida y segura. Un ejemplo de ello se muestra en la imagen, de las migraciones realizadas a lo largo de la implementación del software.

Migraciones realizadas al software.

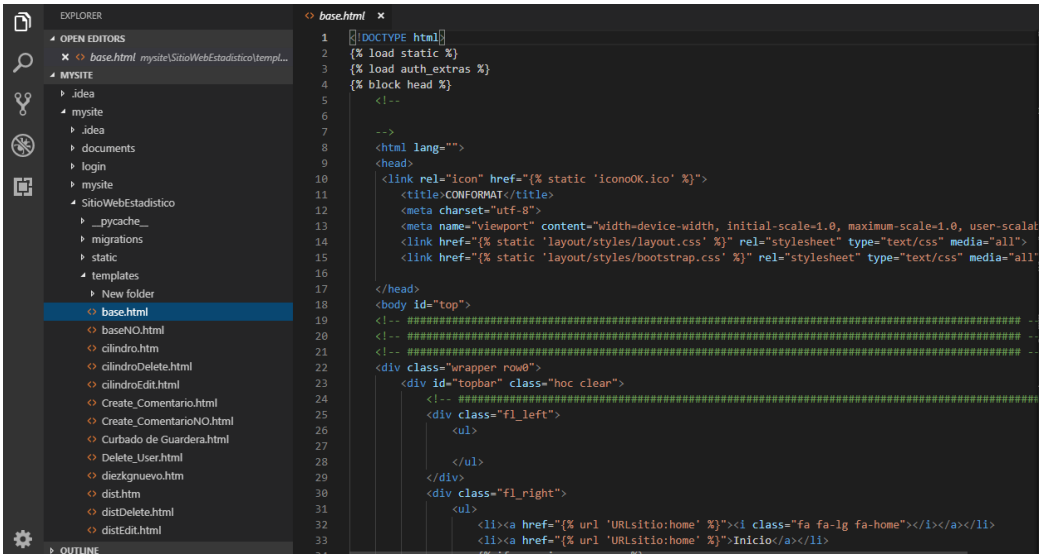


DRY

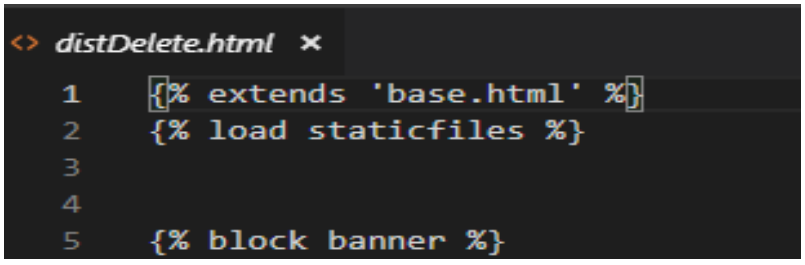
En el software se aplica el principio No te repitas (en inglés Don't Repeat Yourself o DRY, también conocido como Una vez y sólo una) es una filosofía de definición de procesos que promueve la reducción de la duplicación especialmente en la informática. Según este principio toda "pieza de información" nunca debería ser duplicada debido a que la duplicación incrementa la dificultad en los cambios y evolución posterior, puede perjudicar la claridad y crear un espacio para posibles inconsistencias.

Cuando el principio DRY se aplica de forma eficiente los cambios en cualquier parte del proceso requieren cambios en un único lugar. Por el contrario, si algunas partes del proceso están repetidas por varios sitios, los cambios pueden provocar fallos con mayor facilidad si todos los sitios en los que aparece no se encuentran sincronizados.

Base.html



Ejemplo del principio DRY en el software



Estimación de costo del proyecto

Principales Conceptos en el Método por Puntos de Función

• EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier interno.

• EO: Procesos en los que se envía datos al exterior de la aplicación.

• EQ: Procesos consistentes en la combinación de una entrada y una salida en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

• ILF: Grupos de datos relacionados entre sí, internos al sistema.

• EIF: Grupos de datos que se mantienen externamente.

• PFTe: Total Puntos de Función para las entradas del sistema.

• PFTo: Total Puntos de Función para las salidas del sistema.

• PFTq: Total Puntos de Función para las consultas del sistema.

• PFTif: Total Puntos de Función para los archivos internos del sistema.

• PFTef: Total Puntos de Función para los archivos externos del sistema

Tablas de Valores

EI- EO

CLASIFICACION DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

Tabla 2

EO

CLASIFICACION DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

Tabla 1

ILF

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

Tabla 3

EIF

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

Tabla 4

La información que resulte de estas tablas se pondera en una tabla general la cual describe el componente y el nivel en el que se encuentra. Determinando el peso de cada componente

Componente	Bajo	Medio	Alto	Total
EI	6 * 3 = _18	2 * 4 = _8	3* 6 = 18	PFTe=44
EO	4 * 4 = 16	Om * 5 = _	Oa * 7 = _	PFTo=16
EQ	4 * 3 = 12	* 4 = _	Qa * 6 = _	PFTq=12
ILF	4 * 7 = 28	IFm * 10 = _	IFa * 15 = _	PFTif=28
EIF	1 * 5 = _	EFm * 7 = _	EFa * 10 = _	PFTef=5
				PFSA =105

PFSA = PFTe + PFTo + PFTq + PFTif + PFTef

PFSA =44+16+12+28+5

PFSA =105

Luego de obtener los puntos de función sin ajustar, debemos calificar cada uno de los factores de valor de ajuste, utilizando una escala del 0 al 5 con el siguiente desglose:

0 → sin influencia

1 →influencia incidental

2 →influencia moderada

3 →influencia media

4 →influencia significativa

5 →fuertes influencian en toda la aplicación

Calificamos cada uno de los 14 Ítem y sumamos los grados de influencia (TDI) para obtener el factor de complejidad técnica (FCT):

Ítem	TDI
1. Comunicación de datos	5
2. Proceso distribuido de datos	2
3. Desempeño	5
4. Configuración	3
5. Volumen de transacciones	4
6. Captura de datos en Línea	4
7. Eficiencia al usuario final	5
8. Actualización de Datos en Línea	5
9. Complejidad	3
10. Reusabilidad	4
11. Facilidad de instalación	4
12. Facilidad de operación	4
13. Instalación Múltiple	3
14. Facilidad de cambio	4
FCT	55

Puntos de Función Ajustados (PFA)

PFA= PFSA*[0,65+ (0,01* FCT)]

PFA=105*[0.65+ (0.01*55)]

PFA=105*[0.65+0.55]

PFA=105*1.2

PFA=126

Líneas de código (LC)

LC= PFA *(Líneas * PF)

LC=126*100

LC= 12600

Esfuerzo hora/persona

E=PFA/ (1/8 persona/hora)

E=PFA/ (1/5) 1 persona trabaja 5h

E=126/0.2

E=630 horas/persona

Tomando 24 días laborables en el mes y 8 horas productivas al día, obtenemos 192 horas laborables al mes.

Duración del proyecto en meses

630horas/persona/1 persona =630 horas

DM=630 horas /192 horas/mes

DM= 3.28 aproximadamente 3.5 meses + 1mes y 1 semana por el margen de error de la estimación por puntos de función y 1 mes para la fase de prueba

DM =aproximadamente 5.6 meses

Costo total del proyecto

CT= sueldo de 1 persona/ cant de personas*DM

CT= (500/1) *5.6

CT=2800

Conclusiones del capítulo

En este capítulo se plantean los roles, eventos y artefactos necesarios para desarrollar el proyecto según la metodología SCRUM. Se escribe la pila de producto, la pila de Sprint y se planifican los Sprint con sus respectivas tareas. Se pudo obtener al final de la terminación de los cuatro Sprint un sistema informático completamente en funcionamiento. Se pudo determinar la estimación de los costos del sistema informático.

CAPÍTULO III: Validación de la solución propuesta

En el presente y último capítulo se explicarán los casos de prueba, las estrategias de prueba, las pruebas de aceptación con el cliente por medio de pruebas funcionales.

Pruebas de Aceptación

Con la aplicación de las **pruebas de aceptación** se pretende comprobar que el software cumple las expectativas que el cliente espera. Para llevar a cabo la validación del sistema se decidió aplicar pruebas de aceptación a los elementos de la Pila de Producto.

Las **pruebas de aceptación** son destinadas a evaluar si al terminar un Sprint se consiguió la funcionalidad requerida por el cliente. Estas pruebas aseguran el comportamiento del sistema y especifican los aspectos a probar cuando un Sprint ha sido correctamente implementado.

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- **Código**: servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- **EPP**: tendrá el nombre del elemento de la pila de producto al que hace referencia la prueba a realizar.
- **Nombre**: nombre que se le da a la prueba a realizar.
- **Descripción**: se describe la funcionalidad que se desea probar.
- **Condiciones de Ejecución**: mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- **Entradas / Pasos de Ejecución**: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- **Resultado esperado**: se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- **Evaluación de la prueba**: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
- **Satisfactoria**: cuando el resultado de la prueba es exactamente el esperado por el usuario.

- **Parcialmente satisfactoria**: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
- **No satisfactoria**: cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida el EPP.

Se ha tomado una muestra al azar de las pruebas de aceptación, el resto se encuentran en el documento del proyecto.

Tabla 8: Prueba de Aceptación 1

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 1	**No EPP**: 2
Nombre Caso de Prueba: Test de integridad a la base de datos	
Descripción: Verificar que la inserción, edición, eliminación de datos y las relaciones entre los mismo sean correctas	
Condiciones de ejecución: Servidor ejecutándose y Django como administrador de base de datos	
Entradas: Datos de pruebas	
Resultado esperado: Todos los conjuntos de datos probados han sido correctamente procesados	

Evaluación: Prueba satisfactoria

Tabla 9: Prueba de Aceptación 2

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 2	**No EPP**: 3
Nombre Caso de Prueba: Test de adaptabilidad a la Interface de Usuario (UI)	
Descripción: Verificar que la UI se adapte de forma correcta y óptima según la concepción del diseño para las diferentes resoluciones de los dispositivos	
Condiciones de ejecución: Tener acceso a la red con cualquier dispositivo que permita la navegación Web.	
Entradas: Interfaces de la aplicación	
Resultado esperado: Se muestren las interfaces visuales de la aplicación con las características que se diseñaron según la resolución del dispositivo en el cual se está ejecutando	
Evaluación: Prueba satisfactoria	

Tabla 10: Prueba de Aceptación 3

PRUEBAS DE ACEPTACIÓN

Número Caso de Prueba: 3	**No EPP**: 9
Nombre Caso de Prueba: Test de autenticación de usuario	
Descripción: Autenticar un usuario	
Condiciones de ejecución: El usuario tiene que estar agregado en la base de datos y activo	
Entradas: **Nombre de usuario y contraseña**	
Resultado esperado: El usuario es autenticado y según el rol que desempeña la aplicación debe mostrar la vista de inicio (administrador, todas las opciones ; jefe de brigada y supervisor)	
Evaluación: Prueba satisfactoria	

Tabla 11: Prueba de Aceptación 4

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 4	**No EPP**: 14
Nombre Caso de Prueba: Test a agregar producto nuevo	

Descripción: Verificar que el producto se agregue y que no se agreguen productos repetidos.
Condiciones de ejecución: El usuario está autenticado, hace click sobre el menú de producto nuevo y agrega un producto nuevo
Entradas: Introducir el nombre del producto
Resultado esperado: El sistema retorna un error, informando al usuario que ya existe ese producto
Evaluación: Prueba satisfactoria

Pruebas de sistemas web

Esta prueba se realizó con el fin de medir requerimientos inherentes a las aplicaciones web y sus características, como son el rendimiento, tiempo de carga, entre otras. Para esta prueba se utilizó la herramienta Lighthouse, una herramienta de medición desarrollada por Google específicamente para medir aplicaciones web de acuerdo con métricas estandarizadas y otras determinadas por ellos a partir de probar de sus propios productos.

La figura 3.1 muestra las métricas utilizadas por la herramienta Lighthouse para medir la calidad de una aplicación web, estas son el rendimiento, que se trata de los tiempos que toma la aplicación antes de que se renderice algún contenido o está lista para que el usuario interactúe con ella. Por último, el Progressive Web App o aplicación web progresiva son sitios web que utilizan la tecnología de *Service Workers* para indexar y almacenar en el dispositivo del usuario todos los archivos necesarios para el

funcionamiento del sito completamente desde el dispositivo, lo que permite que solo se necesiten hacer las peticiones necesarias para obtener los datos e incluso que la aplicación funcione sin estar conectada a internet. La métrica mide el nivel de implementación de dichas técnicas. (Danenner)

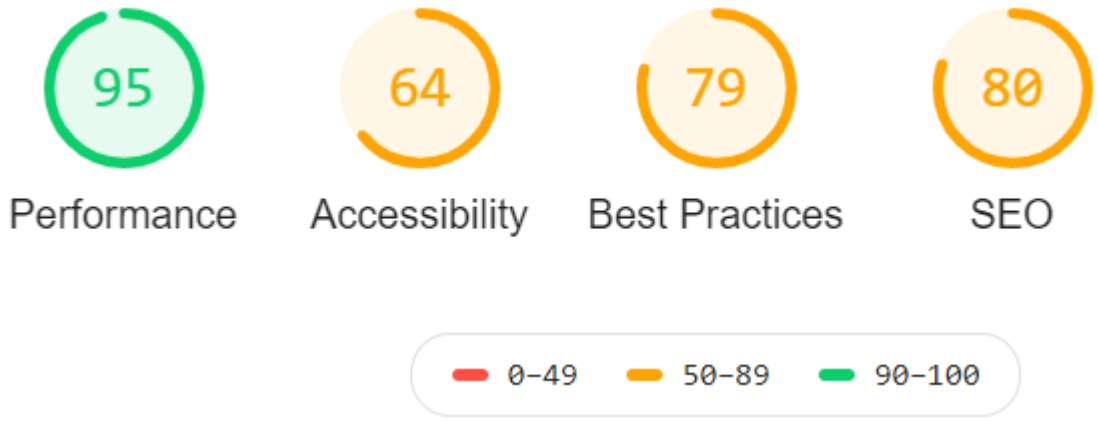


Figura 3.1 Resultados Generales de las mediciones realizadas por la herramienta Ligthouse Report.

En la Figura 3.2 se observan los parámetros utilizados por la herramienta Lighthouse para medir el rendimiento de una aplicación web. La métrica First Contentful Paint o Primer Dibujo de contenido mide cuanto tiempo pasa desde que el navegador realiza la petición hasta que aparece en la pantalla el maquetado de la aplicación con los estilos aplicados. Por otro lado, la métrica First CPU idle o primer estado ocioso de la CPU mide cuanto tiempo pasa la CPU procesando todos los archivos y la lógica necesaria para el primer pintado con sentido, o sea, el momento en que la aplicación esta funcional.

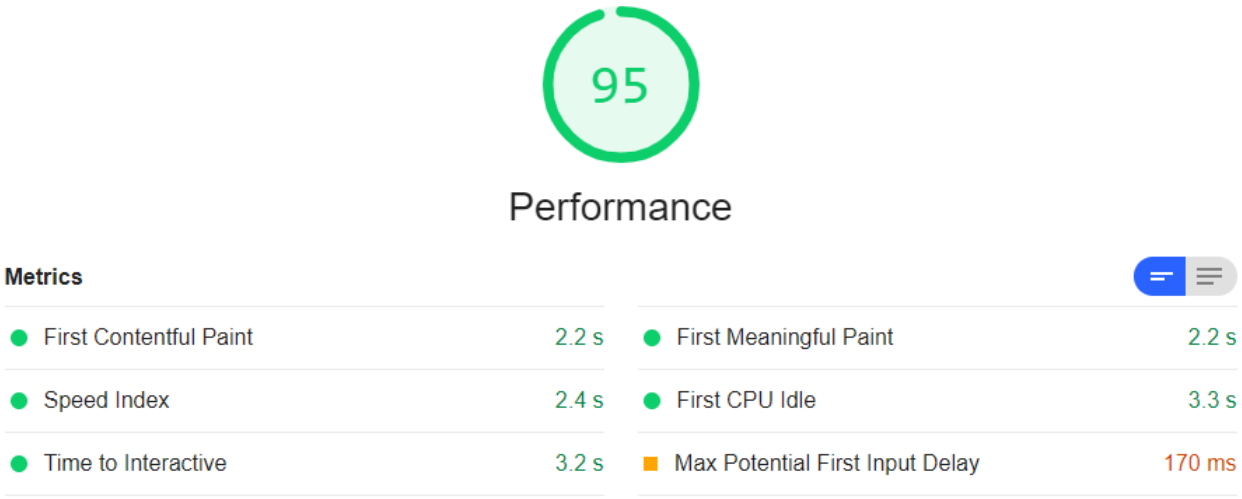


Figura 3.2 Resultados del rendimiento.

La accesibilidad (Figura 3.3) mide mayormente la integración con los screen readers o lectores de pantallas, estos son software utilizados por las personas con discapacidades visuales o que le impiden interactuar con los dispositivos como el resto de las personas, para una correcta integración se utilizan etiquetas especiales que permiten a los software leer al usuario la página y de esta forma este pueda entender la disposición y el objetivo de los componentes y como interactuar con estos.

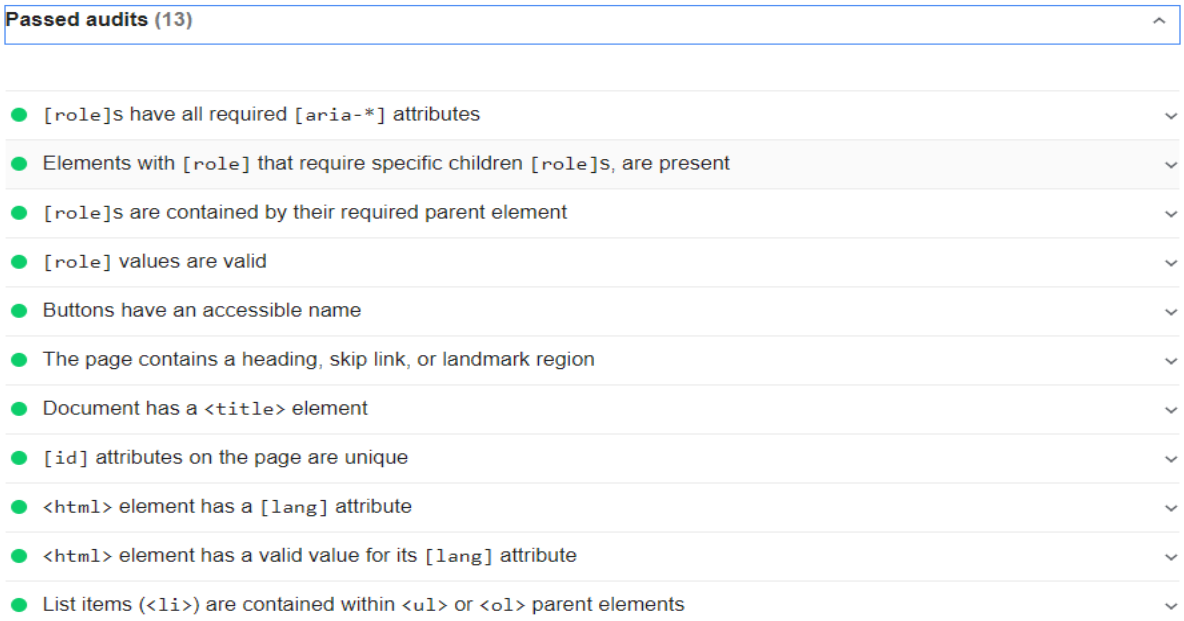
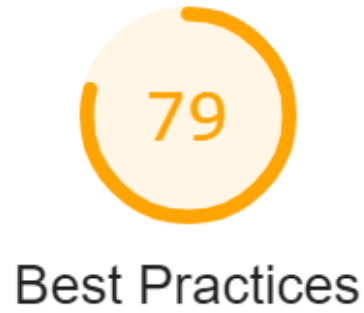


Figura 3.3 Resultados de la accesibilidad.

Las Best Practices(Figura 3.4) o buenas prácticas son soluciones a problemas comunes que están probadas son las mejores posibles hasta el momento, ya sea que aumentan el rendimiento, la usabilidad o el mantenimiento del código, esta métrica se encarga de verificar la implantación de dichas prácticas en el sitio.



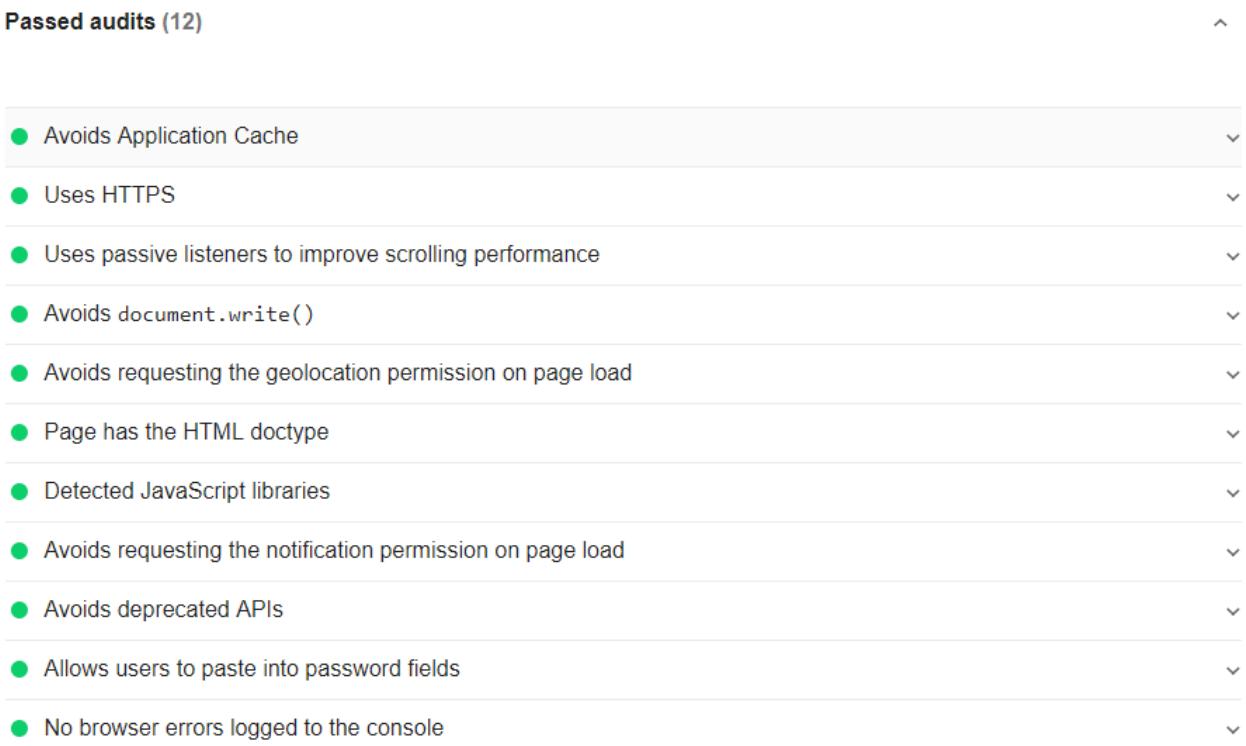


Figura 3.4 Resultados de las buenas prácticas.

El SEO (Figura 3.5) u optimización para motores de búsqueda (Search Engine Optimization), mide el uso de las etiquetas de *keywords* o palabras claves existentes en el sitio web que ayudan a los *web crawlers* o indexadores web que dichos motores emplean para analizar el contenido de la página y así determinar el objetivo y poder posicionarlo mejor en los resultados de las búsquedas. El SEO no es importante para el software dado que no va a estar en internet pero mide la correcta arquitectura de la información, o sea, la disposición del sitio.

Figura 3.5 Resultados de la optimización para motores de búsqueda.

Pruebas Unitarias

Realizar pruebas unitarias en Django es muy sencillo, debido a que trae un fichero que permite la realización de pruebas automatizadas. Este fichero tiene como nombre tests.py. Los tests son rutinas simples que chequean el funcionamiento de nuestro código. Afectan diferentes niveles. Algunos tests pueden aplicar a un detalle menor - un método particular de un modelo devuelve el valor esperado, mientras otros verifican el funcionamiento general del software. Lo que diferencia a los tests automatizados es que el trabajo de hacer las pruebas lo hace el sistema por uno. Uno crea un conjunto de tests una vez, y luego a medida que se hacen cambios a la app, se puede verificar que el código todavía funciona como estaba originalmente pensado, sin tener que usar tiempo para hacer testing manual.

Test realizados para comprobar el funcionamiento del software:

1. Test para comprobar la entrada de dos nuevos productos a la base de datos y verificar la existencia de los mismos.

```
class ProductoTestCase(TestCase):
    def setUp(self):
        Producto.objects.create(nombre="Cilindro 10kg")
        Producto.objects.create(nombre="Cilindro 10kg Nuevo")

    def test_operario(self):
        li = Producto.objects.get(nombre="Cilindro 10kg")
        ca = Producto.objects.get(nombre="Cilindro 10kg Nuevo")
        self.assertEqual(li.nombre, 'Cilindro 10kg')
        self.assertEqual(ca.nombre, 'Cilindro 10kg Nuevo')
```

2. Test para comprobar la entrada de un nuevo operario a la base de datos y verificar la correcta creación del mismo.

```
class OperarioTestCrear(TestCase):
    def test_operario(self):
        a1 = Operario.objects.create(nombre="Carmen Alfonso Diaz")
        a2 = Operario.objects.get(nombre="Carmen Alfonso Diaz")
        self.assertIs(a1.nombre,"Carmen Alfonso Diaz")
```

3. Test para comprobar si se produjo un error en el proceso de borrado de un operario y la existencia del mismo en la base de datos comparando si n es mayor que cero.

```
class OperarioTestCaseEliminar(TestCase):
    def setUp(self):
        Operario.objects.create(nombre="Operario Test")

    def test_operario_error_eliminar(self):
        item = Operario.objects.get(nombre="Operario Test")

        n = len(Operario.objects.filter(nombre="Operario Test"))
        self.assertEqual(True,n>0)
```

4. Test para borrar un operario

```
def test_operario_eliminar(self):
    item = Operario.objects.get(nombre="Operario Test")
    item.delete()

    n = len(Operario.objects.filter(nombre="Operario Test"))
    self.assertEqual(n,0)
```

5. Test para comprobar la existencia de la vista formulario y su correcta ubicación.

```
class OperarioTestCaseView(TestCase):

    def setUp(self):
        pass

    def test_operario_view_index(self):
        client = Client()
        response = client.get(reverse('URLsitio:formulario'))
        self.assertEqual(302,response.status_code)
```

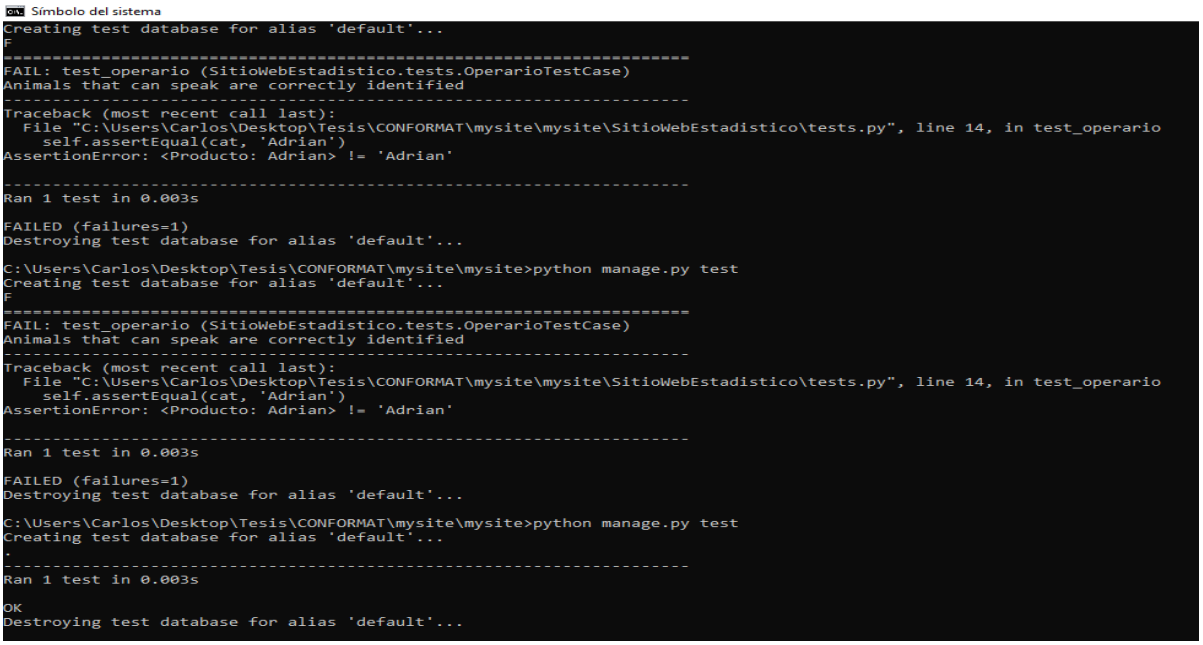
6. Test para comprobar la no existencia del formulario1.

```
    def test_operario_view_index_nofound(self):
        client = Client()
        response = client.get('/formulario/')
        self.assertEqual(302,response.status_code)

        response = client.get('/formulario1/')
        self.assertEqual(404,response.status_code)
```

Progreso de las pruebas

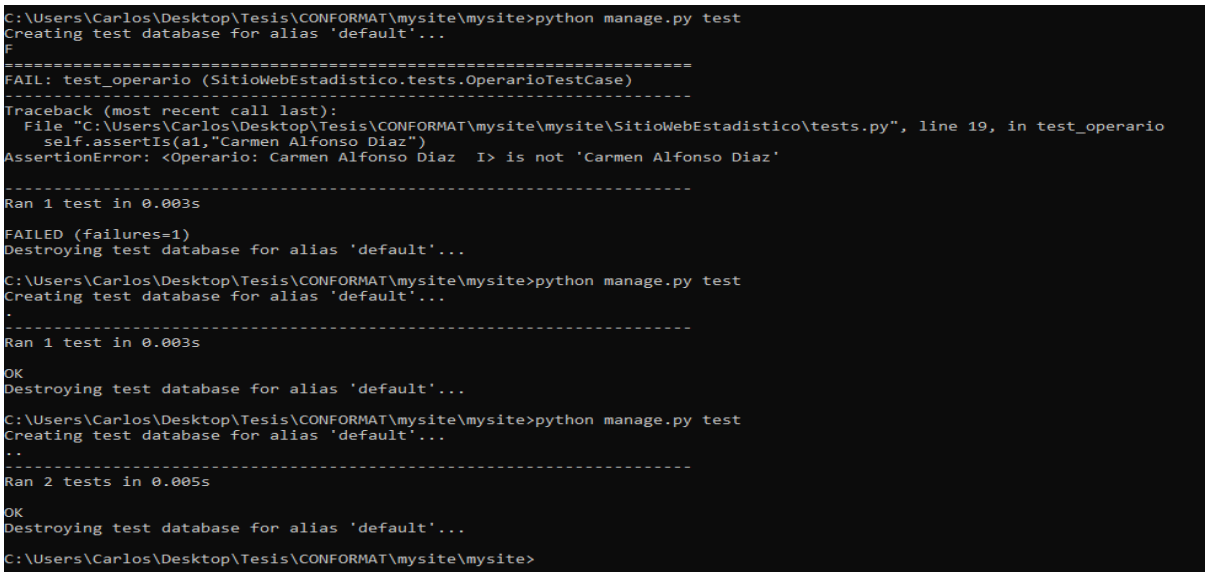
Prueba 1

Prueba fallida (Test 1) desigualdad entre los datos de salida y el resultado esperado.



Prueba 2

Pruebas aceptadas (Test 1 y 2)



Prueba 3

Una fallida (Test 3) siendo la cantidad existente en la base de datos diferente de cero.

Dos aceptadas (Test 1 y 2)

```
C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
F..
======================================================================
FAIL: test_operario (SitioWebEstadistico.tests.OperarioTestCaseEliminar)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite\SitioWebEstadistico\tests.py", line 30, in test_operario
    self.assertEqual(n,0)
AssertionError: <QuerySet []> != 0

----------------------------------------------------------------------
Ran 3 tests in 0.010s

FAILED (failures=1)
Destroying test database for alias 'default'...

C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
F..
======================================================================
FAIL: test_operario (SitioWebEstadistico.tests.OperarioTestCaseEliminar)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite\SitioWebEstadistico\tests.py", line 30, in test_operario
    self.assertEqual(n,0)
AssertionError: 1 != 0

----------------------------------------------------------------------
Ran 3 tests in 0.008s

FAILED (failures=1)
Destroying test database for alias 'default'...
```

Prueba 4

Tests del 1 al 4 correctos.

```
C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
....
----------------------------------------------------------------------
Ran 4 tests in 0.012s

OK
Destroying test database for alias 'default'...
```

Prueba 5

Test 5 fallido debido a que la ubicación del formulario es 302(correcta), diferente de 404 y validando la existencia del mismo.

```
C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
..F..
======================================================================
FAIL: test_operario_view_index (SitioWebEstadistico.tests.OperarioTestCaseView)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite\SitioWebEstadistico\tests.py", line 51, in test_operario_view_index
    self.assertEqual(404,response.status_code)
AssertionError: 404 != 302

----------------------------------------------------------------------
Ran 5 tests in 0.754s

FAILED (failures=1)
Destroying test database for alias 'default'...
```

Test del 1 al 5 aceptados.

```
C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
.....
----------------------------------------------------------------------
Ran 5 tests in 0.791s

OK
Destroying test database for alias 'default'...
```

Prueba 6

Todos los test correctos.

Símbolo del sistema

```
C:\Users\Carlos\Desktop\Tesis\CONFORMAT\mysite\mysite>python manage.py test
Creating test database for alias 'default'...
......
----------------------------------------------------------------------
Ran 6 tests in 0.746s

OK
Destroying test database for alias 'default'...
```

Análisis de los resultados de las pruebas

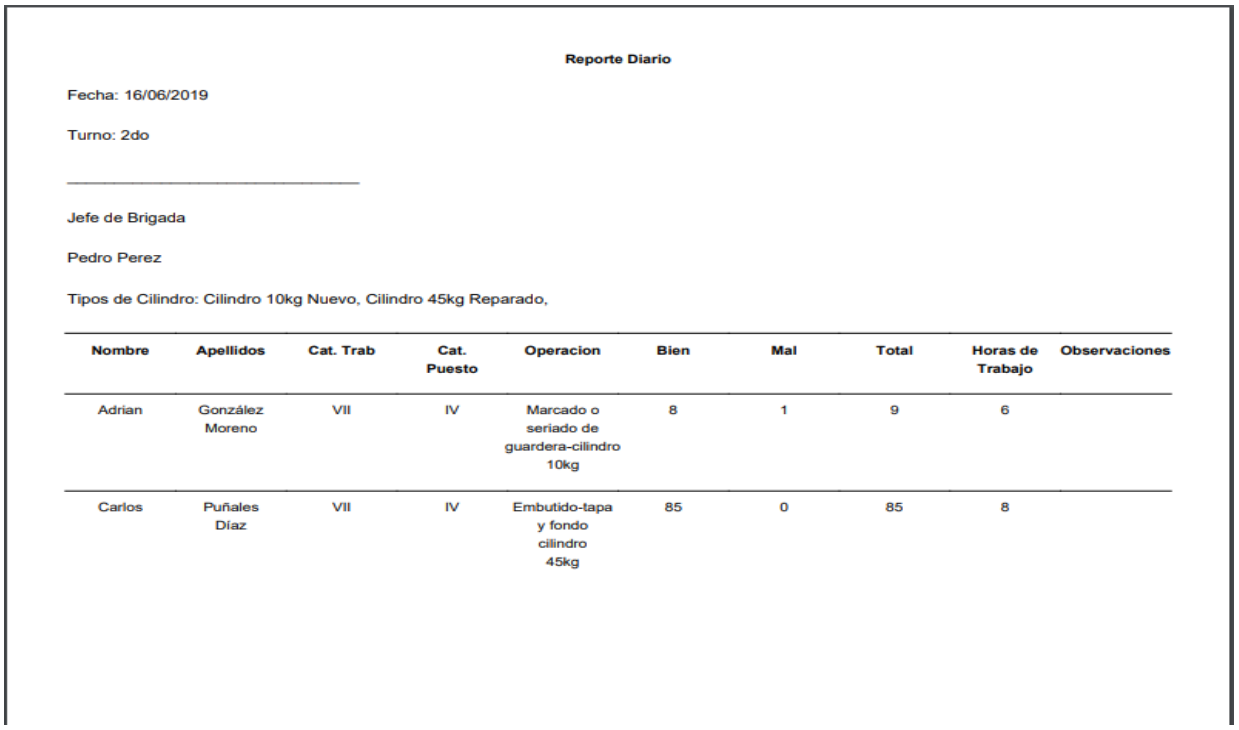
Después de desarrollar todo un proceso de pruebas se lograron resultados satisfactorios, pues tras la detección de diferentes errores que impedían el óptimo funcionamiento de la aplicación, pudiendo hacerle entrega al cliente en el tiempo establecido, un producto funcional y que satisface todas sus necesidades.

Interfaces principales del sitio web

Interfaz de datos de producción

Interfaz de organización del trabajo

Interfaz de reporte final



Reporte Diario

Fecha: 16/06/2019

Turno: 2do

_____

Jefe de Brigada

Pedro Perez

Tipos de Cilindro: Cilindro 10kg Nuevo, Cilindro 45kg Reparado,

Nombre	Apellidos	Cat. Trab	Cat. Puesto	Operacion	Bien	Mal	Total	Horas de Trabajo	Observaciones
Adrian	González Moreno	VII	IV	Marcado o seriado de guardera-cilindro 10kg	8	1	9	6	
Carlos	Puñales Díaz	VII	IV	Embutido-tapa y fondo cilindro 45kg	85	0	85	8	

Conclusiones del capítulo

En este capítulo se mostraron los elementos de prueba y casos de prueba que se aplicaron al software. Se explicaron las estrategias de prueba, además, las pruebas se convierten en una herramienta de desarrollo más y un paso imprescindible para obtener un correcto funcionamiento de un software.

Como resultado final se obtiene una aplicación Web con una apariencia atractiva y fácil de usar, adaptable a cualquier tipo de dispositivos, con todas las funcionalidades requeridas, y satisface todas las expectativas del cliente.

CONCLUSIONES

Durante el desarrollo de la presente investigación, se dio cumplimiento a los objetivos planteados, teniendo como conclusiones las siguientes:

A partir del análisis de la problemática planteada y los antecedentes del proceso, se entendió la necesidad de desarrollar una aplicación Web para la Empresa CONFORMAT ¨Noel Fernández¨, que garantice el acceso a la información actualizada de sus procesos y así monitorear el estado de la producción y la conformación del reporte final de cada turno.

El uso de la metodología SCRUM y la selección del conjunto de herramientas y tecnologías facilitaron el desarrollo del software, permitiendo la correcta implementación de la herramienta, lo cual quedó reflejado en todos los artefactos obtenidos a lo largo del desarrollo.

La realización de pruebas de software ayudó a elevar la calidad y robustez del sistema, librándolo de errores y desperfectos técnicos.

RECOMENDACIONES

Se recomienda poner en uso de la aplicación Web para garantizar el mejoramiento de los procesos dentro de la empresa.

Se recomienda darle continuidad al software para la implementación de nuevas prestaciones tales como la posibilidad de comparar las tablas de datos de producción y la posibilidad de darle una solución a las incidencias planteadas por cada turno laboral.

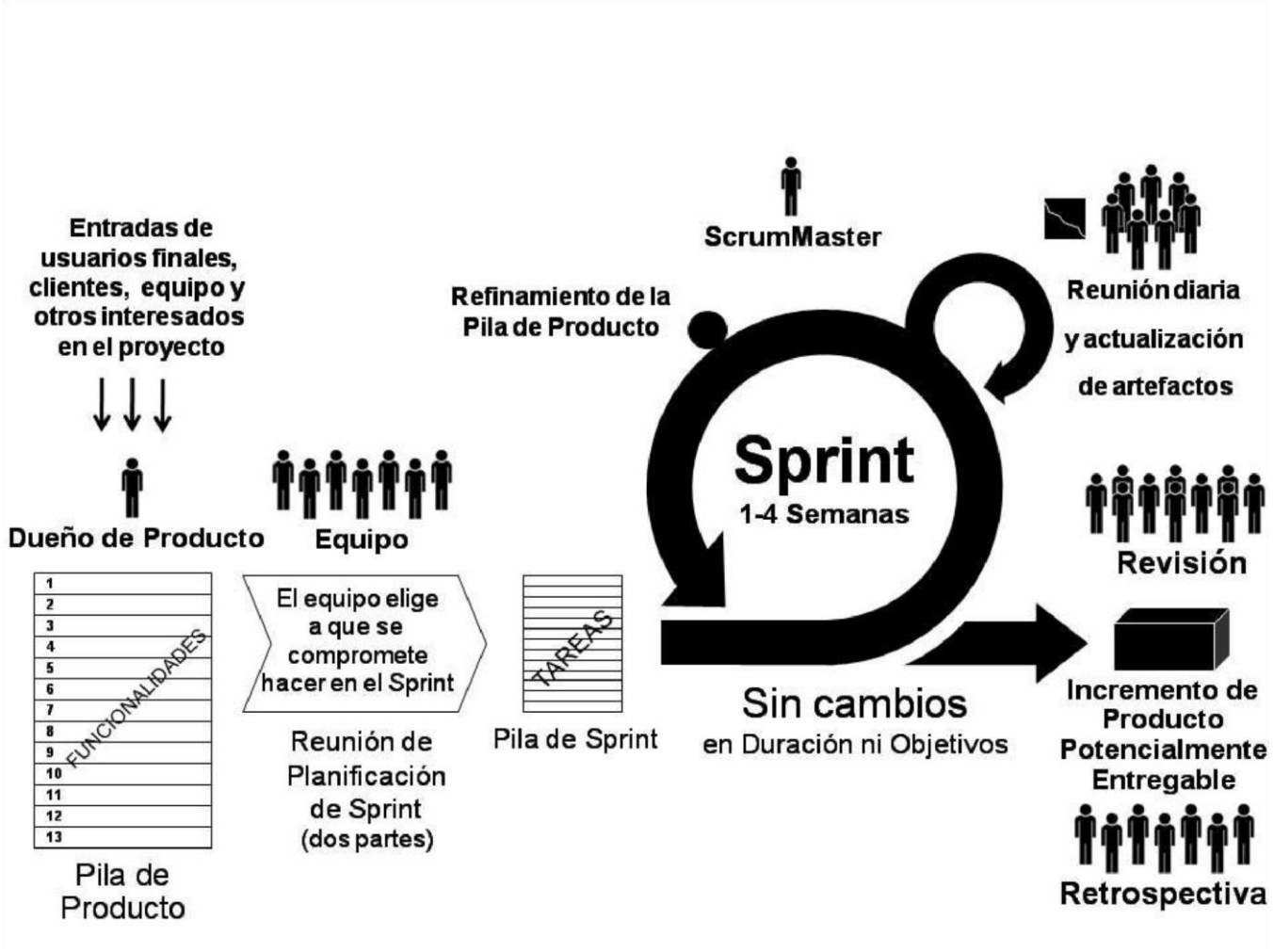
Bibliografía

1. Angel Villegas, I. H. (2008). *Revista INGENIERÍA UC.* Venezuela.
2. Asensio, R. M.-B. (2019). *Lenguajes de programación HTML y CSS.*
3. Asensio, R. M.-B. (s.f.). *Desarrollo de Aplicaciones web.* Recuperado el Junio de 2019
4. Beck K., C. A. (2005). *Extreme Programming Explained: Embrace Change.*
5. Cantón, A. (2017). *Manual de HTML5 en español.*
6. Cascio, B. (2017). *Requerimientos, Técnicas de Especificación y metodologías Ágiles.*
7. Chart.js. (s.f.). Recuperado el 2019, de Using Chart.js with Django - GoDjango.com
8. Cochran, D. (2012). *Twitter Bootstrap Web.*
9. Danenner, G. (s.f.). Recuperado el 2019, de https://medium.com/@giezendanenner/running-lighthouse-reports-on-the-command-line-1691a1b06a56
10. Fundation, F. S. (1986). *La Definición de Software Libre. [En línea] 1986. [Citado el: 14 de mayo de 2019.] http://www.gnu.org/philosophy/free-sw.es.html#translations.*
11. Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y JavaScript.* Recuperado el 2019
12. GUIDE, H. --. (2018). Obtenido de http://www.tutorialspoint.com/highcharts/highcharts_quick_guide.htm
13. JQuery. (2015). *Sitio web oficial.* Recuperado el 2019, de http://jquery.com/
14. Kaplan, A. H. (2015). *La Guía definitiva de Django.*
15. Kaplan-Moss, A. H. (s.f.). *El libro de Django.* Recuperado el 14 de Junio de 2019
16. Kniberg, H. (2017). *Scrum y XP desde las trincheras Como hacemos Scrum.*
17. Llonch, J. (2005). *Curso HTML + CSS.* Recuperado el Junio de 2019
18. Martos, M. A. (2018). *Manual de CSS.*
19. Montiel, D. P. (s.f.). *SQLite 3.* Recuperado el Junio de 2019
20. Mora, S. L. (s.f.). *Programación de aplicaciones web: Historia, principios básicos y clientes web.* Recuperado el 2019

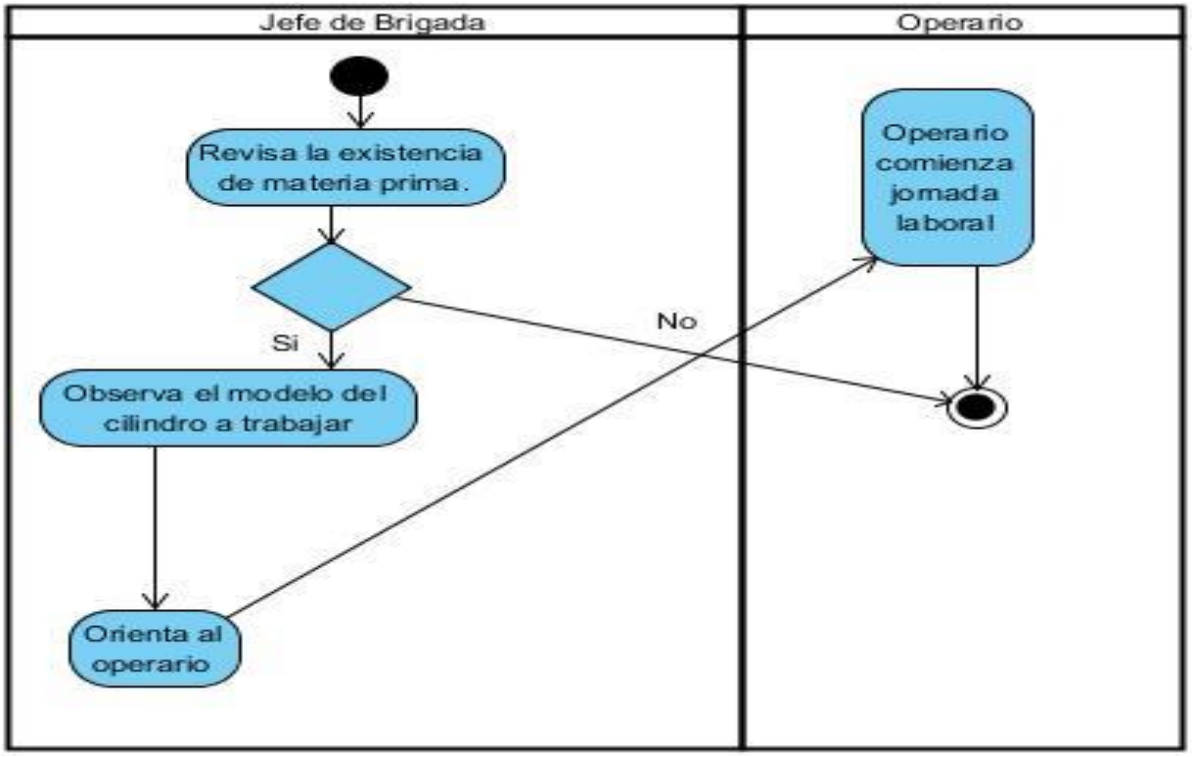
21. Murcia, U. d. (s.f.). *Manual Básico de Creación de Páginas Web.* Recuperado el Junio de 2019
22. Pérez, J. (2016). *Introducción a JavaScript.* Recuperado el Junio de 2019
23. Pete Deemer, G. B. (2012). *Scrum Primer.*
24. Risueño, P. (2015). *Comenzando con Bootstrap, framework responsive.*
25. Rossum, G. v. (2009). *El tutorial de Python .*
26. Rossum, G. v. (2017). *El Tutorial de Python.*
27. Rossum, G. v. (s.f.). *Guía de aprendizaje de Python.* Recuperado el Junio de 2019
28. Sánchez, T. L. (2003). *Metodologías Ágiles en el desarrollo de Software.*
29. SQLite. (s.f.). Recuperado el Junio de 2019, de https://unipython.com/una-base-de-datos-sqlite/
30. Tobias Kahlert, K. G. (2016). *Visual Studio Code Tips and Tricks Vol.1.* Germany.
31. UML, V. P. (2019). Obtenido de http://www.visualparadigm.com/
32. Xhtml2. (2018). *xhtml2pdf Documentation.* Recuperado el Junio de 2019, de https://buildmedia.readthedocs.org/media/pdf/xhtml2pdf/latest/xhtml2pdf.pdf

ANEXOS

Anexo 1: Diagrama de funcionamiento de la metodología SCRUM (Pete Deemer, 2012)



Anexo 2: Diagrama de Actividades



Anexo 3: Diagrama de Actividades