

UNIVERSIDAD DE MATANZAS

Facultad: Ciencias Técnicas

Carrera: Ingeniería Informática



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

Título: Aplicación web de apoyo al proceso de soldadura al arco con metal revestido.

Autor: Marlon Ortega Méndez.

Tutor: Ms.C. Mayli Estopiñán Lantigua.

Resumen

La soldadura de metal por arco eléctrico con electrodos revestidos es un proceso popular para uniones de piezas y componentes mecánicos. La selección de los parámetros apropiados en la creación de las cartas tecnológicas para el proceso es un aspecto importante con vistas a obtener uniones de alta calidad. En la presente investigación se propone como objetivo desarrollar una aplicación web para la creación de cartas tecnológicas asociadas al proceso de soldadura al arco por electrodo revestido. Como tecnologías principales de desarrollo se utilizaron los lenguajes de programación HTML(HyperText Markup Language), CSS(Cascading Style Sheets) y JavaScript para la creación del FrontEnd de la web y el lenguaje Python junto a su Framework Django para el Backend, como editor de código Visual Studio Code y la metodología ágil de desarrollo, XP. La metodología propuesta tiene en cuenta las normas actualmente vigentes en Cuba, así como otras de amplio uso internacional

Abstract

Metal arc welding with stick electrodes is a popular process for joining mechanical parts and components. The selection of the appropriate parameters in the creation of the technological charts for the process is an important aspect with a view to obtaining high-quality joints. This research proposes as objective the design and development of a web application for the creation of technological charts associated with the arc welding process by coated electrode, as well as giving continuity to the degree thesis of the mechanical engineer Jorge Ernesto Pérez Pedroso of the year 2016 about the algorithmization of the methodology for the design of manual electric arc welding technologies with steel-coated electrode. As main development technologies, the programming languages HTML(HyperText Markup Language), CSS(Cascading Style Sheets) and JavaScript will be used for the creation of the FrontEnd of the web and the Python language together with its Django Framework for the Backend, as a code editor Visual Studio Code and the agile development methodology XP. The proposed methodology takes into account the standards currently in force in Cuba, as well as others widely used internationally.

Indice

| | |
|---|----|
| Introducción..... | 1 |
| Desarrollo | 4 |
| Capítulo 1: Marco teórico referencial | 4 |
| 1.1: ¿Qué es una carta tecnológica? | 4 |
| 1.2: Descripción de la organización Objeto de estudio | 5 |
| 1.2: Descripción del flujo del proceso | 5 |
| 1.3: Metodologías y herramientas | 6 |
| 1.3.1: Metodología ágil de desarrollo | 6 |
| 1.3.2 Metodología Extreme Programming..... | 6 |
| 1.4: Tecnologías de desarrollo..... | 8 |
| 1.4.1: HyperText Markup Language o HTML | 8 |
| 1.4.2: CSS o Cascading Style Sheets | 10 |
| 1.4.3: JavaScript..... | 10 |
| 1.4.4: Python..... | 11 |
| 1.4.5: Django..... | 12 |
| 1.4.6: SQLite3..... | 12 |
| 1.4.7: Visual Studio Code(VSCoDe)..... | 12 |
| 1.5: Conclusiones Parciales del Capitulo: | 13 |
| Capítulo 2: Solución propuesta..... | 14 |
| 2.1: Descripción de la Solución..... | 16 |
| 2.2: Etapa de Planificación | 16 |
| 2.2.1: Equipo de Trabajo..... | 16 |
| 2.2.2: Historias de Usuario..... | 17 |
| 2.2.3: Plan de iteraciones | 18 |
| 2.2.4: Plan de Entregas..... | 19 |
| 2.2.5: Análisis de Costos..... | 20 |
| 2.3: Etapa de Diseño | 22 |
| 2.3.1: Prototipo de Interfaz de usuario..... | 22 |
| 2.3.2: Listado de Historias de Usuario | 24 |
| 2.3.3: Listado de Tareas de Ingeniería | 29 |

| | |
|--|----|
| 2.3.4: Modelo de la base de datos | 32 |
| 2.4: Pruebas..... | 33 |
| 2.4.1: Plan de pruebas | 33 |
| 2.4.1: Pruebas de Aceptación | 36 |
| 2.5: Conclusiones parciales del capítulo | 43 |
| Capítulo 3: Resultados de la investigación | 43 |
| 3.1: Producto terminado..... | 43 |
| Conclusiones..... | 47 |
| Recomendaciones | 48 |
| Referencias bibliográficas..... | 49 |
| Anexos | 51 |

Introducción

La soldadura por arco con electrodo metálico revestido (shielded metal arc welding, SMAW) es uno de los procesos mecánicos más comunes utilizados para la unión de elementos metálicos. Esto se debe a sus múltiples ventajas, dentro de las que cabe destacar la flexibilidad de su aplicación que incluye todas las posiciones y un amplio rango de espesores, el relativamente bajo costo del equipamiento necesario y la posibilidad de utilizarse tanto en talleres como a pie de obra (Cary, 1998).

El diseño de tecnologías de SMAW es relativamente sencillo (al menos, si se compara con otros procesos de manufactura, como el maquinado o la conformación), pero es realizado usualmente a un grupo grande de costuras en cada elemento constructivo. Esto hace que, acumulativamente, se vuelva una tarea tediosa y propensa a errores e imprecisiones humanas debidas a la repetición y el cansancio. Debe sumársele, a lo anterior, que la consulta de normativas y manuales para la selección de preparaciones de bordes y consumibles hace aún más lento y engorrosa la tarea de diseñar tecnologías de soldadura. Un aspecto a tener en cuenta, en los procesos de soldadura, es que, la importancia de las uniones realizadas (por el impacto de seguridad humana, ambiental o económico que puede tener un fallo), hace que la formalización de la tecnología, en especificaciones o cartas tecnológicas sea imprescindible (incluso, muchas veces legalmente obligatorio) con el fin de garantizar la trazabilidad del proceso de cara a posibles auditorías técnicas.

Considerando todo lo anterior, resulta evidente la conveniencia de automatizar el proceso de diseño de tecnologías de SMAW, en forma de una aplicación informática, de modo que permita acelerar el proceso y mejorar la precisión y confiabilidad de los resultados. Aunque existen productos de software con este propósito, su costo es elevado y no siempre se adaptan a los requerimientos de la industria cubana. Por otro lado, otros productos desarrollados en Cuba, tal como **Soldasoft** se han quedado obsoletos tanto por los algoritmos y datos que utiliza como por su compatibilidad con los sistemas operativos contemporáneos.

Como antecedente principal de esta investigación tenemos el proyecto de tesis del ingeniero mecánico Jorge Ernesto Pérez Pedroso, el cual en su investigación diseña y formaliza el algoritmo principal usado para la creación de las tecnologías asociadas al proceso de soldadura al arco con electrodo revestido (Pedroso, 2016), así como desarrolla una aplicación desktop para el uso de este algoritmo, una foto del algoritmo puede ser vista en el Anexo 1.

El algoritmo general del diseño de tecnología de SMAW, consiste en cinco etapas fundamentales. La primera etapa consiste en la entrada de los datos requeridos para diseñar la tecnología: la geometría de la unión, las propiedades del metal base y las características de la máquina de soldadura a utilizar. En la segunda etapa se selecciona la preparación de bordes a utilizar y se determina el esquema de soldadura para la costura (es decir, si se utilizarán pasadas de raíz, de relleno y de sellado). En la tercera etapa, se seleccionarán los electrodos a utilizar en cada una de las etapas y se calculará

el número de pasadas. En la cuarta etapa, para cada uno de los cordones a realizar, se selecciona el electrodo a utilizar y se calculan los parámetros tecnológicos (intensidad de la corriente, consumo de electrodos y de energía eléctrica, y costos del proceso). Finalmente, en la quinta etapa, se determina parámetros tecnológicos comunes para toda la costura y se selecciona el equipamiento auxiliar.

Como segundo antecedente principal tenemos la aplicación desarrollada para diseñar tecnologías de SMAW que recibe el nombre de Opifex Spark/SMAW. Fue implementada en lenguaje C++, utilizando el entorno de desarrollo y las librerías de Qt. Aprovechando las funcionalidades de Qt, la aplicación se concibió como multiplataforma, pudiendo funcionar en MS Windows o en Linux.

La aplicación cuenta con una ventana principal cuyo contenido es la carta tecnológica del proceso de soldadura. Dicha carta ha sido elaborada siguiendo las recomendaciones de la ASME BPVC.IX (2015) y se va llenando según se van entrando los datos correspondientes y realizando los cálculos necesarios para determinar los parámetros del régimen de soldadura y los costos del proceso. Una vista de la ventana principal de dicho software puede ser vista en el Anexo 2.

Para el diseño de las tecnologías de SMAW, se utilizaron las indicaciones dadas por las normativas cubanas (NC-ISO, 2004) (NC-ISO, 2011) e internacionales (AWS, 2015) (BPVC.IX, 2015), así como las recomendaciones dadas por la literatura especializada (JUSTEL RAMOS, 2012) (Cary, 1998).

A partir de lo anteriormente descrito podemos definir como **Problema de Investigación:** ¿Cómo contribuir a la eficiencia del proceso de soldadura al arco con metal revestido?

Teniendo como **Objeto de estudio:** El proceso de soldadura al arco por metal revestido (SMAW) y **Campo de acción:** la informatización del proceso de soldadura al arco por metal revestido (SMAW)

A partir de lo descrito deducimos la siguiente **Hipótesis:** Con el desarrollo de una aplicación web que contribuya a la gestión de la información de cartas tecnológicas asociadas al proceso de soldadura al arco por metal revestido (SMAW) se logrará contribuir a la eficiencia de este proceso.

Teniendo como **Variable independiente:** la aplicación web que contribuya a la gestión de la información de cartas tecnológicas asociadas al proceso de soldadura al arco por metal revestido (SMAW). Y como **Variable dependiente:** la eficiencia del proceso de soldadura al arco por metal revestido (SMAW).

Se propone como **objetivo general** de la presente investigación, desarrollar una aplicación web para contribuir a la eficiencia del proceso de soldadura al arco con metal revestido.

Para lograr el cumplimiento del objetivo general, como **objetivos específicos** se proponen:

1. Seleccionar los métodos y técnicas que permitan obtener los fundamentos teóricos y la comprensión necesaria de los distintos elementos del tema que aborda la investigación.
2. Definir los requisitos del sistema, así como diseñar el modelo de datos e interfaz de usuario que sirvan como punto de partida en la adecuada determinación de las herramientas y metodología de desarrollo a utilizar.
3. Implementar la aplicación web que contribuya a la eficiencia del proceso de soldadura al arco con metal revestido.
4. Validar el funcionamiento correcto de la aplicación web desarrollada.

Como **Resultados Esperados de la Investigación** tenemos:

- La creación de una aplicación web que contribuya a la creación de cartas tecnológicas asociadas al proceso de soldadura al arco por metal protegido.
- La disminución de errores humanos en el proceso de creación de dichas cartas tecnológicas.
- El resguardo de la información asociada al proceso en una base de datos.

Como **justificación** de la investigación tenemos que es necesario el desarrollo de esta aplicación web principalmente debido a que el antecedente que existe es una aplicación de escritorio lo que trae como inconveniente que los demás actores que intervienen en la aprobación de la carta solo la pueden ver una vez impresa o trasladándose a la máquina donde se encuentra instalada, mientras que una aplicación web permite el acceso remoto a la información. Además de la aplicación de escritorio no guarda la información respectiva en una Base de datos, lo que impide que se pueda acceder a la información histórica y poder tomar decisiones sobre el uso y desarrollo de las cartas.

En correspondencia con lo tratado anteriormente, el documento queda estructurado de la siguiente forma:

Capítulo I: Marco teórico referencial, se reflejan los conceptos, definiciones y fundamentos asociados al problema de estudio tratado. Es descrito y analizado el flujo actual de trabajo y los antecedentes. Son abordadas las principales características de las herramientas y tecnologías empleadas para el desarrollo del software, entre los que se encuentran el sistema de gestión de base de datos y lenguajes de programación empleados con ese fin.

Capítulo II: Solución Teórica al Problema Científico, se expone la propuesta de solución al problema de investigación, la organización y disposición del proyecto mediante la presentación de una planificación inicial haciendo uso del marco de trabajo de desarrollo

de software XP. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software. Son aplicadas pruebas al software para determinar la funcionalidad y calidad

Capítulo III: Se realiza el análisis de los resultados obtenidos con el objetivo de entregar al cliente un producto terminado que satisfaga sus requerimientos.

Desarrollo

Capítulo 1: Marco teórico referencial

1.1: ¿Qué es una carta tecnológica?

La carta tecnológica es un documento asociado a un proceso de terminado que representa una base sólida de planificación de los recursos, tanto materiales, humanos como financieros, que se requieren en el comportamiento del proceso al que están asociadas, permitiendo conocer, desde su inicio, los principales indicadores técnico-económicos que regirán el plan de producción (Granados Delgado, 2018).

Una vez confeccionada la carta tecnológica, se convierte en guía e instrumento de trabajo para la realización de las programaciones, evitando la improvisación y los gastos innecesarios de materiales y salarios sin respaldo, convirtiéndose en la columna vertebral de la planificación de cualquier actividad

Cada carta Tecnológica asociada a un proceso presenta indicadores únicos dependiendo de las herramientas y cálculos que se realizan en dicho proceso y se establecen como una forma de documentación a la hora de comprobar su calidad o de validar su ejecución

En el proceso de soldadura al arco por metal revestido (SMAW) se usa una carta tecnológica con indicadores únicos del proceso. Se han desarrollado investigaciones que se centra en el desarrollo de cartas tecnológicas para la soldadura al arco con metal revestido de acero al carbono. Se examinaron varios parámetros de soldadura y se generaron cartas tecnológicas que indican las combinaciones óptimas de corriente, tensión y velocidad de alimentación del electrodo para obtener soldaduras de alta calidad (Parmar, 2017). Igualmente se realizan estudios centrados en realizar pruebas de soldadura al arco con metal revestido en acero inoxidable. Donde se desarrollaron cartas tecnológicas específicas para diferentes grados de acero inoxidable, proporcionando información sobre los parámetros de soldadura adecuados, como la corriente, la tensión y la velocidad de alimentación del electrodo, para lograr soldaduras de alta calidad (Syed, 2018). A la hora tratar con la optimización de los parámetros de soldadura al arco con metal revestido utilizando cartas tecnológicas. Se presenta un enfoque sistemático para el uso de las cartas tecnológicas en la selección y ajuste de los parámetros de soldadura, lo que resulta en una mejora significativa en la calidad de las soldaduras. (Khan, 2020)

AL entrar en el tema de la informatización del proceso de desarrollo de soldaduras al arco por metal revestido (SMAW) se deben tener en cuenta indicadores extra en cuanto a la eficiencia del proceso. Se desarrollan sistemas de gestión de soldadura que utiliza cartas tecnológicas informatizadas en el proceso de soldadura al arco por metal revestido. Se describen los beneficios de la informatización y se presentan los resultados de su implementación en un casos de estudios (García, 2015). En otros casos se proponen una metodologías para optimizar el proceso de soldadura al arco por metal revestido utilizando cartas tecnológicas informatizadas con resultados experimentales donde se discuten las ventajas de la informatización en términos de calidad y productividad (Martínez, 2019). Así como similares sistemas con características y beneficios de utilización centrados en términos de eficiencia y control del proceso (Rodríguez, 2020)

1.2: Descripción de la organización Objeto de estudio

La presente investigación se desarrolló en el Centro de Estudios de Fabricación Avanzada y Sostenible (CEFAS). Este centro perteneciente a la Facultad de Ciencias Técnicas de la Universidad de Matanzas, está dirigido a la investigación científica y la innovación en el área de los procesos de manufactura mecánica. Entre otras temáticas, el CEFAS trabaja en la modelación, optimización, monitoreo y control de procesos de maquinado, soldadura y conformación. También desarrolla investigaciones en el campo de procesos de fabricación no convencionales como el micro mecanizado y la fabricación aditiva. El CEFAS realiza no sólo trabajos para el medio académico sino también para el entorno industrial, materializados en proyectos de investigación empresariales. Lleva a cabo, además, numerosas acciones de colaboración con otros grupos y centros de investigación cubanos e internacionales. Las principales líneas de trabajo del CEFAS incluyen la fabricación inteligente que se enfoca al desarrollo y la asimilación de sistemas de fabricación basados en el uso de las tecnologías de la información y las comunicaciones, la automatización y la fabricación avanzada. La fabricación de elementos biomédicos, dirigida al desarrollo de diseños y de procedimientos de fabricación de componentes para la aplicación biomédicas incluyendo implantes, sensores y partes de equipos y la fabricación sostenible a empresas, línea centrada en el desarrollo y evaluación de sistemas de fabricación sostenibles para pequeñas y medianas empresas, con el objetivo de mejorar su eficiencia y competitividad.

1.2: Descripción del flujo del proceso

Anterior a esta investigación el flujo del proceso de creación de cartas tecnológicas asociadas al proceso de soldadura al arco por metal revestido se hacía completamente a mano. Cada tecnólogo debe construir el documento de la carta tecnológica, manuscrito, lo que implicaba la completa recolección de la información acerca de los materiales, métodos y técnicas que componen el proceso. lo que incluye la información referente al metal base de la costura, los datos de la máquina de soldadura a utilizar, el electrodo a emplear, así como la realización de todos los cálculos necesarios a mano y previo al llenado de la carta tecnológica, para plasmarlos correctamente en el documento. Esto como es lógico puede conllevar a varios problemas referentes al error humano en el

proceso dado que depende de la experticia del tecnólogo. En los Anexos 3 y 4 se pueden apreciar los diagramas de flujo del proceso de soldadura al arco por metal revestido y de creación de cartas tecnológicas asociadas a dicho proceso.

1.3: Metodologías y herramientas

1.3.1: Metodología ágil de desarrollo

Las metodologías de desarrollo de software ligeras, más conocidas como metodologías ágiles, surgen en la década de 1990, estas buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala. Las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la metodología y esto retrasa la etapa de desarrollo

El enfoque de las metodologías ágiles está teniendo una amplia efectividad en proyectos donde los requisitos son muy cambiantes, ya que en este tipo de proyectos la comunicación con el cliente debe ser fundamental y precisamente ese es uno de los principios básicos de las metodologías ágiles.

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos resumieron en cuatro postulados lo que hoy en día se conoce como el Manifiesto Ágil (Menzinsky, 2019). Los cuatro postulados mencionados en aquella reunión mencionaban los siguientes criterios: Individuos e interacciones sobre procesos y herramientas, Software que funciona sobre documentación exhaustiva, Colaboración de clientes sobre la negociación del contrato y Respuestas a cambios sobre seguir un plan. Estos cuatro postulados son considerados como la definición canónica del desarrollo ágil.

1.3.2 Metodología Extreme Programming

Extreme Programming o XP es un marco de desarrollo de software ágil que tiene como objetivo producir un software de mayor calidad para mejorar la eficiencia del equipo de desarrollo. Se trata de una metodología de desarrollo cuyo objetivo es promover la aplicación de prácticas de ingeniería apropiadas para la creación de software. Esta metodología la formuló Kent Beck, autor del primer libro sobre este ámbito llamado «Extreme Programming Explained: Embrace Change», publicado en 1999.

La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Crean que ser capaz de

adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

El Extreme Programming se desarrolla teniendo en cuenta cinco valores fundamentales. El objetivo de estos valores es que el equipo de desarrolladores trabaje bajo una mentalidad conjunta para poder colaborar y crear un producto de alto nivel.

1. Comunicación

Como comunicación entendemos no solo una buena interacción interna entre los propios miembros del equipo de desarrolladores, sino también con los clientes. El objetivo es romper las barreras entre negocio y desarrollo. Para ello, la programación XP promueve que todos los requisitos sean comunicados y trabajados con el equipo y no mediante documentación.

2. Simplicidad

Empezar con la solución más simple es clave en la programación XP. Esta metodología pone el foco en codificar las necesidades de hoy, no las de un futuro. Además, también se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Para conservar la simplicidad hay que mantener la refactorización del código, así podremos preservar el código simple a medida que va creciendo.

3. Feedback

Una de las mayores ventajas de que el cliente esté integrado en el proyecto es que su opinión sobre el estado de este lo podemos conocer en tiempo real. Gracias a que se hacen ciclos muy cortos de presentación de resultados, se minimiza el riesgo de tener que rehacer partes que no cumplen con las expectativas del cliente. También, por otro lado, ayuda a los programadores a centrarse en las tareas más importantes.

4. Respeto

El respeto mutuo es fundamental para que un equipo pueda trabajar de forma eficiente y ofrecer un buen rendimiento. Implica desde que un desarrollador no realice modificaciones que puedan tener un impacto negativo en el trabajo de un compañero hasta la forma de llegar al cliente. El respeto se manifiesta de varias formas y todas son cruciales para una mejor autoestima en el equipo, que lleva consigo un mayor ritmo de producción.

5. Valentía

Diseñar y programar para hoy y no para mañana implica valentía en la metodología XP, así como reconocer los errores tan pronto como se detecten. Ningún miembro del equipo puede perder el tiempo en intentar hacer de menos su responsabilidad en un error cometido, ya que esto significará dejar de centrarse en otras cosas e impedirá avanzar al resto, por lo que la productividad bajará.

Los principales artefactos de la metodología XP incluyen

- Historias de Usuario
- Tareas de ingeniería
- Tarjetas CRC(Clase-Responsabilidad-Colaboración)
- Plan de iteraciones
- Plan de entregas
- Modelo de negocio
- Casos de pruebas de aceptación

1.4: Tecnologías de desarrollo

Entre los distintos lenguajes de programación para la Web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente/Servidor, característica de los sistemas Web.

1. El primer grupo lo integran los lenguajes que se ejecutan en el cliente, ejemplos de ellos son: HTML, CSS, JavaScript, estos se encargan de aportar dinamismo a la aplicación en los navegadores.

2. El segundo grupo está formado por los lenguajes que se ejecutan en el servidor, dentro de los que se encuentra ASP, PHP y Java, estos lenguajes se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a bases de datos y el tratamiento de la información.

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.

A continuación, se exponen los diferentes lenguajes y herramientas utilizados para el desarrollo de esta investigación.

1.4.1: HyperText Markup Language o HTML

HTML, siglas en inglés de HyperText Markup Language ('lenguaje de marcado de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web.

Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado (W3C, 2021).

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

HTML es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

1.4.2: CSS o Cascading Style Sheets

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (W3C, 2021)

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo de extensión .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz (mediante un navegador de voz o un lector de pantalla), y dispositivos táctiles basados en el sistema Braille. También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS describe un esquema prioritario para determinar qué reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas son aplicadas con un sistema llamado de cascada, de modo que las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

1.4.3: JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,² basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas³ y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015.⁴

JavaScript se diseñó con una sintaxis similar a C++ y Java, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Su relación es puramente comercial, tras la compra del creador de Java (Sun Microsystems) de Netscape Navigator (creador de LiveScript) y el cambio de nombre del lenguaje de programación.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). Javascript es el único lenguaje de programación que entienden de forma nativa los navegadores.

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

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó⁷). En junio de 2015 se cerró y publicó la versión ECMAScript 6.8 y el 1 de junio de 2021 se publicó ECMAScript 2021 como última actualización.

1.4.4: Python

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional (Python software foundation, 2022). Es un lenguaje interpretado, dinámico y multiplataforma creado a finales de los años ochenta por Guido van Rossum en Stichting Mathematisch Centrum (CWI), en los Países Bajos, como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba.⁶

Administrado por Python Software Foundation, posee una licencia de código abierto, denominada Python Software Foundation License. Python se clasifica constantemente como uno de los lenguajes de programación más populares.

1.4.5: Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Fue desarrollado originalmente para gestionar páginas web orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt (Django software foundation, 2022).

En junio de 2008 fue anunciado que la Empresa Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio «DRY» (del inglés Don't Repeat Yourself, «No te repitas»). El lenguaje Python es usado en todos los componentes del framework, incluso en configuraciones, archivos,1 y en sus modelos de datos.

Al igual que Ruby on Rails, otro popular framework de código abierto, Django se usó en producción durante un tiempo antes de que se liberara al público; fue desarrollado por Adrian Holovaty, Simon Willison, Jacob Kaplan-Moss y Wilson Miner mientras trabajaban en World Online

1.4.6: SQLite3

SQLite3 es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. (SQLite consortium, 2023) El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

1.4.7: Visual Studio Code(VSCoDe)

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de

código (Microsoft, 2023). También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario. Visual Studio Code fue anunciado el 29 de abril de 2015 por Microsoft en la conferencia Build de 2015. Una versión preliminar se lanzó poco después. El 18 de noviembre de 2015, Visual Studio Code fue lanzado bajo la licencia MIT y su código fuente fue publicado en GitHub. También fue anunciada una nueva capacidad para agregar extensiones.

Visual Studio Code es un editor de código fuente. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos .json (por ejemplo, preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo.

1.5: Conclusiones Parciales del Capítulo:

En el desarrollo del primer capítulo se dio a conocer las bases teóricas sobre las cuales se sustenta la propuesta de esta investigación. Se analiza cómo se efectúa el proceso y se corroboró la necesidad de diseñar esta aplicación para solucionar los problemas existentes. Se definen tecnologías y técnicas que se aplicarán al diseño e implementación de la aplicación informática. Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de programación extrema XP. Se fundamenta la utilización de los lenguajes HTML, CSS y JavaScript en el lado del cliente y el lenguaje Python con Django como framework en el lado del servidor, además del uso de Visual Studio Code como entorno de desarrollo y el gestor de bases de datos SQLite, ya que se ajustan a los requerimientos del cliente y de desarrollo del software propuesto.

Capítulo 2: Diseño de la solución propuesta.

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Se aplica la Metodología XP, Extreme Programming (Programación Extrema), metodología ágil de desarrollo, con el objetivo de garantizar el diseño de un programa lo más ajustado posible y se logra como ventaja la incorporación del cliente como un miembro del equipo de desarrollo.

La Programación Extrema o XP (Extreme Programming) es un enfoque de desarrollo de sistemas que acepta lo que se conoce como buenas prácticas en esta área y las lleva al extremo (Rosado Gómez, 2012) (Suardiyana Putra, 2012). En esta metodología cabe resaltar la importancia del cliente, las pruebas, la refactorización, la simplicidad, la propiedad colectiva del código que se ven reflejadas en las cuatro prácticas esenciales de XP (Pressman, 2010):

- Entregas limitadas o pequeñas: Consiste en realizar entregas parciales de módulos del sistema.
- Semana de trabajo de 40 horas: Los equipos de desarrollo de XP trabajan de manera intensa durante una semana típica de 40 horas.
- Cliente en el sitio: Esta práctica insiste en que el cliente debe hacer parte fundamental y activa del grupo de trabajo y debe estar presente durante todo el proceso de desarrollo.
- Programación en Pareja: Con esto se busca aumentar la calidad del código, ahorrar tiempo, estimula la creatividad y la reducción de código fuente.

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. En la Figura 2 se muestra cada una de las actividades de XP, y resalta las tareas claves de cada una (Pressman, 2010) (Sommerville, 2011) :

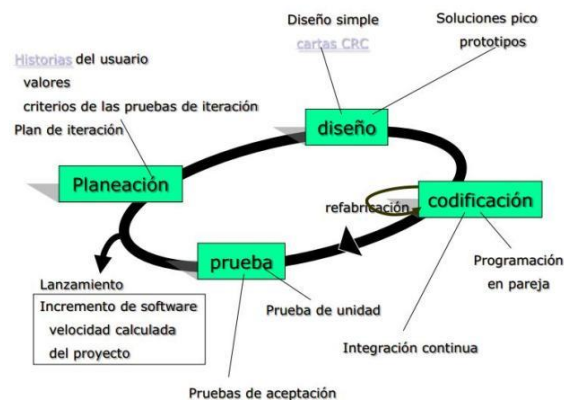


Fig 2 : Etapas de la Programación Extrema

Etapa de Planeación: Esta actividad comienza escuchando a los clientes, para entender el contexto del negocio y definir las características principales y funcionalidad que se requiere, estas características se transforman en requerimientos del negocio que se especifican mediante Historias de Usuario; las cuales recogen la interacción hablada entre desarrolladores y usuarios. Una vez hechas las Historias de Usuario, el equipo de desarrollo las divide en tareas, estima el esfuerzo, recursos requeridos para su implementación, se genera el plan de entregas, las iteraciones, la rotación de parejas y las reuniones diarias.

Etapa de Diseño: Es la etapa en donde son evaluadas las historias de usuario por el equipo del proyecto para dividir las en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC (Clase-Responsabilidad-Colaborador). Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP.

Etapa de Codificación: Se lleva a cabo la programación en pareja, la unidad de pruebas y la integración del código. Durante esta etapa se espera la disponibilidad del cliente para que éste pueda resolver cualquier duda que se presente durante una jornada de trabajo.

Etapa de Pruebas: Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente:

- Se deben crear las aplicaciones que realizarán los test con un entorno de Desarrollo específico para test.
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de códigos acompañados del código que verifican. Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

2.1: Descripción de la Solución

Se propone desarrollar una aplicación web que permita informatizar el proceso de creación de cartas tecnológicas asociadas al proceso de soldadura al arco por metal revestido, logrando una mayor eficiencia de este proceso y permita un mayor control sobre la selección de los diferentes materiales asociados al proceso, como máquinas de soldadura, aceros y electrodos, permitiendo también la disminución del error humano.

Los usuarios que interactúan con la aplicación poseerán los siguientes roles:

- Administrador del sistema: Usuario con la responsabilidad de configurar y auditar la aplicación, así como gestionar los usuarios internos y asignarles permisos.
- Tecnólogo: Usuario que utiliza la aplicación para crear las cartas tecnológicas del proceso de soldadura.
- Supervisor: Usuario encargado de supervisar las cartas tecnológicas creadas, así como descargarlas e imprimirlas para su posterior almacenamiento físico

2.2: Etapa de Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “Historias de Usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en donde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios” (Beck, 2004).

2.2.1: Equipo de Trabajo

La metodología XP define roles de trabajo asociando a cada uno con diversas actividades.

A continuación, se definen los roles, quedando designado el programador que sería el encargado de producir el código del sistema, el jefe de proyecto y el cliente que no es más que el que escribe las historias de usuario, les asigna la prioridad y diseña las pruebas funcionales para validar su implementación. A continuación, se muestra la asignación de estos roles a las personas responsables.

| Miembro | Roles |
|--------------------------|------------------|
| Mayli Estopiñán Lantigua | Jefe de proyecto |
| Marlon Ortega Méndez | Programador |
| Ramón Quiza Sardiñas | Cliente, Tester |

Tabla 2.2.1: Equipo de trabajo y Roles

2.2.2: Historias de Usuario

Las “Historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen.

Escalas equivalentes a la prioridad en el negocio:

- Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.
- Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.
- Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

- Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.
- Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.
- Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

En la tabla 2.2.2 se colocan las Historias de usuarios definidas

| Nombre | Prioridad | Riesgo | Esfuerzo | Iteración | Entrega |
|--|-----------|--------|----------|-----------|---------|
| 1-Diseño y creación de la Base de Datos | Alta | Alto | 2 | 1 | 1 |
| 2-Diseño de la interfaz de Usuario | Media | Medio | 1 | 1 | 1 |
| 3-Authenticación de Usuario | Alta | Medio | 1 | 1 | 1 |
| 4-Gestionar Usuario | Alta | Alto | 2 | 1 | 1 |
| 5-Gestionar materiales de la soldadura | Media | Medio | 1 | 2 | 2 |
| 6-Gestionar máquinas de soldadura | Media | Medio | 1 | 2 | 2 |
| 7-Gestionar electrodos para aceros inoxidables | Media | Medio | 1 | 2 | 2 |
| 8-Gestionar electrodos para aceros al carbono y bajo aleados | Media | Medio | 2 | 2 | 2 |
| 9-Gestionar Cartas Tecnológicas | Alta | Alto | 2 | 3 | 3 |
| 10-Graficar preparación de bordes | Media | Medio | 3 | 3 | 3 |
| 11-Graficar geometría de la soldadura | Media | Medio | 3 | 3 | 3 |
| 12-calcular parámetros del régimen de soldadura | Media | Alto | 1 | 3 | 3 |
| 13-Calcular Costo de soldadura | Media | Alto | 1 | 4 | 4 |
| 14-Exportar Carta Tecnológica | Media | Bajo | 1 | 4 | 4 |
| 15-Generar Reportes | Alta | Medio | 2 | 4 | 4 |

Tabla 2.2.2: Planificación de Historias de Usuario

2.2.3: Plan de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación.

El proyecto fue dividido en cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Además de también tomar en cuenta el riesgo y prioridad de cada historia de usuario

Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades. En la siguiente figura se muestra el plan de iteraciones y la cantidad de semanas que tomo cada iteración

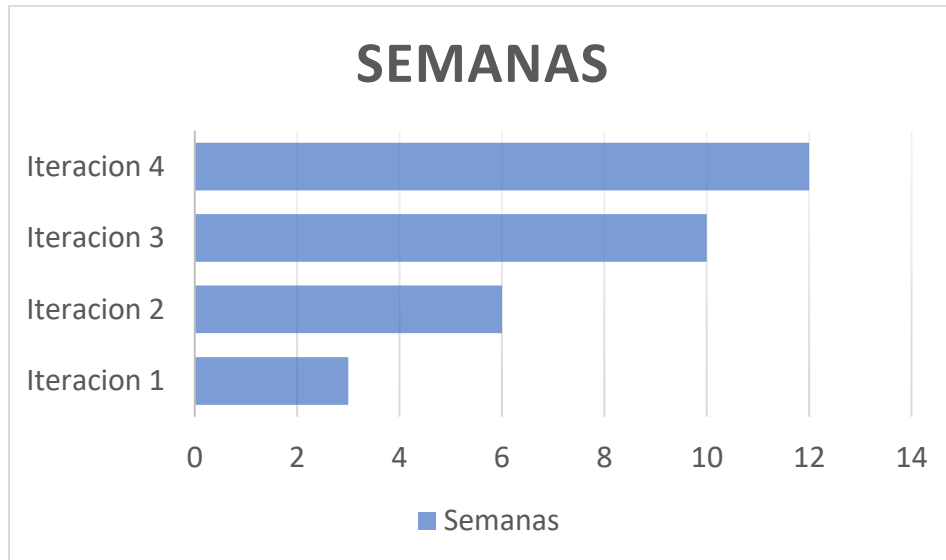


Figura 2.2.3: Plan de iteraciones

2.2.4: Plan de Entregas

El plan de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

| Entrega | Historias a entregar |
|---------|---------------------------------------|
| 1 | Diseño y creación de la Base de Datos |
| | Diseño de la interfaz de Usuario |
| | Autenticación de Usuario |
| | Gestionar Usuario |
| 2 | Gestionar materiales de la soldadura |
| | Gestionar máquinas de soldadura |

| | |
|---|--|
| 3 | Gestionar electrodos para aceros inoxidables |
| | Gestionar electrodos para aceros al carbono y bajo aleados |
| | Gestionar Cartas Tecnológicas |
| | Graficar preparación de bordes |
| 4 | Graficar geometría de la soldadura |
| | calcular parámetros del régimen de soldadura |
| | Calcular Costo de soldadura |
| | Exportar Carta Tecnológica |
| | Generar Reportes |

Tabla 2.2.4: Plan de Entregas

2.2.5: Análisis de Costos

a estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto. Es necesario tener en cuenta numerosos aspectos que afectan a la estimación como la complejidad del proyecto, su estructuración, el tamaño, los recursos involucrados y los riesgos asociados (Pressman, 2010).

La estimación del costo de un software es el proceso de predecir la cantidad de esfuerzo requerido para el desarrollo del sistema y el tiempo para ello. Existen diversos modelos para realizar la estimación del costo de un software como, por ejemplo: COCOMO I, COCOMO II, Puntos de Función, Botton – Up, Top – Down, entre otros.

2.2.5.1: Método de estimación por Puntos de Función

Es una métrica que permite traducir en un número el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto (Tach, 2014).

Componentes:

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

Una vez obtenidos los diferentes elementos del sistema se utilizan las tablas reflejadas en el Anexo X para asignar pesos en función del número de atributos que tengan y el número de archivos a los que afecte (Suardiyana Putra et al., 2012).

| Componentes | Bajo | Medio | Alto | Total |
|-------------|------|-------|------|-------|
| EI | 3*21 | 4*0 | 6*0 | 63 |
| EO | 4*5 | 5*1 | 7*0 | 25 |
| EQ | 3*1 | 4*2 | 6*0 | 11 |
| ILF | 7*0 | 10*1 | 15*0 | 10 |
| EIF | 5*0 | 7*0 | 10*0 | 0 |
| Total | | | | 106 |

Tabla 2.2.5.1: Pesos de los componentes

Cálculo de los Puntos de Función sin Ajustar (PFSA)

Los PFSA se calculan como la suma de los productos de cada componente por su peso determinado en la tabla correspondiente.

$$PFSA = 63+25+11+10+0=106$$

Ajuste de Complejidad Técnica (ACT)

Para calcular el ACT se le va dando un valor entre 0 y 5 a cada Factor de Ajuste como se muestra en el Anexo 4. Cuando cada Factor tenga un valor, se suman todos y así obtenemos el ACT.

Cálculo de los Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0.65 + (0.01 * ACT)]$$

$$PFA = 106 * [0.65 + (0.01 * 29)] = 99.64$$

Esfuerzo en horas / persona (E)

$$E = PFA / (1 \text{ Persona} / 8 \text{ horas})$$

$$E = 99.64 / (0.125) = 797.12 \text{ horas}$$

Tomando 26 días laborables en el mes y 8 horas productivas al día, se obtiene 208 horas laborables al mes.

Duración del mes (DM) = E / (208 horas por mes)

Duración del Mes (DM) = 797.12 / 208 = 4 meses

Cálculo del Presupuesto del Proyecto

Costo total del Proyecto = Sueldo de 1 participante * Cantidad de participantes * DM

Costo total del Proyecto = 3600 * 1 * 4 = 14400

Realizando un análisis de costos y beneficios respecto a lo descrito en este trabajo se pude llegar a la conclusión que los aspectos positivos superan a los costos. Debe destacarse que el costo de desarrollo de este software resulto gratuito dado que es el resultado del trabajo de diploma del autor

2.3: Etapa de Diseño

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y reutilización del código. Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema (Pressman, 2010).

2.3.1: Prototipo de Interfaz de usuario

Para la creación de Interfaces de usuario la metodología Extreme Programming(XP) recurre a la creación de Soluciones Pico-Prototipo que consisten en diseñar los prototipos de interfaces lo más ajustado posible al diseño final de las mismas interfaces. A continuación, se muestran imágenes de las interfaces de usuario de las figuras 2.3.1.1 a la figura 2.3.1.3



Fig 2.3.1.1: Interfaz de página principal

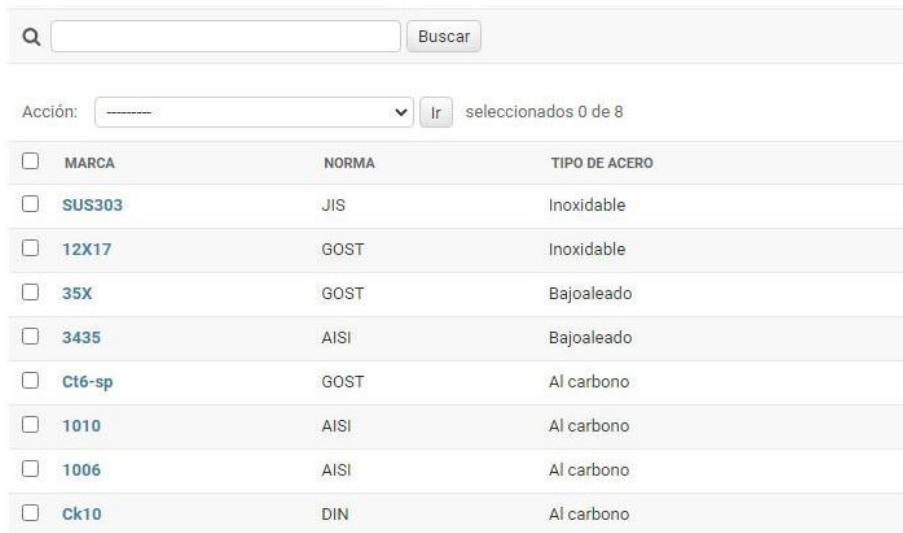


Fig 2.3.1.2: Interfaz de listar Material Base

Llenado de cartas tecnológicas

- Cajetín de usuario
- Geometría de unión
- Metal base
- Máquina de soldadura
- Preparación de bordes
- Selección de electrodo
- Cálculo de proceso
- Equipamiento auxiliar
- Requerimientos tecnológicos

| DATOS | | ELECTRODOS | | | |
|---------------------------|-----------------------|--|-----------------------------|--------------------|--------------------|
| TIPO DE UNIÓN | GEOMETRÍA DE LA UNIÓN | MARCA | PASADAS DE RAZ | PASADAS DE RELLENO | PASADAS DE SELLADO |
| POSICIÓN | DIMENSIONES | FABRICANTE | | | |
| ESPESOR DE LA PLANCHILLA | ESPESOR DE LA COSTURA | CÓDIGO AWS | DIÁMETRO | | |
| | | | | | |
| MATERIAL BASE | | RÉGIMEN DE SOLDADURA | | | |
| MARCA | | CANTIDAD DE PASADAS | DE RAZ | DE RELLENO | DE SELLADO |
| TIPO DE MATER. | MÁQUINA DE SOLDADURA | INTENSIDAD DE CORRIENTE | | | TOTAL |
| | | POLARIDAD | | | UNIDAD |
| MARCA MODELO | | CONSUMO DE ELECTRODO | | | |
| RANCO CORRI. | | CONSUMO DE ELECTRICIDAD | | | kg |
| TIPO DE CORR. | | TIEMPO TECNOLÓGICO | | | min |
| COSTURA | | TIEMPO AUXILIAR | - | - | min |
| PREPARACIÓN DE BORDES | | TIEMPO DE SERVICIO | - | - | min |
| CÓDIGO | NORMA | TIEMPO DE DESCANSO | - | - | min |
| | CRÓQUIS | TIEMPO TOTAL | - | - | min |
| | | PRECALENTAMIENTO | REQUERIMIENTOS TECNOLÓGICOS | | |
| | | TEMPERATURA | °C | | |
| | | TRATAMIENTO TÉRMICO POSTERIOR | | | |
| | | TEMPERATURA | °C | | |
| | | TIEMPO DE PERMANENCIA | min | | |
| | | COSTOS | | | |
| | | COSTO DE ELECTRODO | CLP | | |
| | | COSTO DE MANO DE OBRA | CLP | | |
| | | COSTO DE ELECTRICIDAD | CLP | | |
| | | COSTO TOTAL | CLP | | |
| DIMENSIONES DE LA COSTURA | | EQUIPAMIENTO AUXILIAR | | | |
| | | PORTAELECTRODO | | | |
| | | CABLE | | | |
| | | FILTRO VISUAL | | | |
| | | PROCESO | | | |
| | | SOLDADURA MANUAL CON ELECTRODO REVESTIDO | | | |
| | | NOMBRE | PRIMA | FECHA | |

Fig 2.3.1.3 Interfaz de llenado de Cartas Tecnológicas

2.3.2: Listado de Historias de Usuario

A continuación, para que se pueda comprender el proceso de desarrollo del software, se muestran en detalle algunas de las historias de usuarios que se consideran fundamentales para lograr el objetivo esperado

| Historia de Usuario | | |
|---|-------------------------------|------------------------------|
| Número: 1 | Usuario: Administrador | |
| Nombre: Diseño y creación de la Base de datos | | |
| Prioridad: Alta | Riesgo: Alto | Iteración Asignada: 1 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Se diseña e implementa la base de datos usando SQLite3 y Django | | |
| Observaciones: Usar enfoque Code First para mejorar integración con Django | | |

Tabla 2.3.2.1 Historia de usuario: Diseño y creación de la Base de datos

| Historia de Usuario | | |
|--|-------------------------------|------------------------------|
| Número:2 | Usuario: Administrador | |
| Nombre: Diseño de la interfaz de Usuario | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 1 |
| Programador Responsable: Marlon Ortega Méndez | | |

| |
|--|
| Descripción: Se diseñan las interfaces de la aplicación |
| Observaciones: A petición del cliente usar el color Pantone 163C para crear una interfaz más amigable |

Tabla 2.3.2.2 Historia de usuario: Diseño de la interfaz de Usuario

| | | |
|---|---|------------------------------|
| Historia de Usuario | | |
| Número: 3 | Usuario: Administrador, Tecnólogo , Supervisor | |
| Nombre: Autenticar usuario | | |
| Prioridad: Alta | Riesgo: Medio | Iteración Asignada: 1 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: El usuario introduce sus credenciales, de ser correctas , se le muestra el entorno y funcionalidades de acuerdo a su rol , de lo contrario , se muestra un mensaje de error | | |
| Observaciones: El administrador será el encargado de añadir nuevos usuarios a la aplicación | | |

Tabla 2.3.2.3 Historia de usuario: Autenticar usuario

| | | |
|---|-------------------------------|------------------------------|
| Historia de Usuario | | |
| Número: 4 | Usuario: Administrador | |
| Nombre: Gestionar usuarios | | |
| Prioridad: Alta | Riesgo: Alto | Iteración Asignada: 1 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Gestionar Los nuevos usuarios que acceden a los servicios de la aplicación, así como sus roles y permisos | | |
| Observaciones: El administrador será el encargado de asignar los roles de cada usuario de la aplicación | | |

Tabla 2.3.2.4 Historia de usuario: Gestionar usuarios

| | | |
|--|-------------------------------|------------------------------|
| Historia de Usuario | | |
| Número: 5 | Usuario: Administrador | |
| Nombre: Gestionar materiales de soldadura | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 2 |

| |
|---|
| Programador Responsable: Marlon Ortega Méndez |
| Descripción: Gestionar los materiales que se utilizan en el proceso de soldadura al arco por metal revestido |
| Observaciones: el rol de administrador será el encargado de añadir o modificar los nuevos materiales agregados |

Tabla 2.3.2.5 Historia de usuario: Gestionar materiales de soldadura

| Historia de Usuario | | |
|--|-------------------------------|------------------------------|
| Número: 6 | Usuario: Administrador | |
| Nombre: Gestionar máquinas de soldadura | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 2 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Gestionar las Máquinas de soldadura que se utilizan en el proceso de soldadura al arco por metal revestido | | |
| Observaciones: El rol de administrador será el encargado de añadir o modificar las nuevas máquinas agregadas | | |

Tabla 2.3.2.6 Historia de usuario: Gestionar máquinas de soldadura

| Historia de Usuario | | |
|---|-------------------------------|------------------------------|
| Número: 7 | Usuario: Administrador | |
| Nombre: Gestionar electrodos para aceros inoxidables | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 2 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Gestionar los electrodos para aceros inoxidables que se utilizan en el proceso de soldadura al arco por metal revestido | | |
| Observaciones: El rol de administrador será el encargado de añadir o modificar los nuevos electrodos para aceros inoxidables agregados | | |

Tabla 2.3.2.7 Historia de usuario: Gestionar electrodos para aceros inoxidables

| Historia de Usuario | |
|----------------------------|-------------------------------|
| Número: 8 | Usuario: Administrador |

| | | |
|--|----------------------|------------------------------|
| Nombre: Gestionar electrodos para aceros al carbono y bajo aleados | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 2 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Gestionar los electrodos para aceros al carbono y bajoaleados que se utilizan en el proceso de soldadura al arco por metal revestido | | |
| Observaciones: El rol de administrador será el encargado de añadir o modificar los nuevos electrodos para aceros al carbono y bajoaleados agregados | | |

Tabla 2.3.2.8 Historia de usuario: Gestionar electrodos para aceros al carbono y bajo aleados

| | | |
|--|---------------------------|------------------------------|
| Historia de Usuario | | |
| Número: 9 | Usuario: Tecnólogo | |
| Nombre: Gestionar Cartas Tecnológicas | | |
| Prioridad: Alta | Riesgo: Alto | Iteración Asignada: 3 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el tecnólogo accede a la aplicación correctamente usando sus credenciales y procede a llenar los formularios correspondientes para añadir o modificar la carta tecnológica en la que trabaja | | |
| Observaciones: El tecnólogo puede crear y modificar nuevas cartas tecnológicas, pero solo el administrador puede borrarlas | | |

Tabla 2.3.2.9 Historia de usuario: Gestionar Cartas Tecnológicas

| | | |
|--|---------------------------|------------------------------|
| Historia de Usuario | | |
| Número: 10 | Usuario: Tecnólogo | |
| Nombre: Graficar preparación de bordes | | |
| Prioridad: Media | Riesgo: Medio | Iteración Asignada: 3 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el tecnólogo accede a la funcionalidad de preparación de bordes que le permitirá generar un gráfico del apartado preparación de bordes en la carta tecnológica | | |
| Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación | | |

Tabla 2.3.2.10 Historia de usuario: Graficar preparación de bordes

| Historia de Usuario | | |
|--|---------------------------|------------------------------|
| Número: 11 | Usuario: Tecnólogo | |
| Nombre: Graficar geometría de la soldadura | | |
| Prioridad: Media | Riesgo: Media | Iteración Asignada: 3 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el tecnólogo accede a la funcionalidad de geometría de la soldadura que le permitirá generar un gráfico del apartado geometría de la soldadura en la carta tecnológica | | |
| Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación | | |

Tabla 2.3.2.11 Historia de usuario: Graficar geometría de la soldadura

| Historia de Usuario | | |
|---|---------------------------|------------------------------|
| Número: 12 | Usuario: Tecnólogo | |
| Nombre: calcular parámetros del régimen de soldadura | | |
| Prioridad: Media | Riesgo: Alto | Iteración Asignada: 3 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el tecnólogo accede a la funcionalidad de cálculos de Proceso que permite llenar los apartados de tiempo tecnológico, Pre calentamiento y Tratamiento térmico posterior de la carta tecnológica | | |
| Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación | | |

Tabla 2.3.2.12 Historia de usuario: calcular parámetros del régimen de soldadura

| Historia de Usuario | | |
|---|---------------------------|------------------------------|
| Número: 13 | Usuario: Tecnólogo | |
| Nombre: Calcular costos de soldadura | | |
| Prioridad: Media | Riesgo: Alto | Iteración Asignada: 4 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el tecnólogo accede a la funcionalidad de cálculos de Proceso que permite llenar los apartados de costo de la soldadura de la carta tecnológica | | |

Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación

Tabla 2.3.2.13 Historia de usuario: Calcular costos de soldadura

| Historia de Usuario | | |
|---|----------------------------|------------------------------|
| Número: 14 | Usuario: Supervisor | |
| Nombre: Exportar Carta tecnológica | | |
| Prioridad: Media | Riesgo: Bajo | Iteración Asignada: 4 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el supervisor accede a la funcionalidad de Exportar cartas tecnológicas que le permite exportar una imagen de la carta tecnológica en formato .JPEG | | |
| Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación | | |

Tabla 2.3.2.14 Historia de usuario: Exportar Carta tecnológica

| Historia de Usuario | | |
|--|----------------------------|------------------------------|
| Número: 15 | Usuario: Supervisor | |
| Nombre: Generar Reportes | | |
| Prioridad: Alta | Riesgo: Medio | Iteración Asignada: 4 |
| Programador Responsable: Marlon Ortega Méndez | | |
| Descripción: Inicia cuando el supervisor accede a la funcionalidad de ver reportes, los reportes se mostrarán por fechas | | |
| Observaciones: La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación | | |

Tabla 2.3.2.15 Historia de usuario: Generar Reportes

2.3.3: Listado de Tareas de Ingeniería

Es importante descomponer cada historia de usuario en tareas de ingeniería. A continuación, se muestra un listado con las tareas de ingeniería por cada historia de usuario

| No | Nombre de HU | Tarea de Ingeniería |
|-----------|--|--|
| 1 | Diseño y creación de la Base de Datos | 1-Diseñar base de datos 2-Crear la base de datos |
| 2 | Diseño de la interfaz de Usuario | 3- Diseñar interfaz de usuario |
| 3 | Autenticación de Usuario | 4- Autenticar usuario existente |
| 4 | Gestionar Usuario | 5-Listar usuario 6-Filtrar usuario 7-Crear usuario 8-Editar usuario 9-Eliminar usuario |
| 5 | Gestionar materiales de la soldadura | 10-Listar materiales de soldadura 11-Filtrar materiales de soldadura 12-Crear materiales de soldadura 13-Editar materiales de soldadura 14-Eliminar materiales de soldadura |
| 6 | Gestionar máquinas de soldadura | 15-Listar máquinas de soldadura 16-Filtrar máquinas de soldadura 17-Crear máquinas de soldadura 18-Editar máquinas de soldadura 19-Eliminar máquinas de soldadura |
| 7 | Gestionar electrodos para aceros inoxidables | 20-Listar electrodos para aceros inoxidables 21-Filtrar electrodos para aceros inoxidables 22-Crear electrodos para aceros inoxidables 23-Editar electrodos para aceros inoxidables 24-Eliminar electrodos para aceros inoxidables |
| 8 | Gestionar electrodos para aceros al carbono y bajo aleados | 25-Listar electrodos para aceros al carbono y bajo aleados 26-Filtrar electrodos para aceros al carbono y bajo aleados 27-Crear electrodos para aceros al carbono y bajo aleados 28-Editar electrodos para aceros al carbono y bajo aleados 29-Eliminar electrodos para aceros al carbono y bajo aleados |
| 9 | Gestionar Cartas Tecnológicas | 30-Crear Cartas Tecnológicas 31-Listar Cartas Tecnológicas 32-Eliminar Cartas Tecnológicas |

| | | |
|----|--|--|
| | | 33-Filtrar Cartas Tecnológicas 34-Editar Cartas Tecnológicas |
| 10 | Graficar preparación de bordes | 35-Graficar preparación de bordes |
| 11 | Graficar geometría de la soldadura | 36-Graficar geometría de la soldadura |
| 12 | calcular parámetros del régimen de soldadura | 37-Calcular tiempo tecnológico 38-Calcular tiempo de servicio 39-Calcular tiempo de descanso 40-Calcular tiempo total |
| 13 | Calcular Costo de soldadura | 41-Calcular costo de electrodos 42-Calcular costo de mano de obra 43-Calcular costo de electricidad 44-Calcular costo total |
| 14 | Exportar Carta Tecnológica | 45-Exportar Carta tecnológica |
| 15 | Generar Reportes | 46-Reporte de costo total 47-Reporte de costo por fechas 49-Reporte de consumo total 50-Reporte de consumo por fechas |

Tabla 2.3.3.1: Listado de tareas de ingeniería

A continuación, se muestran algunas de las tareas de ingeniería fundamentales para el desarrollo de esta investigación.

| Tarea de Ingeniería | |
|--|---|
| Numero de tarea: 1 | Numero de historia de usuario: 1 |
| Nombre: Diseñar la base de datos | |
| Tipo de tarea: Diseño | Puntos estimados: 0.5 |
| Programador responsable: Marlon Ortega Méndez | |
| Descripción: Analizar el negocio en profundidad para diseñar una base de datos que permita almacenar la información necesaria | |

Tabla 2.3.3.2: Tarea de ingeniería: Diseñar la base de datos

| Tarea de Ingeniería | |
|--|---|
| Numero de tarea: 2 | Numero de historia de usuario: 1 |
| Nombre: Crear la base de datos | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Programador responsable: Marlon Ortega Méndez | |
| Descripción: Crear la base de datos con sus respectivas relaciones e integridad entre las deferentes tablas | |

Tabla 2.3.3.3: Tarea de ingeniería: Crear la base de datos

| Tarea de Ingeniería | |
|--|---|
| Numero de tarea: 12 | Numero de historia de usuario: 5 |
| Nombre: Crear materiales de soldadura | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.4 |
| Programador responsable: Marlon Ortega Méndez | |
| Descripción: Se mostrará un formulario que permita al usuario con los permisos pertinentes añadir un nuevo material de soldadura a la base de datos | |

Tabla 2.3.3.4: Tarea de ingeniería: Crear materiales de soldadura

| Tarea de Ingeniería | |
|--|--|
| Numero de tarea: 11 | Numero de historia de usuario: 36 |
| Nombre: Graficar geometría de la soldadura | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.7 |
| Programador responsable: Marlon Ortega Méndez | |
| Descripción: Se mostrará un formulario que permita al usuario con los permisos pertinentes seleccionar los datos necesarios para generar el grafico de geometría de soldadura en la carta tecnológica | |

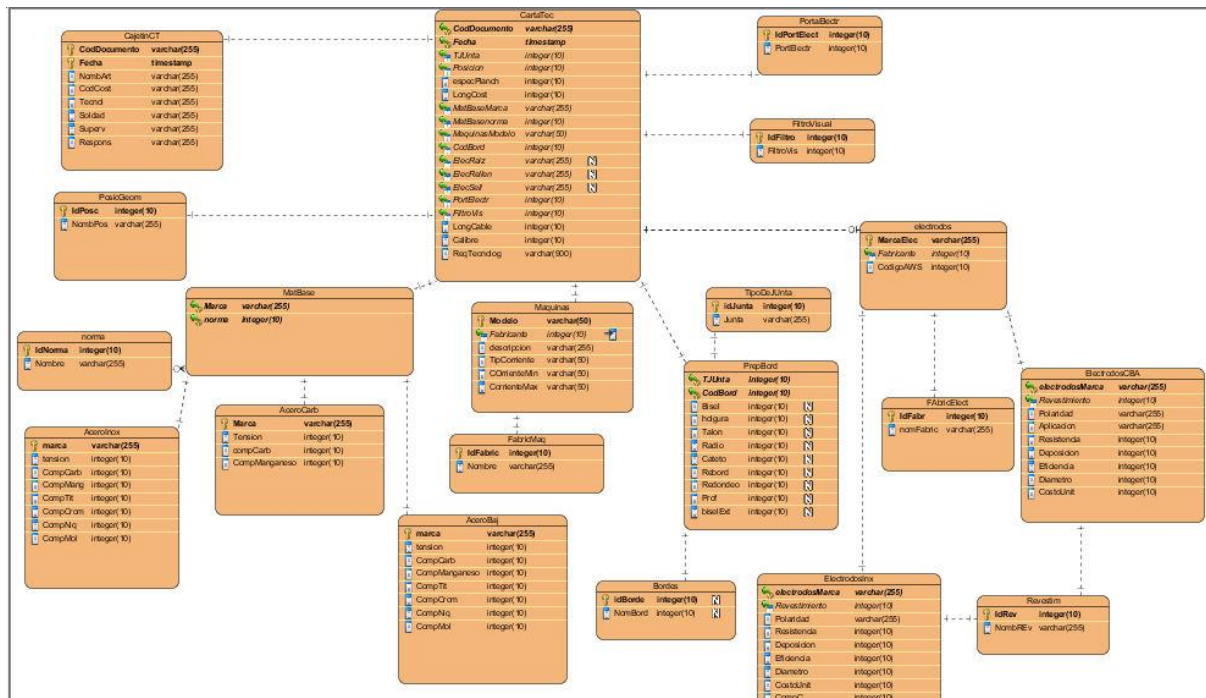
Tabla 2.3.3.5: Tarea de ingeniería: Graficar geometría de la soldadura

| Tarea de Ingeniería | |
|---|--|
| Numero de tarea: 36 | Numero de historia de usuario: 12 |
| Nombre: Calcular tiempo tecnológico | |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.6 |
| Programador responsable: Marlon Ortega Méndez | |
| Descripción: Se mostrará un formulario que permita al usuario con los permisos pertinentes introducir los datos necesarios para calcular el tiempo tecnológico de la carta tecnológica | |

Tabla 2.3.3.5: Tarea de ingeniería: Calcular tiempo tecnológico

2.3.4: Modelo de la base de datos

En la siguiente ilustración se muestra el modelo de la base de datos de la aplicación desarrollada



2.4: Pruebas

El proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar (IBM, 2023). En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Sistema.
- Prueba de Aceptación.
- Prueba de Seguridad.

2.4.1: Plan de pruebas

El plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a

través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento (Reyes Chirino, 2021).

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está entregando al cliente

| No | Nombre de HU | Prueba a realizar |
|-----------|---|--|
| 1 | Diseño y creación de la Base de Datos | Test de base de datos |
| 2 | Diseño de la interfaz de Usuario | Test de diseño de interfaz de usuario |
| 3 | Autenticación de Usuario | Test de autenticación de usuario |
| 4 | Gestionar Usuario | Test de listar usuario Test de filtrar usuario Test de crear usuario Test de editar usuario Test de eliminar usuario |
| 5 | Gestionar materiales de la soldadura | Test de listar materiales de soldadura Test de filtrar materiales de soldadura Test de crear materiales de soldadura Test de editar materiales de soldadura Test de eliminar materiales de soldadura |
| 6 | Gestionar máquinas de soldadura | Test de listar máquinas de soldadura Test de filtrar máquinas de soldadura Test de crear máquinas de soldadura Test de editar máquinas de soldadura Test de eliminar máquinas de soldadura |
| 7 | Gestionar electrodos para aceros inoxidable | Test de listar electrodos para aceros inoxidable Test de filtrar electrodos para aceros inoxidable Test de crear electrodos para aceros inoxidable Test de editar electrodos para |

| | | |
|-----------|--|---|
| | | aceros inoxidables Test de eliminar electrodos para aceros inoxidables |
| 8 | Gestionar electrodos para aceros al carbono y bajo aleados | Test de listar electrodos para aceros al carbono y bajo aleados Test de filtrar electrodos para aceros al carbono y bajo aleados Test de crear electrodos para aceros al carbono y bajo aleados Test de editar electrodos para aceros al carbono y bajo aleados Test de eliminar electrodos para aceros al carbono y bajo aleados |
| 9 | Gestionar Cartas Tecnológicas | Test de crear Cartas Tecnológicas Test de listar Cartas Tecnológicas Test de eliminar Cartas Tecnológicas Test de filtrar Cartas Tecnológicas Test de editar Cartas Tecnológicas |
| 10 | Graficar preparación de bordes | Test de graficar preparación de bordes |
| 11 | Graficar geometría de la soldadura | Test de graficar geometría de la soldadura |
| 12 | calcular parámetros del régimen de soldadura | Test de calcular tiempo tecnológico Test de calcular tiempo de servicio Test de calcular tiempo de descanso Test de calcular tiempo total |
| 13 | Calcular Costo de soldadura | Test de calcular costo de electrodos Test de calcular costo de mano de obra Test de calcular costo de electricidad Test de calcular costo total |
| 14 | Exportar Carta Tecnológica | Test de exportar Carta tecnológica |
| 15 | Generar Reportes | Test de reporte de costo total Test de reporte de costo por fechas Test de reporte de consumo total |

| | |
|--|---------------------------------------|
| | Test de reporte de consumo por fechas |
|--|---------------------------------------|

Tabla 2.4.1.1: Plan de pruebas

2.4.1: Pruebas de Aceptación

Cuando se construye software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema (Pressman, 2010).

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

| Caso de prueba | |
|---|------------------------------|
| Número: 1 | Historia de Usuario:1 |
| Nombre del caso de pruebas: Test de base de datos | |
| Descripción: Verificar el correcto funcionamiento de la base de datos | |
| Condición de ejecución: estar conectado a la base de datos | |
| Entrada: Valores para leer o escribir en la base de datos , por ejemplo Usuario y Contraseña | |
| Resultado esperado: Se muestran o guardan los datos correctamente | |
| Evaluación: Prueba satisfactoria | |

Tabla 2.4.1.1: Caso de prueba: test de base de datos

| Caso de prueba | |
|---|-------------------------------|
| Número: 3 | Historia de Usuario: 3 |
| Nombre del caso de pruebas: Test de autenticación de usuario | |
| Descripción: Se inserta el nombre de usuario y contraseña para entrar al sistema. Se insertarán de forma incorrecta, dejando campos en blanco para | |

| |
|---|
| verificar la validación. Luego se insertarán los datos de manera correcta para comprobar esta funcionalidad |
| Condición de ejecución: máquinas de soldadura |
| Entrada: Nombre de usuario y contraseña |
| Resultado esperado: El sistema notifica al usuario cuando se inserten datos erróneos o cuando se dejen campos en blanco. Cuando se inserten los datos correctamente, según el rol que tenga el usuario, se le otorgan los permisos correspondientes. |
| Evaluación: Prueba satisfactoria |

Tabla 2.4.1.2 Caso de prueba test de autenticación de usuario

| Caso de prueba | |
|--|-------------------------------|
| Número: 6 | Historia de Usuario: 6 |
| Nombre del caso de pruebas: Test de gestionar máquinas de soldadura | |
| Descripción: Se inserta los datos para crear una nueva máquina de soldadura, se inserta de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. Se modifican los datos de la máquina de soldadura, se modifican incorrectamente dejando campos en blanco para verificar la validación, se modifican correctamente para que los sean almacenados. Se elimina la máquina de soldadura aceptando el mensaje de confirmación. | |
| Condición de ejecución: máquinas de soldadura | |
| Entrada: Datos de la máquina de soldadura | |
| Resultado esperado: El sistema debe alertar cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. | |
| Evaluación: Prueba satisfactoria | |

Tabla 2.4.1.3Caso de prueba test de gestionar máquinas de soldadura

En la siguiente tabla se muestran las clases de equivalencia de las pruebas de aceptación a realizar al sistema de la HU

| Condición de entrada | Clases valida | representante | Clases invalidas | Representantes |
|-----------------------------|--|----------------------|-------------------------|-----------------------|
| Modelo | 1-Cualquier cadena de caracteres alfanuméricos, permite espacios | Master 2850 | 2-cadena vacía | NULL |
| Fabricante | 3-cualquier cadena de | Kemmpi | 4-cadena vacía | NULL |

| | | | | |
|-------------------|--|-------------------|----------------------------------|-------|
| | caracteres alfabéticos, no admite espacios | | | |
| Descripción | 5-Cualquier cadena de caracteres alfanuméricos, permite espacios y caracteres especiales | Transformador(3F) | 6-cadena vacía | NULL |
| Tipo de corriente | 7-Seleccionar una de las opciones | Directa | 8-No seleccionar ninguna opción | NULL |
| Corriente mínima | 9-Cualquier número positivo y entero | 5 | 10-número negativo | -15 |
| | | | 11-número negativo con decimales | -12.6 |
| | | | 12-Numero positivo con decimales | 9.4 |
| | | | 13-Cadena con letras | 12A |
| | | | 14-Cadena vacía | NULL |
| Corriente máxima | 15-Cualquier número positivo y entero | 285 | 16-número negativo | -100 |
| | | | 17-número negativo con decimales | -45.9 |
| | | | 18-Numero positivo con decimales | 250.8 |
| | | | 19-Cadena con letras | 250A |
| | | | 20-Cadena vacía | NULL |

Tabla 2.4.1.4: Clases de equivalencia

A continuación, se muestran algunas tablas de prueba por caso de prueba

| Tabla de prueba | |
|------------------------------|---|
| No. Caso de Prueba: | 1 |
| Requerimientos: | La aplicación debe estar en ejecución |
| Objetivo: | Probar la acción de insertar una máquina de soldadura (Para cubrir las clases 1,3,5,7,9,15) |
| Tipo de prueba: | Funcional |
| Hardware: | Sistema de cómputo con procesador Core I7, Memoria RAM de 8GB y disco duro de 2TB |
| Software: | Navegador de Internet Google Chrome v103.0 |
| Personal: | Encargados de pruebas |
| Datos de entrada: | Marca: Master 250, Modelo: kemmpi, Descripción: Transformador (3F), Tipo de corriente: Directa, Corriente mínima: 5, Corriente máxima: 250 |
| Resultados esperados: | Máquina de soldadura insertada correctamente |
| Resultados obtenidos: | SI(X) No() |
| Casos de excepción | |
| Aprobado por: | Mayli Estopiñan Lantigua |
| Cargo: | Jefe del proyecto, Tester |
| Líder: | Marlon Ortega Méndez |

Tabla 2.4.1.5: Caso de prueba No.1

| Tabla de prueba | |
|------------------------------|--|
| No. Caso de Prueba: | 2 |
| Requerimientos: | La aplicación debe estar en ejecución |
| Objetivo: | Probar la acción de insertar una máquina de soldadura (Para cubrir las clases 1,3,6,7,9,15) |
| Tipo de prueba: | Funcional |
| Hardware: | Sistema de cómputo con procesador Core I7, Memoria RAM de 8GB y disco duro de 2TB |
| Software: | Navegador de Internet Google Chrome v103.0 |
| Personal: | Encargados de pruebas |
| Datos de entrada: | Marca: Master 250, Modelo: kemmpi, Descripción: NULL, Tipo de corriente: Directa, Corriente mínima: 5, Corriente máxima: 250 |
| Resultados esperados: | La descripción es requerida |
| Resultados obtenidos: | SI(X) No() |
| Casos de excepción | |
| Aprobado por: | Mayli Estopiñan Lantigua |
| Cargo: | Jefe del proyecto, Tester |
| Líder: | Marlon Ortega Méndez |

Tabla 2.4.1.6: Caso de prueba No.2

| Tabla de prueba | |
|----------------------------|---|
| No. Caso de Prueba: | 3 |

| | |
|------------------------------|--|
| Requerimientos: | La aplicación debe estar en ejecución |
| Objetivo: | Probar la acción de insertar una máquina de soldadura (Para cubrir las clases 1,3,5,8,9,15) |
| Tipo de prueba: | Funcional |
| Hardware: | Sistema de cómputo con procesador Core I7, Memoria RAM de 8GB y disco duro de 2TB |
| Software: | Navegador de Internet Google Chrome v103.0 |
| Personal: | Encargados de pruebas |
| Datos de entrada: | Marca: Master 250, Modelo: kemmpi, Descripción: Transformador (3F), Tipo de corriente: NULL, Corriente mínima: 5, Corriente máxima: 250 |
| Resultados esperados: | Debe seleccionar un tipo de corriente |
| Resultados obtenidos: | SI(X) No() |
| Casos de excepción | |
| Aprobado por: | Mayli Estopiñan Lantigua |
| Cargo: | Jefe del proyecto, Tester |
| Líder: | Marlon Ortega Méndez |

Tabla 2.4.1.7: Caso de prueba No.3

| Tabla de prueba | |
|----------------------------|--|
| No. Caso de Prueba: | 4 |
| Requerimientos: | La aplicación debe estar en ejecución |
| Objetivo: | Probar la acción de insertar una máquina |

| | |
|------------------------------|---|
| | de soldadura (Para cubrir las clases 1,3,5,7,9,19) |
| Tipo de prueba: | Funcional |
| Hardware: | Sistema de cómputo con procesador Core I7, Memoria RAM de 8GB y disco duro de 2TB |
| Software: | Navegador de Internet Google Chrome v103.0 |
| Personal: | Encargados de pruebas |
| Datos de entrada: | Marca: Master 250, Modelo: kemmpi, Descripción: Transformador (3F), Tipo de corriente: Directa, Corriente mínima: 5, Corriente máxima: 250A |
| Resultados esperados: | La corriente máxima no pude contener letras |
| Resultados obtenidos: | SI(X) No() |
| Casos de excepción | |
| Aprobado por: | Mayli Estopiñan Lantigua |
| Cargo: | Jefe del proyecto, Tester |
| Líder: | Marlon Ortega Méndez |

Tabla 2.4.1.8: Caso de prueba No.4

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las pruebas realizadas, se solucionaron varios de ellos que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión. Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa y a medida que fueron avanzando se desplazaron a módulos integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentra en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento

2.5: Conclusiones parciales del capítulo

Entre las principales conclusiones a las que se pudieron llegar en este capítulo está que mediante la metodología XP, se reflejaron las etapas imprescindibles para desarrollar el. Se planteó la composición del equipo de desarrollo, las diferentes historias de usuarios y las respectivas interacciones necesarios para acometer la labor de construcción de la aplicación. Se realizó la planificación del proyecto y análisis de costo de este, aplicando el modelo de Puntos de Función, siendo de \$ 14400. Los aspectos abordados anteriormente permitieron reflejar el diseño, estructura y funcionalidad del software a desarrollar, proporcionando una visión clara del proyecto en su conjunto para que las distintas partes involucradas en el mismo llegasen a un acuerdo. Además, al validar la propuesta de solución al problema científico se logra un software que satisface los términos de tiempo y calidad acordados previamente por las partes.

Capítulo 3: Resultados de la investigación

Como resultado obtenido de la presente investigación se tiene el Software nombrado SparkWeb para la elaboración y gestión de cartas tecnológicas en apoyo al proceso de soldadura al arco por metal revestido

3.1: Producto terminado

A continuación, se muestran algunas vistas de la solución desarrollada.

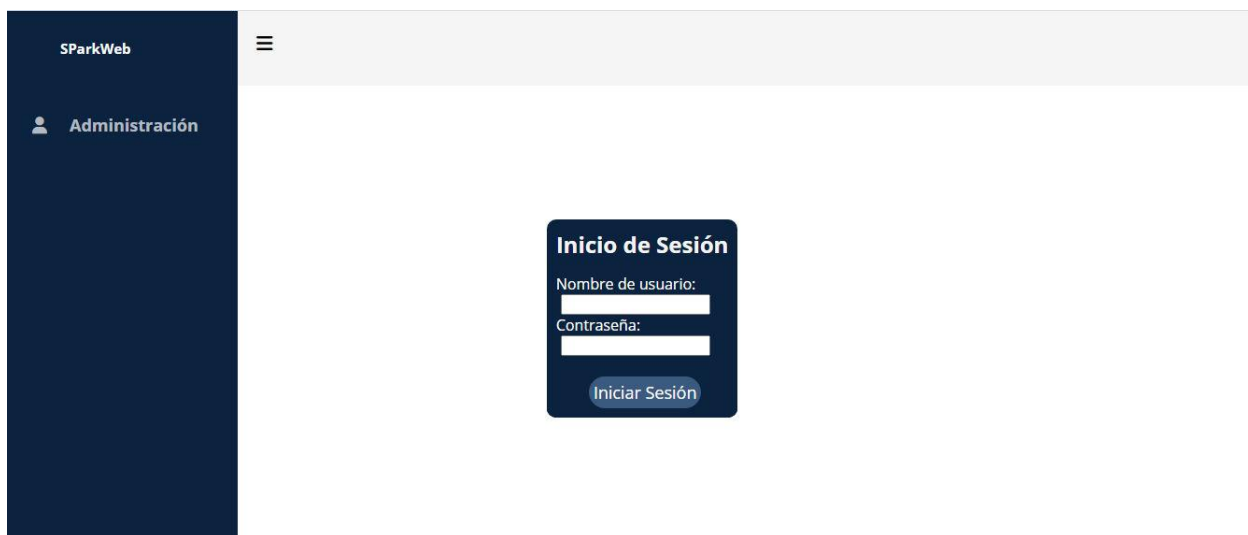


Fig 3.1.1 Página de inicio de sesión Fuente: Elaboración Propia

La primera vista del software que muestra al acceder por primera vez a este consiste en el formulario de inicio de sesión donde el usuario debe insertar su nombre de usuario y su contraseña para acceder a las funciones de la aplicación web. Adicionalmente se tiene

un enlace de acceso a la página de administración del sitio también protegida por su respectivo formulario de inicio de sesión

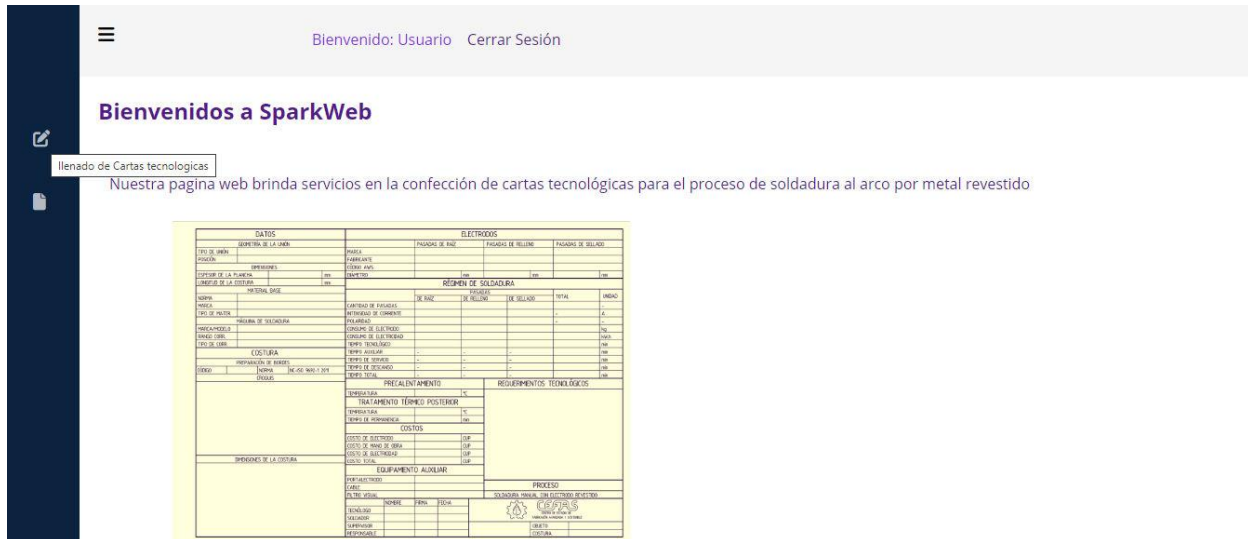


Fig 3.1.2 Página principal de la aplicación Fuente: Elaboración Propia

Una vez el usuario ha insertado sus credenciales correctamente es redirigido a la página principal del sistema donde se da una breve descripción de los servicios disponibles de la aplicación. Así como se le da enlaces a dichos diferentes servicios

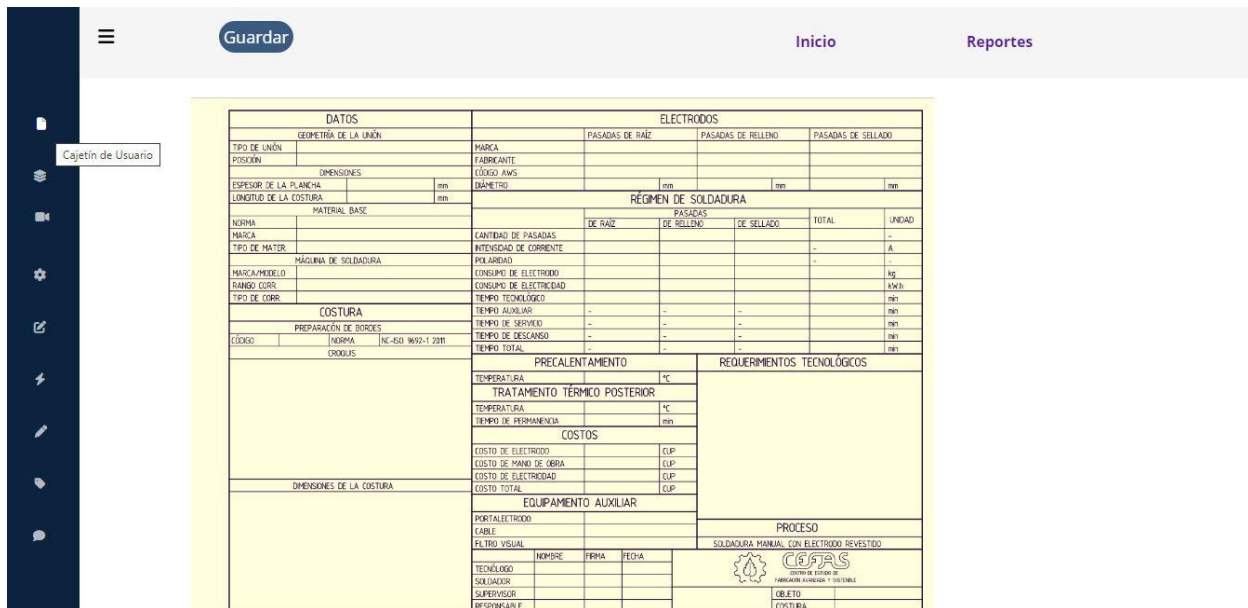


Fig 3.1.3 Página de llenado de cartas tecnológicas Fuente: Elaboración Propia

En esta vista es donde se desarrolla el servicio principal de la aplicación, el llenado de cada aspecto de la carta tecnológica a través del acceso a los diferentes formularios finalizando con el guardado de la carta tecnológica en la base de datos.

Una vista del producto final de la carta tecnológica puede ser vista en el Anexo 5



Fig 3.1.5 Página de Reportes Fuente: Elaboración Propia

En esta vista se ve reflejada la funcionalidad de ver reportes estadísticos referentes a al costo y consumo de materiales asociados al proceso de soldadura reflejado en la carta tecnológica

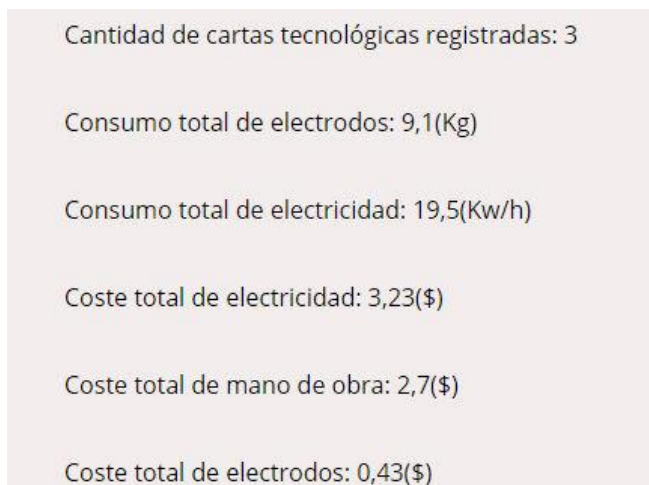


Fig 3.1.6 Reporte estadístico básico de coste y consumo Fuente: Elaboración Propia

Con el desarrollo de este software se ha influido positivamente en el proceso de creación de cartas tecnológicas asociadas al proceso de soldadura el arco por metal revestido al informatizar el proceso se ha mejorado la eficiencia del mismo, ya que los actores que intervienen en este proceso pueden contar con una herramienta de fácil acceso y usabilidad, además de que se pueda acceder a ella desde cualquier dispositivo móvil. Brindándole la posibilidad a los directores de la Empresa donde se ejecuta este proceso un grupo de reportes estadísticos y del trabajo histórico que le facilitan tomar decisiones

de manera más acertada. Cumpliendo con el objetivo principal de la investigación y probando su Hipótesis, se ha demostrado que el software desarrollado durante esta investigación posee la funcionalidad y calidad demandada. Se presenta como una herramienta de fácil manejo que permite contribuir a la eficiencia del proceso de creación de cartas tecnológicas asociadas al proceso de soldadura la arco por metal revestido.

Conclusiones

Luego de realizar un estudio exhaustivo de la problemática abordada en la presente investigación y lográndose dar respuesta a ella de manera satisfactoria, se ha podido llegar a las siguientes conclusiones:

- El estudio realizado sobre los antecedentes, la bibliografía y documentos relacionados con el objeto de estudio, permitió confeccionar un marco teórico referencial que, de conjunto al estudio de las herramientas y tecnologías a utilizar, proporcionó los elementos necesarios para dar solución a la problemática abordada.
- La metodología de desarrollo y herramientas más idóneas para el desarrollo de la propuesta de solución son la metodología de programación extrema XP y los lenguajes HTML, CSS y JavaScript en el lado del cliente y el lenguaje Python con Django como framework en el lado del servidor, además Visual Studio Code como entorno de desarrollo y SQLite3 como Base de datos utilizada.
- Se implementó la aplicación web para la creación de cartas tecnológicas en el proceso de soldadura al arco por metal revestido, siguiendo la metodología de desarrollo de software XP, ajustándose a los requerimientos del usuario y logrando un producto informático que cumple con el objetivo general trazado en la investigación.
- Mediante la aplicación de pruebas a la aplicación web se pudo obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron corregidos como parte del desarrollo de la aplicación

De manera general se concluye que la investigación ha sido satisfactoria al presentar un producto funcional que se ajusta a las necesidades del cliente, cumpliendo con el objetivo principal de la investigación al contribuir a la eficiencia del proceso de creación de cartas tecnológicas para la soldadura al arco por metal revestido

Recomendaciones

Desde el punto de vista del alcance de la presente investigación se proponen las siguientes recomendaciones

1. Continuar la ampliación de las funcionalidades de la aplicación con el enfoque de brindar una mayor personalización de la aplicación
2. Ampliar las funcionalidades con el objetivo de brindar un mayor control y análisis de los datos almacenados, así como una mejor interpretación del flujo de la información

Referencias bibliográficas

AWS, D1.1/D1.1M. 2015. *Structural welding code - Steel.* 2015.

Beck, K., & Andres, C. 2004. *Extreme Programming Explained: Embrace Change, 2nd Edition.* 2004.

BPVC.IX, ASME. 2015. *Boiler and pressure vessel code. Section IX: Qualification standard for welding and brazing procedures, welders, brazers, and welding and brazing operators.* 2015.

Cary, Hb. 1998. *Modern welding technology. 4th Ed.* 1998.

Django software foundation. 2022. about us . www.djangoproject.com. [En línea] 2022.

García, A. 2015. *Revista: Revista de Soldadura y Técnicas Especiales.* 2015.

Granados Delgado, O. 2018. Soldadura con electrodo revestido (SMAW). [En línea] 2018.
<https://repository.usta.edu.co/handle/11634/14740>.

IBM. 2023. Que son las pruebas de software? www.ibm.com. [En línea] 2023.

JUSTEL RAMOS, M.A. 2012. *Soldadura: Manual para operarios y técnicos.* 2012.

Khan, M. S. 2020. *Optimization of Shielded Metal Arc Welding (SMAW) Parameters using Technological Charts.* 2020.

Martínez, R. 2019. "Optimización del proceso de soldadura al arco por metal revestido mediante el uso de cartas tecnológicas informatizadas" *Revista de Investigación en Ciencia e Ingeniería.* 2019.

Menzinsky, A., López, G., & Palacio, J. 2019. *Scrum Manager: Temario Troncal.* 2019.

Microsoft. 2023. index of vscode. code.visualstudio.com. [En línea] 2023.

NC-ISO, 6947. 2004. *Soldaduras - Posiciones de trabajo - Definición de los ángulos de pendiente y de rotación.* 2004.

NC-ISO, 9692-1. 2011. *Soldadura y procesos afines - Recomendaciones para la preparación de la unión. Parte 1: Soldadura por arco con electrodos revestidos, soldadura por arco protegido con gas y electrodo de aporte, soldadura por llama, soldadura por arco con gas inerte .* 2011.

Parmar, Rs. 2017. *Development of Technological Charts for Shielded Metal Arc Welding (SMAW) of Mild Steel.* 2017.

Pedroso, Jorge E Perez. 2016. CICT. *Sitio web de la Universidad de Matanzas.* [En línea] 2016.
[http://cict.umcc.cu/repositorio/tesis/Trabajos%20de%20Diploma/Ingenier%C3%ADa%20Mec%C3%A1nica/2015/Formalizaci%C3%B3n%20y%20algoritmizaci%C3%B3n%20de%20la%20metodolog%C3%ADa%20\(Jorge_E_Perez_Pedroso\).pdf](http://cict.umcc.cu/repositorio/tesis/Trabajos%20de%20Diploma/Ingenier%C3%ADa%20Mec%C3%A1nica/2015/Formalizaci%C3%B3n%20y%20algoritmizaci%C3%B3n%20de%20la%20metodolog%C3%ADa%20(Jorge_E_Perez_Pedroso).pdf).

Pressman, R. S. 2010. *Ingeniería del software: Un enfoque práctico (Séptima edición ed.).* 2010.

Python software foundation. 2022. About us. www.Python.org. [En línea] 2022.

Reyes Chirino, R., & Delgado Fernández, T. 2021. *Modelo funcional de un laboratorio ciudadano de innovación digital. UNIVERSIDAD Y SOCIEDAD | Revista Científica de la Universidad de Cienfuegos.* 2021.

Rodríguez, C. 2020. "Desarrollo de un software para la generación y gestión de cartas tecnológicas en la soldadura al arco por metal revestido" *Revista de Ingeniería y Tecnología.* 2020.

Rosado Gómez, A., Quintero Duarte, A., & Meneses Guevara, C. D. 2012. *Desarrollo ágil de software aplicando programación extrema. Revista Ingenio.* 2012.

Sommerville, I. 2011. *Ingeniería de Software. Pearson Educación de México.* 2011.

SQLite consortium. 2023. what is SQLite? www.sqlite.org. [En línea] 2023.

Suardiyana Putra, I. P. E., Yuliawati, A., & Mursanto, P. 2012. *dustrial extreme programming practice's implementation in rational unified process on agile development theme. Proceedings of the 2012 International Conference on Advanced Computer Science and Information Systems.* 2012.

Syed, H. M. 2018. *Development of Technological Charts for Shielded Metal Arc Welding (SMAW) of Stainless Steel".* 2018.

Tach, Luis Accdpsy. 2014. Puntos de Funcion. www.prezi.com. [En línea] 2014.

W3C. 2021. W3schools. W3schools.com. [En línea] 2021.

Anexos


```

01: INICIO ALGORITMO
02:  ENTRAR: Datos de la unión
03:  ENTRAR: Datos del metal base
04:  ENTRAR: Datos de la máquina de soldadura
05:  SELECCIONAR: Preparación de bordes
06:  DETERMINAR: Esquema de soldadura
07:  PARA Cada tipo de pasada (raíz, relleno y sellado)
08:    SELECCIONAR: Electrodo
09:  FIN PARA
10:  CALCULAR: Número de pasadas
11:  PARA Cada pasada
12:    CALCULAR: Intensidad de la corriente
13:    CALCULAR: Consumo de electrodo
14:    CALCULAR: Consumo de energía eléctrica
15:    CALCULAR: Tiempo tecnológico
16:  FIN PARA
17:  CALCULAR: Temperatura de precalentamiento
18:  CALCULAR: Tratamiento térmico posterior
19:  CALCULAR: Tiempo total
20:  CALCULAR: Costos
21:  SELECCIONAR: Equipamiento auxiliar
22: FIN ALGORITMO
    
```

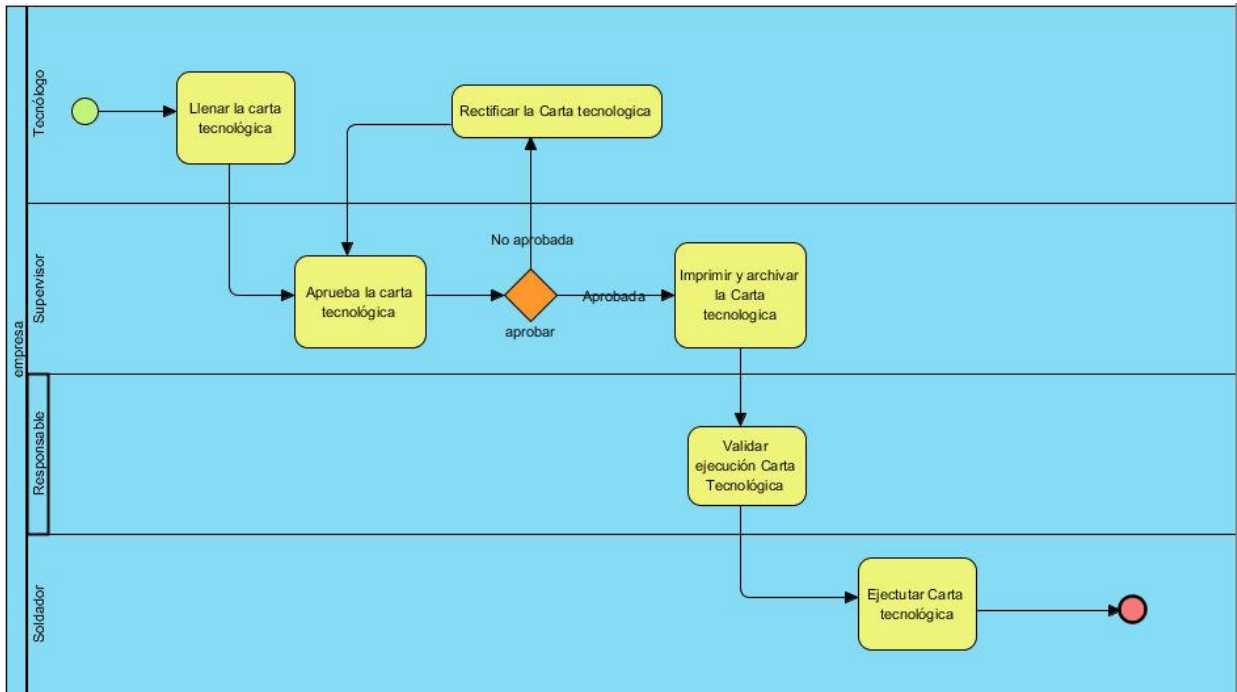
Anexo 1: Algoritmo de creación de cartas tecnológicas en el proceso de soldadura al arco por metal revestido(SMAW) Fuente:

Opifex Spark/SMAW []

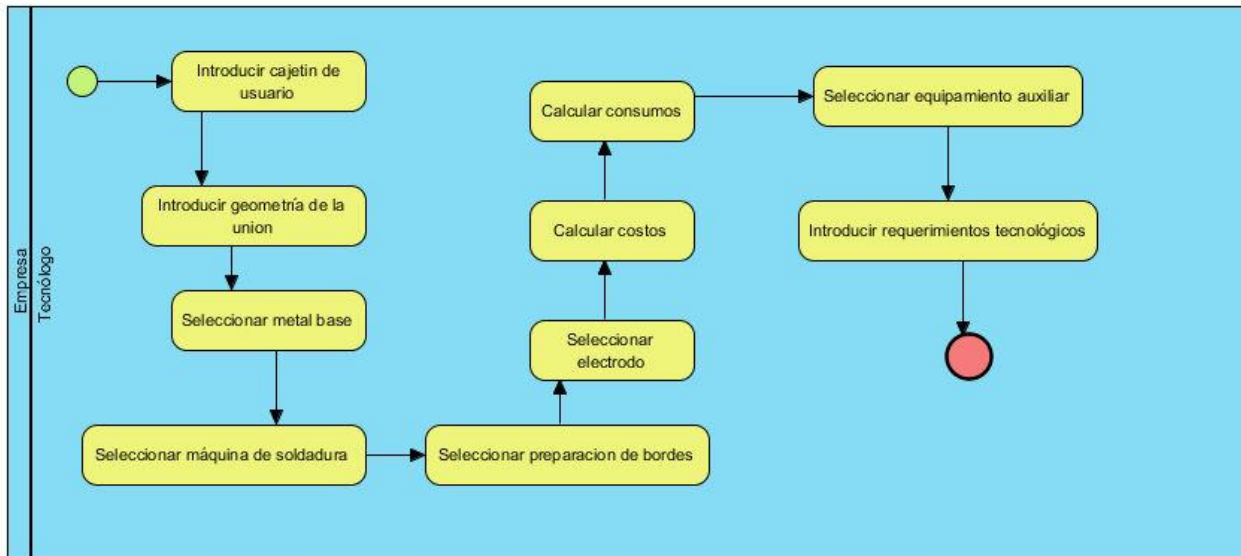
Archivo Datos Tecnología Ayuda

| DATOS | | ELECTRODOS | | |
|---------------------------|--------------------------|---|--------------------|-----------------------------|
| GEOMETRÍA DE LA UNIÓN | | PASADAS DE RAÍZ | PASADAS DE RELLENO | PASADAS DE SELLADO |
| TIPO DE UNIÓN | | | | |
| POSICIÓN | | | | |
| DIMENSIONES | | | | |
| ESPESOR DE LA PLANCHILLA | mm | DIÁMETRO | mm | mm |
| LONGITUD DE LA COSTURA | mm | | | |
| MATERIAL BASE | | RÉGIMEN DE SOLDADURA | | |
| NORMA | | DE RAÍZ | DE RELLENO | DE SELLADO |
| MARCA | | | | TOTAL |
| TIPO DE MATER. | | | | UNIDAD |
| MÁQUINA DE SOLDADURA | | | | |
| MARCA/MODELO | | | | |
| RANGO CORR. | | | | |
| TIPO DE CORR. | | | | |
| COSTURA | | | | |
| PREPARACIÓN DE BORDES | | | | |
| CÓDIGO | NORMA NC-ISO 9692-1 2011 | | | |
| | CRÓQUIS | | | |
| DIMENSIONES DE LA COSTURA | | PRECALENTAMIENTO | | REQUERIMIENTOS TECNOLÓGICOS |
| | | TEMPERATURA | °C | |
| | | TRATAMIENTO TÉRMICO POSTERIOR | | |
| | | TEMPERATURA | °C | |
| | | COSTOS | | |
| | | TIEMPO DE PERMANENCIA | min | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | COSTO DE ELECTRODO | CUP | |
| | | COSTOS | | |
| | | COSTO DE MANO DE OBRA | CUP | |
| | | COSTOS | | |
| | | COSTO DE ELECTRICIDAD | CUP | |
| | | COSTOS | | |
| | | COSTO TOTAL | CUP | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | PORTAELECTRODO | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | CABLE | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | FILTRO VISUAL | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | TECNÓLOGO | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | SOLDADOR | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | SUPERVISOR | | |
| | | EQUIPAMIENTO AUXILIAR | | |
| | | RESPONSABLE | | |
| | | PROCESO | | |
| | | SOLDADURA MANUAL CON ELECTRODO REVISTIDO | | |
| | |  | | |
| | | OBJETO | | |
| | | COSTURA | | |

Anexo 2: Ventana principal de la aplicación Opifex Spark/SMAW **Fuente:** Elaboración propia



Anexo 3: Flujo de trabajo en el proceso de soldadura al arco por metal revestido **Fuente:** CEFAS



Anexo 4: Flujo del proceso de llenado de cartas tecnológicas **Fuente:** CEFA

| Clasificación de entradas y consultas | 1 – 19 Atributos | 20 – 50 Atributos | Mas 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 2 ficheros accedidos | MEDIA 10 | ALTA 15 | ALTA 15 |

| Clasificación de entradas y consultas | 1 – 4 Atributos | 5 – 15 Atributos | Mas de 15 Atributos |
|---------------------------------------|-----------------|------------------|---------------------|
| 0 o 1 ficheros accedidos | BAJA 5 | BAJA 5 | MEDIA 7 |
| 2 ficheros accedidos | BAJA 5 | MEDIA 7 | ALTA 10 |
| Más de 2 ficheros accedidos | MEDIA 7 | ALTA 10 | ALTA 10 |

| Clasificación de entradas y consultas | 1 – 4 Atributos | 5 – 15 Atributos | Mas 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 |

| Clasificación de entradas y consultas | 1 – 5 Atributos | 5 – 19 Atributos | Mas de 19 Atributos |
|--|-----------------|------------------|---------------------|
| 0 o 1 ficheros accedidos | BAJA 4 | BAJA 4 | MEDIA 5 |
| 2 a 3 ficheros accedidos | BAJA 4 | MEDIA 5 | ALTA 7 |
| Más de 5 Entidades o registros lógicos | MEDIA 5 | ALTA 7 | ALTA 7 |

Anexo 5:Tabla de ponderaciones para EI, EQ, EO, ILF, EIF. **Fuente:** Elaboración Propia.

| No | Nombre del factor: | Valor |
|----|---|-------|
| 1 | Comunicación de datos | 4 |
| 2 | Procesamiento distribuido | 1 |
| 3 | Rendimiento | 4 |
| 4 | Configuración de Explotación Compartida | 1 |
| 5 | Tasa de transacciones | 1 |

| | | |
|----|--|-----------|
| 6 | Entrada de Datos en Línea | 3 |
| 7 | Eficiencia con el Usuario Final | 3 |
| 8 | Actualizaciones en Línea | 1 |
| 9 | Procesamiento complejo | 1 |
| 10 | Reusabilidad del Código | 1 |
| 11 | Facilidad de implementación | 2 |
| 12 | Facilidad de Operación | 4 |
| 13 | Instalaciones Múltiples | 1 |
| 14 | Facilidad de Cambios | 2 |
| | Ajuste de Complejidad Técnica (ACT) | 29 |

Anexo 6: Tabla de ajustes de complejidad técnica Fuente: Elaboración propia

| DATOS | | | ELECTRODOS | | | | | |
|------------------------|------------------|--------------------------|-------------------------------|--|--|--|--------------------|--|
| GEOMETRÍA DE LA UNIÓN | | | PASADAS DE RAÍZ | | PASADAS DE RELLENO | | PASADAS DE SELLADO | |
| TIPO DE UNIÓN | A solape | | Sureweld10P | | Sureweld10P | | — | |
| POSICIÓN | Plana | | FABRICANTE ESAB | | ESAB | | — | |
| DIMENSIONES | | | CÓDIGO AWS E6010 | | E6010 | | — | |
| ESPOSOR DE LA PLANCHA | 7.0 | mm | DIÁMETRO 2.4 | | mm | | — | |
| LONGITUD DE LA COSTURA | 12.0 | mm | — | | — | | — | |
| MATERIAL BASE | | | RÉGIMEN DE SOLDADURA | | | | | |
| NORMA | ANSI | | PASADAS DE RAÍZ | | PASADAS DE RELLENO | | PASADAS DE SELLADO | |
| MARCA | 1010 | | DE RAÍZ | | DE RELLENO | | DE SELLADO | |
| TIPO DE MATER. | Al carbono | | CANTIDAD DE PASADAS | | TOTAL | | UNIDAD | |
| MÁQUINA DE SOLDADURA | Indura150/Indura | | 82.6 | | 82.6 | | — | |
| MARCA/MODELO | 40-150(A) | | POLARIDAD CD + CDEN | | CD + CDEN | | — | |
| RANGO CORR. | CD | | CONSUMO DE ELECTRODO | | 2.0 | | kg | |
| TIPO DE CORR. | — | | CONSUMO DE ELECTRICIDAD | | 4.3 | | kWh | |
| COSTURA | | | TIEMPO TECNOLÓGICO | | 20.6 | | min | |
| PREPARACIÓN DE BORDES | | | TIEMPO AUXILIAR | | 0.6 | | min | |
| CÓDIGO | 1.3 | NORMA NC-ISO 9692-1 2011 | TIEMPO DE SERVICIO | | 0.8 | | min | |
| CROQUIS | | | TIEMPO DE DESCANSO | | 1.3 | | min | |
| CROQUIS | | | TIEMPO TOTAL | | 23.3 | | min | |
| | | | PRECALENTAMIENTO | | REQUERIMIENTOS TECNOLÓGICOS | | | |
| | | | TEMPERATURA — °C | | No hay Requerimientos | | | |
| | | | TRATAMIENTO TÉRMICO POSTERIOR | | | | | |
| | | | TEMPERATURA — °C | | | | | |
| | | | TIEMPO DE PERMANENCIA — min | | | | | |
| | | | COSTOS | | | | | |
| | | | COSTO DE ELECTRODO 3.23 | | CUP | | | |
| | | | COSTO DE MANO DE OBRA 2.7 | | CUP | | | |
| | | | COSTO DE ELECTRICIDAD 0.43 | | CUP | | | |
| | | | COSTO TOTAL 6.36 | | CUP | | | |
| | | | EQUIPAMIENTO AUXILIAR | | | | | |
| | | | PORTAELECTRODO Micro | | PROCESO | | | |
| | | | CABLE 2/0 x15.0m | | SOLDADURA MANUAL CON ELECTRODO REVESTIDO | | | |
| | | | FILTRO VISUAL N11 | | | | | |
| | | | TECNÓLOGO M.ortega | | | | | |
| | | | SOLDADOR B.Blanco | | OBJETO barril | | | |
| | | | SUPERVISOR R.Quiza | | COSTURA 33-65 | | | |
| | | | RESPONSABLE J.Perez | | CRT-11 | | | |

Anexo 7: Carta tecnológica finalizada Fuente: Elaboración propia