

UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA
CARRERA DE INGENIERÍA INFORMÁTICA



DESARROLLO DE UN SOFTWARE PARA LA OBTENCIÓN DE MODELOS MATEMÁTICOS EN LA
ETAPA DE EXTRACCIÓN PARA EL PROCESO DE FABRICACIÓN DE AZÚCAR CRUDO

Trabajo de Diploma para Optar por el Título de Ingeniero Informático

Autor: Hansser J. Díaz Rodríguez

Tutores: Msc. Jonathan Serrano Febles.

Ing. Eduardo J. Berrio Turiño.

Matanzas, 2022

Pensamiento.

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

Albert Einstein

Jamás ha habido una persona que haya tenido éxito sin haber pronunciado alguna vez en su vida las palabras "he fracasado"...

Arnold Schwarzenegger

Dedicatoria.

Agradecimientos.

Declaración de autoría.

Opinión del usuario del Trabajo de diploma.

Opinión del tutor del Trabajo de Diploma.

DATOS PERSONALES DEL TUTOR

Nombre y apellidos: Eduardo Javier Berrio Turiño.

Centro de trabajo: Universidad de Matanzas.

Organismo a que pertenece: Ministerio de Educación Superior – MES.

Cargo que ocupa: Profesor e Investigador.

Especialidad de la que es graduado: Ingeniero Informática. Universidad de Matanzas, 2018.

Categoría docente o investigativa: Asistente.

DATOS DE LA TESIS Y EL DIPLOMANTE

Nombre y apellidos: Hansser J. Díaz Rodríguez.

Centro de estudio: Universidad de Matanzas sede “Camilo Cienfuegos”.

Título de la Tesis: Desarrollo de un software para la obtención de modelos matemáticos en la etapa de extracción para el proceso de fabricación de azúcar crudo

OPINION SOBRE EL TRABAJO

La tesis presentada posee gran actualidad, pues intenta resolver un problema real, y de gran importancia. El tutor de este trabajo de diploma considera que, durante su ejecución, la estudiante mostró las cualidades que a continuación se detallan:

Fueron jornadas adentrándose en temas complejos y nuevos que profundizan en gran medida los recibidos durante la carrera, los cuales logró captar, adquiriendo el conocimiento necesario para enfrentar el problema planteado. Fue consecuente con los aspectos tanto metodológicos como de la investigación científica propiamente. Esto le permitió una feliz culminación del método desarrollado, de la documentación y de las pruebas realizadas.

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

También fueron horas de revisión, discusión y consenso en las que demostró cualidades para la investigación. El trabajo que hoy presenta y que sintetiza un periodo de aprendizaje no solo académico.

Como resultado se derivó en la obtención de un producto al nivel de las exigencias y expectativas. Por todo lo anteriormente señalado, considero que la estudiante Hansser J. Díaz Rodríguez reúne los requisitos para el título de Ingeniero Informático y espero le sea otorgada una buena calificación por parte de este Tribunal.



Ing. Eduardo J. Berrio Turiño

Dpto. Informática

Universidad de Matanzas

Noviembre/2022

Resumen

Intentar desarrollar un modelo matemático que refleje el comportamiento real del proceso el proceso de fabricación de azúcar crudo, en su etapa de extracción presenta dificultades, debido a que existen una gran cantidad de factores que no son considerados por las ecuaciones físicas teóricas que los describen. Podemos afirmar que los modelos matemáticos que delinear de mejor manera la respuesta son los modelos de regresión, ya que entre sus parámetros toma en cuenta la presencia de ruidos. El empleo de técnicas de inteligencia artificial específicamente la minería de datos constituye algo novedoso en esta área sin embargo, la necesidad del dominio de diferentes lenguajes de programación para su ejecución constituye una barrera actualmente debido a eso, la presente investigación persigue el desarrollo de un software que permita comunicación entre el entorno de aplicaciones de escritorio y lenguaje de programación enfocado en estadística, R y el lenguaje de programación Python para facilitar la interacción entre el software y el usuario para la obtención de modelos matemáticos predictivos , obteniendo así una interfaz más amigable. Esta investigación se realiza a raíz de una solicitud explícita del grupo empresarial AZCUBA a la facultad de Ciencias Técnicas de la Universidad de Matanzas.

Abstract

Trying to develop a mathematical model that reflects the real behavior of the process, the raw sugar manufacturing process, in its extraction stage presents difficulties, because there are a large number of factors that are not considered by the theoretical physical equations that describe them. We can affirm that the mathematical models that best delineate the response are the regression models, since the presence of noise is taken into account among its parameters. The use of artificial intelligence techniques, specifically data mining, constitutes something new in this area, however, the need to master different programming languages for its execution constitutes a barrier currently, due to this, the present investigation pursues the development of software that allows communication between the desktop application environment and programming language focused on statistics, R and the Python programming language to facilitate the interaction between the software and the user to obtain predictive mathematical models, thus obtaining a friendlier interface. This research is carried out as a result of an explicit request from the AZCUBA business group to the Faculty of Technical Sciences of the University of Matanzas.

Índice

Introducción	14
CAPÍTULO 1: Elementos teóricos relacionados con el entorno.	18
Introducción del capítulo	18
1.1 Python. Breve Reseña.	19
1.2 Interfaz gráfica del usuario.	20
1.2.1 Características.	21
1.2.2 Funciones Generales.	23
1.3 Construyendo interfaces visuales para Python.	23
1.3.1 Tk.....	23
1.3.2 Qt.....	24
1.4 Lenguaje R.....	25
1.5 Librerías Python	26
1.6 Conclusiones Parciales.....	27
CAPÍTULO 2: Diseño e implementación del software.	28
Introducción del capítulo	28
2.1 Principales conceptos de diseño y programación orientada a objetos.....	28
2.1.1 Conceptos Generales de la Programación Orientada a Objetos	28
2.2 Metodología de Software Extreme Programming (XP)	29
2.3 Captura de requisitos	30
2.3.1 Requerimientos Funcionales.....	30
2.3.2 Requerimientos no Funcionales	31

2.4 Historia de usuario	32
Orden	32
HU	32
Nombre	32
Cargar documento Excel	32
Riesgos.....	32
Medio	32
Prioridad	32
Alta	32
Iteración.....	33
1	33
Puntos de estimación.....	33
0.4	33
Descripción	33
El software debe ser capaz de leer los datos de las variables almacenadas en el documento Excel cuyo directorio dentro de la estación de trabajo lo establece el usuario. Los datos presentes en el Excel deben ser mostrados en pantalla en forma de tabla.....	33
Observación.....	33
Los nombres de las variables presentes en el fichero deben ser los nombres de las cabeceras de las columnas de las tablas.	33
2.5 Estimación de esfuerzo por Historias de Usuario	34
2.5.1 Plan de iteraciones	35
2.5.2 Plan de entrega.....	36
2.5.3 Estimación del costo.....	37
2.5.4 Coste de personal.....	37
2.5.5 Coste total	37
2.6 Conclusiones del capítulo.....	38
Capítulo 3: Implementación y prueba.....	39
Introducción	39
3.1 Tarea de ingeniería.....	39
3.2 Estándares de codificación.....	40
3.3 Pruebas de software.....	40
3.3.1 Pruebas de aceptación	40
3.3.2 Pruebas de satisfacción de usuarios	41

3.4 Resultados de las pruebas	42
3.5 Conclusiones parciales del capítulo.....	44
Conclusiones Generales.....	45

Índice de Tablas

Tabla 1 Requisitos Funcionales	30
Tabla 2 Requisitos no Funcionales.....	31
Tabla 3 Historias de usuario	32
Tabla 4 Historias de usuarios agrupadas por requisitos	33
Tabla 5 Estimación de esfuerzo por Historia de Usuario.	34
Tabla 6 Plan de Iteraciones	35
Tabla 7 Tabla ejemplo de tarea de ingeniería.	39
Tabla 8 Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios.	42
Tabla 9 Resultados de la pregunta 2 de las pruebas de satisfacción de usuarios.	43
Tabla 10 Resultados de la pregunta 3 de las pruebas de satisfacción de usuarios.	43
Tabla 11. Resultados de la pregunta 5 de las pruebas de satisfacción de usuarios.....	44

Introducción

Décadas atrás Cuba constituyó por mucho tiempo uno de los más grandes productores de azúcar en el orbe; no obstante, el esplendoroso desarrollo azucarero sufrió las serias consecuencias de depresiones económicas y de decisiones no acertadas que provocaron la paralización y retroceso de esta actividad. Ante ello y por el reposicionamiento del azúcar como un producto altamente cotizado en el mercado internacional, desde los últimos años el grupo empresarial AZCUBA, encausa cuantiosos recursos por potenciar la intensificación de la producción fabril.

Establecer las condiciones de operaciones adecuadas para la etapa de extracción de dicho proceso, es una de las decisiones más importantes y difíciles, dada las circunstancias tan cambiantes que existen en la caña molida (Hugot, 2014; Rein, 2012). Incorrectos valores de agua de imbibición conllevan a un mal aprovechamiento del azúcar que trae la caña, así como compuestos indeseables que afectan la calidad del mijo, una incorporación inadecuada del almidón y deficiencias en la generación de vapor por exceso de humedad del bagazo, entre otros graves problemas operacionales, que repercuten directamente sobre la productividad del proceso e indicadores de eficiencias económicos, tecnológicos y medioambientales (Batule, 2009; Jenkins, 2013; Kent, 2011).

En la actualidad ningún proceso real tiene una respuesta lineal e invariante en el tiempo por lo que presenta distintas funciones de transferencia en diferentes rangos de trabajo. Para realizar un modelado matemático teórico, se debe recurrir a ecuaciones de física, las que son idealizadas; por tanto, son irreales e imprecisas. Para poder plasmar las no linealidades en ecuaciones físicas se necesita conocer teorías de control no lineal y analizar los diferentes efectos que se producen.

Intentar desarrollar un modelo matemático que refleje el comportamiento real del proceso el proceso de fabricación de azúcar crudo, en su etapa de extracción presenta dificultades, debido a que existen una gran cantidad de factores que no son considerados por las ecuaciones físicas teóricas que los describen. El empleo de técnicas de inteligencia artificial específicamente la minería de datos constituye algo novedoso

en esta área, sin embargo, la necesidad del dominio de diferentes lenguajes de programación para su ejecución constituye una barrera actualmente debido a eso.

Para mitigar ese inconveniente se hace imprescindible disponer de herramientas informáticas que faciliten la toma de decisiones operacionales. El empleo de softwares para el control industrial de procesos tecnológicos es clave para alcanzar adecuados rendimientos productivos. Estos programas permiten el análisis de información de proceso, para la detección temprana de problemas operacionales y la mejora continua y sistemática del mismo.

Propiamente para la etapa de extracción de la industria azucarera, no se dispone de un programa informático, que permita en un entorno accesible, el desarrollo de modelos matemáticos predictivos. La mayoría de los trabajos realizados al respecto son sobre la base de programas de Matlab y R, los que requieren del conocimiento de estos lenguajes de programación; frecuentemente son de difícil comprensión para el público no especializado. Ello imposibilita que el personal técnico pueda desarrollar las herramientas de análisis necesarias para un mejor control y mejora del proceso industrial.

De la situación problémica anteriormente descrita se identifica el siguiente **problema científico** de investigación, ¿es posible elevar la efectividad en la etapa de extracción para el proceso de fabricación de azúcar crudo empresarial a partir de una herramienta automatizada para la obtención de modelos matemáticos?

Para la solución del problema se trazó **la hipótesis** que rige la investigación, si se desarrolla software para la obtención de modelos matemáticos en la etapa de extracción para el proceso de fabricación de azúcar crudo se contribuirá a elevar la efectividad de este proceso.

El **objeto de estudio** de esta investigación que se identifica como el proceso obtención de modelos matemáticos en la etapa de extracción para el proceso de fabricación de azúcar crudo.

Como **campo de acción** la informatización del proceso de obtención de modelos matemáticos en la etapa de extracción para el proceso de fabricación de azúcar crudo.

Se define como **objetivo general** de la investigación: desarrollar un software para la obtención de modelos matemáticos en la etapa de extracción para el proceso de fabricación de azúcar crudo que contribuya a la elevación de la eficacia y la eficiencia de este proceso.

Con el propósito de dar cumplimiento al **objetivo general** de esta investigación se trazaron los siguientes objetivos específicos:

1. Analizar la bibliografía que permita sentar las bases de la investigación y conformar el marco teórico referencial de la misma.
2. Diseñar la propuesta de solución utilizando la metodología XP.

3. Validar la propuesta de solución a partir de la aplicación de pruebas de software.

Durante el desarrollo de la investigación se emplearon diferentes **métodos teóricos** como:

- **Analítico – Sintético:** Se utilizó para realizar un análisis y una revisión de forma general de toda la bibliografía e información relacionada con el tema, y posteriormente extraer los aspectos fundamentales que sirvieron de apoyo para el desarrollo de la investigación.
- **Dialéctico:** Se realizó un estudio de todos los procesos que intervienen en la aplicación a implementar, considerando que constantemente pueden estar sujetos al cambio, es decir, siempre hay que tener presente que el sistema que se desarrolla puede sufrir modificaciones de acuerdo a las necesidades, que nunca se va a mantener igual.
- **Sistémico:** Se consideran a todos los elementos, componentes y procesos presentes en la aplicación interrelacionados entre sí y conformando un sistema, es decir, hay que verlo como un todo, no se puede estudiar cada proceso por separado, porque cada uno va a depender de los demás, y el funcionamiento de todos ellos es lo que va a constituir un sistema.
- **Modelación:** Define los procesos del dominio del problema, estima los principales riesgos del proyecto y la forma de mitigarlos, además se identifican las necesidades del cliente, se define la estructura del software y se validan los artefactos generados.

Así mismo, se emplearon **métodos empíricos**:

- **Entrevistas:** A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y análisis de los resultados.
- **Observación:** Se aplicó con el objetivo de captar información de forma visual, es decir, mediante observaciones realizadas a otras aplicaciones ya existentes vinculados con el tema, se pudieron tomar elementos que sirvieron de apoyo para el desarrollo de este sistema.
- **Tormenta de ideas:** esta técnica genera muchas ideas y también soluciones a problemas determinados. Además, se aprovecha la capacidad de todos y no de unos pocos. Genera un sentido de responsabilidad compartido.
- **Análisis de antecedentes:** consiste en estudiar los sistemas ya creados que estén relacionados con aplicaciones móviles para la comunicación entre personas oyentes y no oyentes.

Entre los **aportes** de la investigación se destacan:

- El **teórico-investigativo**, al analizar los métodos y procedimientos tradicionales más utilizados por autores relacionados con los catálogos multiplataforma a través de diferentes fases, etapas y pasos que permiten orientar metodológicamente la secuencia de acciones lógicas a desarrollar y los elementos a tener en cuenta para la continuidad de la investigación.
- El **práctico**, al desarrollar una aplicación para móviles que permita la comunicación entre personas oyentes y no oyentes.
- El **económico**, al contribuir a un manejo más eficiente de la información, lo que aumenta la efectividad del trabajo y, por ende, la producción de la empresa.

El resultado esperado de este trabajo es contar con una aplicación móvil que viabilice la comunicación entre personas que hablen lengua de señas y personas que hablan lenguaje natural, mediante el empleo de algoritmos de reconocimiento de patrones y por medio de imágenes consumiendo servicios.

Atendiendo a lo planteado anteriormente, la tesis queda estructurada en introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas, según sigue:

- Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
- Un primer capítulo donde se recoge el marco teórico referencial del tema y los principales conceptos que constituyen la base teórica de la investigación, así como el análisis de las principales tendencias tecnológicas y el estudio de los antecedentes que enmarcan la problemática planteada.
- Un segundo capítulo en el que se argumenta la solución que se propone al problema de investigación mediante su descripción
- Un tercer capítulo En el que se describe el software y se le realizan las pruebas con el objetivo de entregarle al cliente un producto totalmente funcional, que cumple con todos los requisitos demandados por el mismo y que satisfaga sus necesidades. Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
- Y las referencias de la bibliografía citada.

CAPÍTULO 1: Elementos teóricos relacionados con el entorno.

Introducción del capítulo

El objetivo de este capítulo es realizar una descripción general de los aspectos esenciales tanto del lenguaje Python en sí como exponer las características principales de una interfaz de usuario, y proponer algunas de las herramientas principales para la construcción de la misma. El lenguaje Python es usado en la actualidad para el desarrollo de aplicaciones, estas están relacionadas con disímiles ramas. La selección de las técnicas correctas para la programación de dichas aplicaciones para obtener resultados que se reflejen en un producto final de calidad reviste gran importancia, dada la gran variedad de herramientas existentes para tal propósito.

Al comenzar con la investigación se realizó una búsqueda con el objetivo de verificar la existencia de sistemas o aplicaciones informáticos similares que cumpliera las exigencias del cliente, pero no se encontraron resultados semejantes.

El *software* tendrá un conjunto de modelos matemáticos predeterminados, los que fueron desarrollados por Serrano y Orozco (2020). Estos modelos matemáticos cuentan con la aprobación del grupo empresarial AZCUBA y se tiene el permiso de uso de los autores. Dichos modelos matemáticos son los siguientes:

$$JM_{BX} = 0.858 \cdot JP_{BX} - 0.01872 \cdot \lambda + 0.014 \cdot T + 0.876 \quad (1)$$

$$JM_{POL} = 0.843 \cdot JP_{POL} - 0.01744 \cdot \lambda + 0.011 \cdot T + 1.592 \quad (2)$$

$$Bg_W = 48.156 + 0.01043 \cdot \lambda - 0.010 \cdot T \quad (3)$$

$$Bg_{BX} = \text{EXP}(1.620 - 0.00425 \cdot \lambda - 0.002 \cdot T + 0.040 \cdot JP_{BX}) \quad (4)$$

$$Bg_{POL} = \text{EXP}(3.8363 - 0.683 \cdot \ln\lambda - 0.002 \cdot T + 0.039 \cdot JP_{POL}) \quad (5)$$

Donde:

JM_{BX} : Brix del Jugo Mezclado (°Bx)

λ : Razón de Imbibición (%)

T : Temperatura del agua de imbibición (°C)

JP_{BX} : Brix del jugo primario (°Bx)

JM_{POL} : Pol del jugo mezclado (%)

JP_{POL} : Pol del jugo primario (%)

Bg_W : Humedad del bagazo (%)

Bg_{BX} : Brix del bagazo (%)

Bg_{POL} : Pol del bagazo (%)

1.1 Python. Breve Reseña.

Python es un lenguaje de programación creado por Guido van Rossum en el año 1990. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Python permite dividir el programa en módulos reutilizables desde otros programas.

Python viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas o como ejemplos para empezar a aprender Python. También hay módulos incluidos que proporcionan entrada y salida de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk, Gtk, Qt entre otros, siendo estos los más conocidos.

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete de Python estándar incluye un *modo interactivo*, en el cual se escriben las instrucciones en una especie de shell, las expresiones pueden ser introducidas una por una, pudiendo verse el resultado de su evaluación inmediatamente. Esto resulta útil tanto para las personas que se están familiarizando con el lenguaje como también para los programadores más avanzados, pues se pueden probar porciones de código en el modo interactivo antes de integrarlo como parte de un programa.

En los últimos años Python ha experimentado un importantísimo aumento del número de programadores y empresas que lo utilizan. Algunos de sus desarrolladores y clientes opinan:

Python ha sido parte importante de Google desde el principio, y lo sigue siendo a medida que el sistema crece y evoluciona. Hoy día, docenas de ingenieros de Google usan Python y seguimos buscando gente diestra en este lenguaje. Peter Norvig, director de calidad de búsquedas de Google Inc. (Gracia, 2003)

Python juega un papel clave en nuestra cadena de producción. Sin él, un proyecto de la envergadura de Star Wars: Episodio II hubiera sido muy difícil de sacar adelante. Visualización de multitudes, proceso de lotes, composición de escenas. . . Python es lo que lo une todo. Tommy Brunette, director técnico senior de Industrial Light & Magic. (Gracia, 2003)

Python está en todas partes de Industrial Light & Magic. Se usa para extender la capacidad de nuestras aplicaciones y para proporcionar la cola que las une. Cada imagen generada por computador que creamos incluye a Python en algún punto del proceso. Philip Peterson, ingeniero principal de I+D de Industrial Light & Magic (Gracia, 2003)

Este lenguaje debido a sus características es usado en varias ramas de la industria, sobre todo en el área científica, el desarrollo de software y en la ingeniería.

1.2 Interfaz gráfica del usuario.

Los avances de la ciencia y la tecnología han puesto al hombre en un plano intermedio entre lo tangible e intangible computacionalmente hablando, es ahora tan común el convivir con una computadora diariamente que cada vez se hace más imperativo la mejor interacción hombre-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde comodidad y eficiencia.

La idea fundamental en el concepto de interfaz es el de mediación, entre hombre y máquina. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como la computadora. Esto implica, además, que se trata de un sistema de traducción, ya que los dos "hablan" lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso del procesador electrónico.

De una manera más técnica se define Interfaz de usuario, *como conjunto de componentes empleados por los usuarios para comunicarse con las computadoras.* El usuario dirige el funcionamiento de la máquina mediante instrucciones, denominadas genéricamente entradas. Las entradas se introducen mediante diversos dispositivos, por ejemplo un teclado, y se convierten en señales electrónicas que pueden ser procesadas por la computadora. Estas señales se transmiten a través de circuitos conocidos como bus, y son coordinadas y controladas por la unidad de proceso central y por un soporte lógico conocido como sistema operativo. Una vez que el procesador ha ejecutado las instrucciones indicadas por el usuario, puede

comunicar los resultados mediante señales electrónicas, o salidas, que se transmiten por el bus a uno o más dispositivos de salida, por ejemplo una impresora o un monitor.

Resumiendo entonces, una interfaz de software es la parte de una aplicación que el usuario ve y con la cual interactúa. Está relacionada con la subyacente estructura, la arquitectura, y el código que hace el trabajo del software, pero no se confunde con ellos. La interfaz incluye las pantallas, ventanas, controles, menús, metáforas, la ayuda en línea, la documentación y el entrenamiento. Cualquier cosa que el usuario ve y con lo cual interactúa es parte de la interfaz. Una interfaz inteligente es fácil de aprender y usar. Permite a los usuarios hacer su trabajo o desempeñar una tarea en la manera que hace más sentido para ellos, en vez de tener que ajustarse al software. Una interfaz inteligente se diseña específicamente para la gente que la usará. (Toledo, 1996)

1.2.1 Características.

Para lograr una buena interacción con el usuario según (Lores, 2006), la interfaz debe lograr una armonía entre las principales características de una interfaz que son:

Facilidad de manejo: permite poder familiarizarse con su funcionamiento básico de cara a centrarse en los contenidos y en la navegación. La mejor interfaz de usuario es la que requiere el menor esfuerzo de aprendizaje.

Originalidad: con ello se consigue motivar a la utilización de la aplicación y promover la exploración por los elementos de la interfaz.

Homogeneidad: la interfaz es la herramienta que va a acompañar a lo largo del viaje por la información y por eso tiene que ser lo suficientemente estable u homogénea como para familiarizarse con su funcionamiento en el menor tiempo posible, y lo suficientemente versátil para que se amolde a distintas actividades dentro de la misma aplicación. La interfaz permitirá contar con unas funciones básicas claras como: salir del programa, seleccionar un módulo, cambiar de tipo de actividad, mapa general para moverse por la aplicación, etc.

Consistencia: en situaciones similares se debe emplear la misma secuencia de acciones. Se debe utilizar una terminología idéntica en los mensajes, menús y pantallas de ayuda. Un ejemplo de consistencia sería usar siempre el comando Nuevo para crear cualquier elemento, en vez de tener comandos alternativos en función de la situación.

Versatilidad: hará que la interfaz incorpore, manteniendo esa estructura básica, nuevas funciones específicas para una actividad concreta como pueden ser la resolución de casos, el control de variables en una simulación, etc.

Reversibilidad: debe permitir deshacer acciones. Las unidades reversibles pueden ser una acción simple, una entrada de un dato o un grupo de acciones completo. Por ejemplo, cuando se borra un elemento por confusión se debe poder anular la acción, recuperando el elemento eliminado.

Adaptabilidad: consiste en ofrecer al usuario la posibilidad de seleccionar las modalidades de navegación en función del tipo de contenidos, de los destinatarios y de los niveles de profundidad previstos.

Multimodalidad: es una característica principal de una interfaz de navegación.

Se trata de ver la forma de cómo integrar las distintas modalidades de comunicación que pueden ser necesarias para cada concepto. Un concepto o idea puede requerir en paralelo un visionado de un proceso en forma de esquema y a su vez la imagen real con la posibilidad de consultar elementos concretos del proceso.

Agilidad: una buena interfaz no debe quedarse en una inmejorable presentación física o artística, sino que, sobre todo, tiene que haber un impecable engranaje que se ponga en marcha para cada acción del usuario y que se traduce en un funcionamiento ágil y dinámico del programa.

Transparencia: la interfaz debe ser ante todo natural y transparente al usuario, de forma que las posibles actividades implicadas en el aprendizaje de un tema concreto se puedan producir directamente sin mediar procesos incomprensibles o complejos.

Interactividad: el sistema de navegación y la interfaz han de ser interactivos, esto es, deben ofrecer la posibilidad al usuario de que se sienta protagonista de su propia experiencia.

Conectividad: un aspecto de la interfaz que se está investigando mucho es la incorporación de la interconexión de redes. En este ámbito se tiende igualmente a simplificar y hacer amigable el acceso remoto a la información disponible a través de terminales de todo el mundo.

1.2.2 Funciones Generales.

Una interfaz de usuario según (Lores, 2006) ha de contar con unas funciones generales que se pueden agrupar en:

1. Utilidades:

Incluye herramientas para la edición, la planificación y la comunicación. Con la utilidad de edición se pueden tomar notas, crear vínculos entre los contenidos, modificar, imprimir, etc.

Con la utilidad de planificación el usuario puede preparar sus actividades. La utilidad de comunicación permite el envío y recepción de información tanto con el programa informático como con el exterior y el contacto con otros usuarios.

2. Navegación:

Se incluyen funciones para un recorrido que puede ser libre o guiado a través de la red de contenidos y que permita saber en todo momento donde se está, de donde se viene y a donde se puede ir (claves para la orientación).

3. Información:

Están las funciones de ayuda del sistema, profundización en temas y seguimiento del camino recorrido.

En la ayuda se encuentra la información sobre el funcionamiento del entorno.

En el seguimiento se tiene acceso a una descripción, y en su caso, valoración del progreso en función del grado de consecución de los objetivos.

1.3 Construyendo interfaces visuales para Python.

Python es un lenguaje que permite la implementación de interfaces de usuario, a través de las extensiones que existen para el propio lenguaje, de los principales sistemas de ventanas multiplataformas existentes en la actualidad. En este epígrafe realizaremos un análisis de algunas de ellas.

1.3.1 Tk.

Tk es una aplicación software libre, multiplataforma constituye una biblioteca de elementos básicos para construir una interfaz gráfica que fue desarrollado por Jhon Ousterhout.

Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python, con estos queremos decir que se encuentra disponible para varios lenguajes de programación entre los cuales se encuentra Python. Este no es más que una capa de esta librería para el lenguaje Python con lo cual usar Tk en otro lenguaje no nos supondrá un inconveniente.

1.3.2 Qt

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario. Fue creada por la compañía noruega Trolltech. Qt es utilizada en KDE, un entorno de escritorio para sistemas como GNU/Linux o FreeBSD entre otros. Utiliza el lenguaje de programación C++ de forma nativa y además existen extensiones para C, Python, Java y varios lenguajes más.

La extensión para Python es conocida como PyQt está desarrollada por la firma británica Riverbank Computing y está disponible para Windows, GNU/Linux y Mac OS X bajo diferentes licencias. Es un completo conjunto de elementos gráficos (listados, arboles, grillas, etc.)Flexible y potente control del comportamiento de la interface. Posee un mecanismo de conexión de señales y eventos simple. Se puede definir los eventos más sencillos en diseñador de GUI's (ej: al pulsar este botón, borrar este campo de texto) y en el código python, definir las acciones más avanzadas. Rápido y de Apariencia nativa (las últimas versiones utilizan funciones nativas en windows) se puede separar el diseño de la interface, pero usa un "compilador" pyuic para crear las clases python. Arquitectura opcional para Modelo/Vista para las tablas, listas y árboles.

Tanto Tkinter como PyQt son útiles para diseñar interfaces gráficas de usuario aceptables, pero al mismo tiempo difieren en términos de adaptabilidad y funcionalidad.

La mayoría de las veces, Tkinter se trata de escribir GUI usted mismo, programar su configuración o funcionalidad en el mismo script.

Por otro lado, en PyQt, se separa la interfaz gráfica de usuario en un script y se utiliza el conocimiento de Python de otro script.

En lugar de crear su propio código para la interfaz de usuario, *puede simplemente adoptar las funciones Qt Designer para desarrollar su aplicación.*

Por lo tanto, veamos cuáles son las principales diferencias y ventajas de PyQt vs. Tkinter.

Ventajas de usar PyQt

1. Flexibilidad de codificación - La programación GUI con Qt está diseñada alrededor del concepto de señales y slots para establecer comunicación entre objetos. Esto permite flexibilidad cuando se trata de eventos GUI y resulta en una base de código más fluida.

2. Más que un marco de trabajo - Qt utiliza una amplia gama de APIs de plataformas nativas con el propósito de *redes, creación de bases de datos y mucho más*. Ofrece acceso primario a ellos a través de una API única.
3. Varios componentes de la interfaz de usuario - Qt ofrece varios widgets, como botones o menús, todos ellos diseñados con una apariencia básica en todas las plataformas soportadas.
4. Varios recursos de aprendizaje: dado que PyQt es uno de los marcos de trabajo de interfaz de usuarios más utilizados para Python, puede acceder fácilmente a una amplia gama de documentación.
5. Fácil de dominar - PyQt viene con una funcionalidad API sencilla y fácil de usar, junto con clases específicas vinculadas a Qt C++. Esto permite al usuario utilizar conocimientos previos de Qt o C++, haciendo que PyQt sea fácil de entender.

Desventajas de usar PyQt

1. Falta de documentación específica de Python para las clases de PyQt5
2. Requiere mucho tiempo para entender todos los detalles de PyQt, lo que significa que es una curva de aprendizaje bastante empinada.

Ventajas de usar Tkinter

1. Disponible sin cargo para uso comercial.
2. Se presenta en la biblioteca Python subyacente.
3. La creación de ejecutables para las aplicaciones de Tkinter es más accesible ya que Tkinter está incluido en Python, y, como consecuencia, no viene con ninguna otra dependencia.
4. Simple de entender y dominar, ya que Tkinter es una librería limitada con una API simple, siendo la opción principal para crear interfaces gráficas de usuario rápidas para scripts Python.

Desventajas de usar Tkinter

1. Tkinter no incluye widgets avanzados.
2. No tiene una herramienta similar a Qt Designer para Tkinter.
3. No tiene un aspecto nativo

1.4 Lenguaje R

Es un entorno y lenguaje de programación con enfoque al análisis estadístico. Acumula, en un único programa, todo lo necesario para analizar los datos con eficacia. Las diversas herramientas de medición de datos y de realización de estadísticas se convierten en imprescindibles para ahorrar tiempo y esfuerzo. Incluye modelos lineales y no lineales usados en el sector de la estadística. Es idóneo para la clasificación y el agrupamiento de los datos, facilitando así su posterior interpretación. Pero a su vez cuenta con desventajas considerables. No soporta gráficos en tres dimensiones o dinámicos. El resultado de cada informe puede ser algo pobre visualmente y bastante anticuado en comparación con el ofrecido por otros programas. Su lentitud le resta efectividad y competitividad. Existen lenguajes de programación más rápidos que ofrecen una capacidad de respuesta más adecuada. Los algoritmos no están unificados. Cada uno de ellos se almacena en un paquete distinto, por lo que hay que ir cambiando de opción para leer los datos obtenidos. No tiene medidas de seguridad, por lo que es muy fácil de vulnerar por parte de los ciberdelincuentes. Poner en riesgo la información personal de los clientes de una empresa puede provocar la toma de medidas legales que terminen generando un serio problema económico.

1.5 Librerías Python

Numpy: añade a Python funcionalidades para el manejo sencillo y eficiente de operaciones matriciales y vectoriales, desde las más básicas hasta las más complejas. Además, proporciona herramientas que permiten incorporar código fuente de otros lenguajes de programación como C/C++ o Fortran, lo que incrementa notablemente su compatibilidad e implementación.

Pandas: Define nuevas estructuras de datos basadas en los arrays de la librería NumPy pero con nuevas funcionalidades. Permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL. Permite acceder a los datos mediante índices o nombres para filas y columnas. Ofrece métodos para reordenar, dividir y combinar conjuntos de datos. Permite trabajar con series temporales. Realiza todas estas operaciones de manera muy eficiente.

Statsmodels: es una biblioteca de análisis y modelado de datos estadísticos. Completa en gran medida la regresión lineal clásica proponiendo nuevos estimadores de mínimos cuadrados. Normalmente, se utilizan los mínimos cuadrados ordinarios (OLS) para estimar la regresión lineal. Sin embargo, en el caso de que algunos residuos estén correlacionados, la regresión lineal deja de ser eficaz. Los mínimos cuadrados generalizados (GLS) pueden entonces ayudar a resolver este problema. Paralelamente a este método, statsmodels presenta otros dos estimadores de mínimos cuadrados ordinarios más complejos: mínimos cuadrados recursivos (RLS) y mínimos cuadrados móviles (MLS).

Scikit-learn: es uno de los open-source y bibliotecas de aprendizaje automático más populares en Python. La biblioteca scikit-learn contiene muchas herramientas eficientes para aprendizaje automático y modelado estadístico, incluyendo clasificación, regresión, agrupación, y reducción de dimensionalidad.

Scipy: es una librería de herramientas numéricas que usa Numpy como base. Posee módulos para optimización de funciones, integración, funciones especiales, y resolución de ecuaciones diferenciales ordinarias, entre otros.

Matplotlib: es una biblioteca para la generación de gráficos 2D y 3D a partir de datos contenidos en listas o arreglos.

Los entornos de interfaz de usuario: PyQt5

Lo forman un conjunto de funciones y clases definidas que realizan las acciones pertinentes para procesar los resultados. Todo ello está escrito en lenguaje Python de programación y requiere que el usuario interactúe con el programa introduciendo las variables de entrada que se procesan. Para ello, se implementó en su momento una interfaz de usuario preparada con la librería de Python PyQt5. De esta manera, el usuario interactúa con estas funciones y puede realizar las acciones del programa cómodamente, mediante una ventana que incluye botones y una pantalla de texto junto con una tabla de datos. Los botones de la interfaz están conectados con las funciones del código del programa de manera que cada vez que se hace clic en uno de los iconos, se llama a una parte del código que realiza las acciones correspondientes a cada botón. Los resultados procesados por las funciones, se presentan al usuario en la misma pantalla de la interfaz.

1.6 Conclusiones Parciales.

Después de realizada una revisión bibliográfica sobre temas relacionados con el tema referente en este capítulo se ha arribado a las siguientes conclusiones:

Python es un lenguaje usado en la elaboración de herramientas para disímiles ramas y su uso a nivel mundial va en aumento.

Python debido a sus características y prestaciones es un lenguaje a tener en cuenta por el país en su estrategia en el área de la informática.

En la construcción de interfaces visuales de usuario, se tienen en cuenta algunas características para lograr la calidad de la misma.

El uso de software libre permite en gran medida el posterior desarrollo y mejora de los proyectos desarrollados bajo este concepto.

CAPÍTULO 2: Diseño e implementación del software.

Introducción del capítulo

En el presente capítulo se abordan aspectos generales relacionados con la implementación de una aplicación capaz de lograr interactuar con Python y se analizan algunos aspectos relacionados con la Programación Orientada a Objetos, paradigma empleado en la elaboración del software.

2.1 Principales conceptos de diseño y programación orientada a objetos

Es importante tratar algunos conceptos fundamentales de la Programación Orientada a Objetos (POO), así como aspectos sobre la metodología de análisis y diseño orientada a objetos UML utilizada durante las etapas de desarrollo del sistema (Rumbaugh et al., 1997a, Rumbaugh et al., 1997b)

2.1.1 Conceptos Generales de la Programación Orientada a Objetos

Los módulos que en este capítulo se describen son realizados utilizando el paradigma de POO. Es necesario precisar cuáles son las características particulares y las propiedades que son utilizadas en este diseño para identificar los objetos.

En (Rodríguez, 1999) se presentan las siguientes propiedades de los objetos:

Estado: Se define a partir de los valores que en un momento dado tienen los atributos del objeto. La estructura del objeto se define como el conjunto de todos los atributos o propiedades. Además, un objeto puede conocer o contener a otros objetos, estas relaciones son también parte de su estado.

Comportamiento: Define cómo actúan los objetos frente a estímulos externos en términos de cambio de estados.

Identidad: Es la propiedad de un objeto que lo distingue del conjunto de todos los demás objetos del universo al que pertenece. Los modelos de POO son representaciones abstractas de este tipo.

El marco de referencia conceptual en un sistema orientado a objeto es el modelo de objetos que incluye cuatro conceptos fundamentales:

Abstracción: Denota las características esenciales de un objeto que lo distinguen de todos los demás tipos de objeto y proporciona así fronteras conceptuales nítidamente definidas respecto a la perspectiva del observador.

Encapsulamiento: Es uno de los principios más importantes de la POO, ha permitido la reusabilidad de objetos. Constituye el proceso de almacenar en un mismo compartimiento los elementos de una abstracción que constituyen su estructura y su comportamiento. El cliente se interesa por lo que hace el objeto y no cómo lo hace

Modularidad: Es la propiedad que tiene un sistema que ha sido descompuesto en un conjunto de módulos cohesivos y débilmente acoplados.

Jerarquía: Es una clasificación u ordenamiento de abstracciones.

Otros conceptos importantes a tener en cuenta son:

Concurrencia: Es la propiedad que distingue un objeto activo de uno que no está activo.

Persistencia: Es la propiedad de un objeto por la que su existencia trasciende el tiempo, el espacio, o ambos.

En (Rodríguez, 1999) se definen los conceptos de clase y tipo: *La clase no es más que una representación abstracta que define la estructura y el comportamiento que le son comunes a un grupo de objetos. Mientras que el tipo es un protocolo usado en los mecanismos de comunicación e interacción entre objetos. Tiene identidad y generalmente está más relacionado a los mecanismos de comunicación que a la propia naturaleza de los objetos.*

2.2 Metodología de Software Extreme Programming (XP)

Extreme Programming (XP) es una metodología ágil de desarrollo de software con bases en la comunicación constante y la retroalimentación. Uno de sus fines principales es el de construir un producto que vaya en línea con los requerimientos del cliente.

En ese sentido es adaptable a los cambios, generando una rápida respuesta frente a cualquier inconveniente. Por otro lado, el equipo de trabajo tiene la ventaja de potenciar sus relaciones, ya que el proceso que de este se desprende es abierto, conjunto y de aprendizaje continuo.

En la siguiente nota se presentan las 4 etapas que sigue la Programación Extrema:

1. **Planificación:** toma como referencia la identificación de la historia del usuario con pequeñas versiones que se irán revisando en periodos cortos con el fin de obtener un software funcional.
2. **Diseño:** trabaja el código orientado a objetivos y, sobre todo, usando los recursos necesarios para que funcione.
3. **Codificación:** se refiere al proceso de programación organizada en parejas, estandarizada y que resulte en un código universal entendible.

4. **Pruebas:** consiste en un testeo automático y continuo en el que el cliente tiene voz para validar y proponer. Es, en pocas palabras, la prueba de aceptación.

2.3 Captura de requisitos

A través de la captura de requisitos se recopila la información y se transforma en un conjunto de requerimientos que son los que darán límite al alcance del sistema. Estos requisitos pueden ser funcionales y no funcionales. Los requisitos funcionales especifican los servicios o funciones que proveerá el sistema, mientras que los no funcionales definen las restricciones que tendrá este, las cuales limitan las elecciones para construir una solución.

2.3.1 Requerimientos Funcionales

Tabla 1 Requisitos Funcionales

Nombre de la Funcionalidad		
Código	Requisito	Descripción
RF01	Cargar documento Excel.	El software debe ser capaz de interpretar los datos de las variables almacenadas en el documento Excel que establezca el usuario y mostrarlos en pantalla.
RF02	Seleccionar las variables a relacionar para la confección de modelos.	El usuario debe seleccionar las ya enlistadas variables que desea correlacionar para la creación de los modelos matemáticos.
RF03	Generar gráficas de correlación de variables.	Se mostrarán gráficos de dispersión de apoyo al usuario para la confección de los modelos matemáticos más adecuados.
RF04	Gestionar modelos.	El software debe permitir la gestión de los modelos (Crear, modificar, eliminar y visualizar detalles.)
RF05	Análisis de modelos.	El software debe permitir la visualización de parámetros que permiten la comparación entre los modelos confeccionados.

RF06	Pruebas del modelo seleccionado.	El software debe ser capaz de mostrar pruebas de estadísticas que permitan analizar la calidad de los modelos matemáticos y visualizar el cumplimiento de los supuestos matemáticos de la regresión lineal.
RF07	Calidad de ajuste al modelo	El software debe determinar pruebas de bondad de ajuste matemático.
RF08	Información de extrapolación.	El software debe ser capaz de mostrar resultados de extrapolación oculta.
RF09	Validación.	Realizar pruebas de validación de modelos.

2.3.2 Requerimientos no Funcionales

Tabla 2 Requisitos no Funcionales

Código	Requisito	Descripción
RFN01	Usabilidad	El sistema deberá ser usado por personas con conocimientos básicos en el manejo de computadoras y que sean especialistas en la etapa de extracción para la fabricación del azúcar crudo.
RFN02	Estabilidad	El sistema debe ser capaz de manejar los errores ocurridos durante la ejecución de la misma y avisando de la naturalidad del error
RFN03	Interfaz	El sistema tendrá una interfaz fácil de usar, sencilla, amigable, que

		<p>permita que los usuarios sean capaces de interactuar con la aplicación aun con conocimientos básicos de informática. Será diseñada para adaptarse a la resolución del usuario, con la utilización de colores refrescantes, agradables y se emplearán imágenes identificadas con el negocio donde se utilizará el sistema.</p>
RFN04	Rendimiento	<p>El sistema debe desempeñar su función de una manera fluida. Se debe buscar la experiencia de uso más agradable para el usuario,</p>
RFN05	Software	<p>Sistema operativo: Windows XP o superior.</p>
RFN06	Hardware	<p>Procesador Pentium III a 1,33GHz con 256 Mb de memoria RAM.</p>

2.4 Historia de usuario

Tabla 3 Historias de usuario

Orden	HU	Nombre	Cargar documento Excel
Riesgos	Medio	Prioridad	Alta

Iteración	1	Puntos de estimación	0.4
Descripción	El software debe ser capaz de leer los datos de las variables almacenadas en el documento Excel cuyo directorio dentro de la estación de trabajo lo establece el usuario. Los datos presentes en el Excel deben ser mostrados en pantalla en forma de tabla.		
Observación	Los nombres de las variables presentes en el fichero deben ser los nombres de las cabeceras de las columnas de las tablas.		

Descripción de los campos que componen las Historias de Usuario:

Orden: está constituido por dos partes. La primera está referido al nomenclador HU (Historia de Usuario) y la segunda corresponde el número de la funcionalidad que representa.

Nombre: nombre que identificará a la Historia de Usuario.

Riesgo: es el grado de incertidumbre en el desarrollo que se asocia a la Historia de Usuario. Determina la posibilidad real de implementarse o no con las condiciones previstas por el equipo de desarrollo (tiempo, recursos, personal). Puede ser Bajo, Medio o Alto.

Prioridad: la prioridad la define el cliente, y es el grado de importancia que le concede a la funcionalidad.

Iteración: es el número de la fase en la cual se define la Historia de Usuario.

Puntos estimados: es un número entero que representa la cantidad de semanas que se dispone para el desarrollo de la Historia de Usuario. Las Historias de Usuario con altos puntos estimados deben ser separadas en varias tareas. Un punto es una semana efectiva de desarrollo,

Descripción: se escribe una fundamentación de lo que hace la funcionalidad,

Observación: se escribe los elementos o detalles que se deben tener en cuenta para la implementación de la misma.

A partir de la solución propuesta se identificaron 19 requisitos funcionales agrupados en las siguientes Historias de Usuario:

Tabla 4 Historias de usuarios agrupadas por requisitos

Número	Historia de usuario
1.1	Cargar Excel.
2.1	Seleccionar variables.
3.1	Generar gráficos de correlación.
4.1	Adicionar modelo.
4.2	Eliminar modelo.
4.3	Editar modelo.

4.5	Ver detalles de modelo.
5.1	Analizar modelo.
5.2	Prueba de modelo seleccionado Homocedasticidad de residuales.
5.3	Prueba de modelo seleccionado Independencia de residuales.
5.4	Prueba de modelo seleccionado Normalidad de los residuales.
5.5	Prueba de modelo seleccionado Multicolinealidad.
6.1	Prueba de ajuste de bondad de Fisher.
6,2	Prueba de ajuste de bondad de RMSE.
6.3	Prueba de ajuste de bondad de AIC.
6,4	Prueba de ajuste de bondad de BIC.
7.1	Prueba extrapolación oculta de Montgometry.
8.1	Validación del modelo por K-Fold
8.2	Validación del modelo por Bootstroping.

2.5 Estimación de esfuerzo por Historias de Usuario

Las estimaciones de esfuerzo asociado a la implementación de las Historias de Usuario se realizan con el objetivo de lograr una planificación real en el desarrollo del sistema de gestión económica y llevar un registro de la velocidad de desarrollo, basándose principalmente en la suma de puntos correspondientes a las Historias de Usuario.

La planificación se puede realizar basándose en el tiempo. La velocidad de desarrollo es utilizada para establecer cuántas Historias de Usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de éstas. Se establece como medida el punto estimado. Un punto estimado equivale a una semana ideal de programación.

Teniendo en cuenta lo expuesto anteriormente la estimación de esfuerzo de las Historias de Usuario queda como se muestra en la siguiente tabla:

Tabla 5 Estimación de esfuerzo por Historia de Usuario.

Historia de usuario.	Estimación de esfuerzo.
Cargar Excel. Seleccionar variables.	1 Semana.
Generar gráficos de correlación.	1 Semana.
Adicionar modelo.	

Eliminar modelo. Editar modelo. Ver detalles de modelo.	1 Semana.
Analizar modelo.	1 Semana.
Prueba de modelo seleccionado Homocedasticidad de residuales. Prueba de modelo seleccionado Independencia de residuales.	1 Semana.
Prueba de modelo seleccionado Normalidad de los residuales. Prueba de modelo seleccionado Multicolinealidad.	1 Semana.
Prueba de ajuste de bondad de Fisher.	1Semana.
Prueba de ajuste de bondad de RMSE.	1 Semana.
Prueba de ajuste de bondad de AIC. Prueba de ajuste de bondad de BIC.	1 Semana.
Prueba extrapolación oculta de Montgometry.	3 Semanas.
Validación del modelo por K-Fold.	3 Semanas.
Validación del modelo por Bootstroping.	2 Semanas.

A partir de la suma de los puntos de estimación de esfuerzo por cada Historia de Usuario, se calcula que el desarrollo del sistema tendrá una duración de 17 semanas.

2.5.1 Plan de iteraciones

Para lograr una mejor organización del trabajo y proporcionar un desarrollo iterativo e incremental, se crea el plan de iteraciones donde se planifica el orden de desarrollo de las Historias de Usuario. Se definió realizar 6 iteraciones, su orden está determinada según las prioridades de las Historias de Usuario y las dependencias existentes entre ellas. La duración total de cada iteración dependerá de los puntos estimados de las Historias de Usuario que en él se desarrollan.

Tabla 6 Plan de Iteraciones

Iteración.	Historia de usuario.
1	Cargar Excel. Seleccionar variables. Generar gráficos de correlación.

	Adicionar modelo. Eliminar modelo. Editar modelo. Ver detalles de modelo.
2	Analizar modelo. Prueba de modelo seleccionado Homocedasticidad de residuales. Prueba de modelo seleccionado Independencia de residuales. Prueba de modelo seleccionado Normalidad de los residuales. Prueba de modelo seleccionado Multicolinealidad.
3	Prueba de ajuste de bondad de Fisher. Prueba de ajuste de bondad de RMSE. Prueba de ajuste de bondad de AIC. Prueba de ajuste de bondad de BIC.
4	Prueba extrapolación oculta de Montgomery.
5	Validación del modelo por K-Fold.
6	Validación del modelo por Bootstrapping.

2.5.2 Plan de entrega

El plan de entrega es un documento que especifica con exactitud qué Historias de Usuario serán implementadas en cada entrega del sistema y sus prioridades, de modo que también permita conocer con claridad qué Historias de Usuario serán implementadas en la próxima iteración. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo de desarrollado durante las reuniones de planificación de entregas, la idea es hacer entregas frecuentes para obtener una mayor retroalimentación. A continuación se muestra en el plan de entrega definido para el ciclo de desarrollo:

Tabla: Plan de entrega

Iteración.	Historia de usuario.	Fecha de entrega.
1	7	26 de agosto del 2022
2	5	16 de septiembre del 2022
3	4	7 de octubre del 2022
4	1	28 de octubre del 2022
5	1	11 de noviembre del 2022

6	1	25 de noviembre del 2022
---	---	--------------------------

2.5.3 Estimación del costo

Entre los aspectos que no se puede dejar abordar dentro de este capítulo de análisis está el del análisis económico de la solución propuesta. A continuación se realizará un desglose del coste de los elementos necesarios en esta investigación. Dichos elementos incluyen costes de personal, de hardware y de software.

La investigación se realizará entre el 26 de agosto del 2022 al 25 de noviembre de 2022, por lo tanto, han sido 17 semanas de trabajo. Teniendo en cuenta una jornada laboral de 8 horas tendremos un total de 680 horas de trabajo, distribuidas entre diferentes tareas y diferentes roles profesionales que las llevan a cabo.

2..5.4 Coste de personal

La metodología de software escogida propone un equipo de desarrollo pequeño donde cada integrante tiene su rol y funciones bien definidas. Para determinar el coste del personal involucrado se va desglosar el equipo de acuerdo a la categoría de cada uno así como en la fase donde participa quedando el desglose del coste como se aprecia en la siguiente tabla:

Tabla: Coste de personal

Fase.	Categoría.	Horas.	Coste/Horas	Coste
Documentación.	Analista.	61	95.00 cup	5795.00 cup
Análisis.	Analista.	75	95.00 cup	7125.00 cup
Diseño.	Diseñador.	122	70.00 cup	8540 .00cup
Diseño gráfico.	Diseñador gráfico.	48	70.00 cup	3360.00 cup
Implementación.	Programador.	326	50.00 cup	16300,00 cup
Pruebas.	Programador.	48	50.00 cup	2400.00 cup
Total.		680		43520.00 cup

2.5.5 Coste total

A partir del coste de cada uno de los elementos necesarios para la investigación se puede llegar al coste total, como se aprecia en la siguiente tabla:

Tipo de coste.	Total.
Coste personal.	43520.00 cup
Coste del software.	0 cup
Total	43520.00 cup

El costo de desarrollo del sistema fue \$43520.00 aproximadamente, lo que en materia económica constituye una cifra moderada de dinero con respecto a los beneficios que se evidencian a continuación.

Dentro de los beneficios que reporta el software cabe mencionar que, permite a la realizar una predicción rápida y confiable de forma automática del proceso que se analiza. Este software provee una herramienta eficiente que elimina el trabajo manual, disminuye los costos y el tiempo de realización del estudio. La seguridad y protección de los datos se corresponda con el nivel requerido por el cliente y cumple con las expectativas especificadas en los requerimientos funcionales del cliente.

2.6 Conclusiones del capítulo.

Luego de describir el diseño de la solución propuesta al problema científico de esta investigación se concluye que:

1. El levantamiento de los requisitos funcionales y su descripción fue fundamental para lograr un diseño más apropiado para la realización del proyecto.
2. Se proporciona una visión más completa del módulo a desarrollar dado que se modelan todos los procesos que intervienen en el mismo.
3. Se realiza un estudio de factibilidad donde se precisan los beneficios tangibles e intangibles con respecto a los costos de desarrollo de la aplicación y se demuestra la factibilidad de la misma

Capítulo 3: Implementación y prueba

Introducción

En el presente capítulo se abordan las tareas de implementación a través de las Tareas de Ingeniería. Se definen y se describen los estándares de codificación para el desarrollo de la solución propuesta y se confeccionan los casos de prueba y se muestran sus resultados para la validación de la misma.

3.1 Tarea de ingeniería

Tabla 7 Tabla ejemplo de tarea de ingeniería.

Número de tareas: 1	Numero de historias de usuario: H101
Nombre de tarea: Leer archivo Excel	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio:	Fecha fin:
Programador responsable: Hansser J. Díaz Rodríguez	
Descripción: Implementar un mecanismo que permita leer el contenido de un fichero en extensión .xls o .xlsx.	

Los campos de la tarjeta de las Tareas de Ingeniería reflejan lo siguiente:

Número tarea: representa el número por el que se identifica a la tarea. Cada tarea tiene un único número que la identifica

Número Historia de Usuario: es el número de la Historia de Usuario a la que responde la tarea.

Nombre de tarea: define el nombre o funcionalidad concreta a la que se dedica la tarea, debe estar expresado en forma infinitiva'

Tipo de tarea: información del tipo de tarea a realizar, la misma puede ser:

- **Desarrollo:** tarea que se realizará por primera vez,
- **Corrección:** tarea que se realiza a partir de una anterior que no se realizó correctamente, es decir, que no pasó los casos de prueba satisfactoriamente,
- **Mejora:** tarea que se realiza a partir de una anterior incorporándole nuevos requerimientos.
- **Otra:** tarea que no corresponde con una de las anteriores, en este caso es necesario especificar el tipo de tarea o realizar una descripción más profunda de esta.

Puntos estimados: tiempo de duración de la tarea. El tiempo estimado es reflejado en días. La suma de los puntos estimados de las tareas de ingeniería de una Historia de Usuario no puede superar la cantidad de puntos estimados definidos para la Historia de Usuario.

Fecha inicial: fecha en la que se inicia el desarrollo de la tarea de ingeniería,

Fecha final: fecha en la que se concluye el desarrollo de la tarea de ingeniería,

3.2 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es necesario que se sigan ciertos estándares de programación. Para la implementación del sistema se siguieron normas y estándares desarrollados, que se relacionan a continuación:

El código será escrito en inglés y la documentación en español.

Las variables y los nombres de las funciones deben comenzar con letra minúscula, para concatenar otra palabra debe hacerse con letra mayúscula o guión bajo (underscore).

Los nombres de las clases deben comenzar con letra mayúscula, para concatenar otra palabra debe hacerse con letra mayúscula.

3.3 Pruebas de software

Es el proceso de verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa; probando el comportamiento del mismo con el fin proporcionar información objetiva e independiente sobre la calidad del producto. Es una actividad más en el proceso de control de calidad.

3.3.1 Pruebas de aceptación

Casos de prueba

Código: PA1_HU1	Historia de usuario: H101
Nombre: Cargar archivo correcto 1.	
Descripción: Se intentará abrir un archivo de extensión .xls o .xlsx con la información de las variables con la q se quieren construir el modelo.	
Condiciones de ejecución: el software no debe haber cargado un archivo con esta extension con anterioridad.	

Pasos de ejecución: se selecciona el icono q aparece en la esquina superior izquierda de la pantalla y se le da click, luego se localiza el fichero en el equipo y se da click en cargar
Resultados Esperados: se deben cargar los datos contenidos en el fichero y mostrarlos en forma tabulada donde las cabeceras se deben corresponder con cada una de las variables contenidas en el fichero.
Evaluación: Satisfactoria.

Campos del caso de prueba

Código: identificador del caso de prueba. Dividido en dos partes. La primera representa la inicial del artefacto y la segunda representa el número con que se identifica la prueba.

Historia de Usuario: es el número de la Historia de Usuario a la que responde el caso de prueba,

Descripción: es una breve descripción del propósito de la prueba.

Condiciones de ejecución: condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

Entradas / pasos de ejecución: entradas o funciones que deben ejecutarse para realizar el caso de prueba.

Resultado esperado: salida u objetivo que debe cumplir la funcionalidad a la que se le realiza el caso de prueba.

Evaluación: evaluación de éxito del caso de prueba. Prueba satisfactoria en caso de éxito o prueba insatisfactoria en caso de fallo.

Los casos de prueba son agregados a los artefactos de entrega que se realiza al cliente al terminar cada fase o iteración del proyecto. Las Historias de Usuario con evaluación insatisfactoria, serán corregidas en la próxima iteración a partir de nuevas tareas de ingeniería.

3.3.2 Pruebas de satisfacción de usuarios

Otro factor a analizar para saber si el sistema cumple con los requisitos necesarios para el cliente, es realizar pruebas para determinar la satisfacción del usuario. La población utilizada para estas pruebas constaba de un equipo de 5 personas que enviaban comentarios y sugerencias para el sistema. Para este cometido se les ha realizado una encuesta con las siguientes cuestiones:

1. El sistema es fácil de entender para el usuario.
2. La interfaz del sistema es atractiva y amigable.
3. En términos generales, el sistema cumple su cometido y con buen rendimiento.
4. ¿Qué es lo que más le ha gustado del sistema: ¿interfaz, facilidad de uso, funcionalidades o rendimiento?

3.4 Resultados de las pruebas

Pruebas de aceptación

Como resultado de las pruebas de aceptación se detectaron un total de 15 no conformidades. A medida que se avanzó en las iteraciones se disminuyeron los números de no conformidades hasta ser cero. De esta manera el sistema queda listo para ser utilizado. En la siguiente figura se resumen las no conformidades detectadas en cada una de las iteraciones:

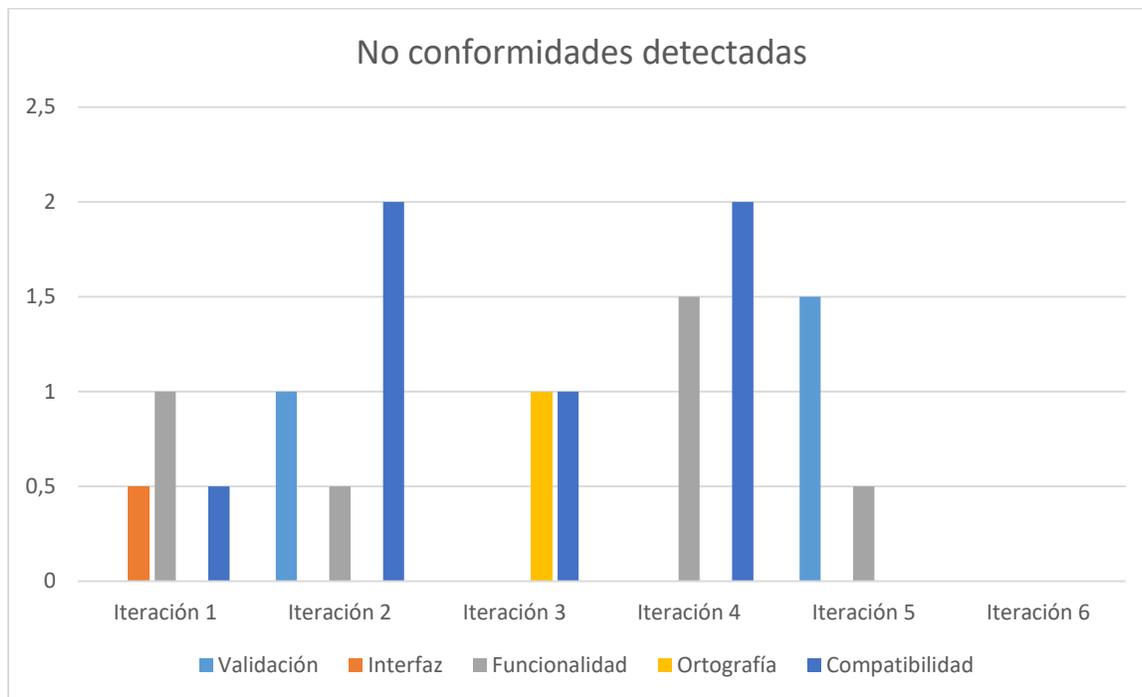


Figura: Resultados de las no conformidades detectadas

Pruebas de satisfacción de usuarios

Para las cuatro primeras pruebas de satisfacción se les solicitó a los usuarios involucrados en estas pruebas que dieran una valoración de 1 a 5. Donde 1 significa estar en desacuerdo y 5 si está completamente de acuerdo.

El sistema es fácil de entender para el usuario

Tabla 8 Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios.

Valoración	Votos	Por ciento
Valoración 1	0	0.0%

Valoración 2	0	0.0%
Valoración 3	0	0.0%
Valoración 4	2	0.4%
Valoración 5	3	0.6%

Como se puede observar la mayoría de los usuarios que probaron el sistema quedaron satisfechos con la disposición de sus funcionalidades y su facilidad de uso.

La interfaz del sistema es atractiva y amigable.

Tabla 9 Resultados de la pregunta 2 de las pruebas de satisfacción de usuarios.

Valoración	Votos	Por ciento
Valoración 1	0	0.0%
Valoración 2	0	0.0%
Valoración 3	0	0.0%
Valoración 4	1	0.2%
Valoración 5	4	0.8%

Como se puede apreciar, los usuarios consideran el diseño de la interfaz de la aplicación atractiva y amigable.

En términos generales, el sistema cumple su cometido y con buen rendimiento.

Tabla 10 Resultados de la pregunta 3 de las pruebas de satisfacción de usuarios.

Valoración	Votos	Por ciento
Valoración 1	0	0.0%
Valoración 2	0	0.0%
Valoración 3	0	0.0%
Valoración 4	1	0.2%
Valoración 5	4	0.8%

Como se puede apreciar, los usuarios están satisfechos con las funcionalidades que brinda el sistema. Se podría decir que se ha obtenido el rendimiento y la cantidad de funcionalidades que todos los usuarios esperaban. Por lo tanto podemos concluir que el sistema cumple con los requisitos no funcionales expuestos en la fase de exploración.

Tabla 11. Resultados de la pregunta 5 de las pruebas de satisfacción de usuarios.

Valoración	Votos	Por ciento
Interfaz	2	0.4%
Facilidad de uso	1	0.2%
Funcionalidades	1	0.2%
Rendimiento	1	0.2%

Como se puede apreciar, la opción más votada fue las funcionalidades, seguido de la facilidad de uso y rendimiento. Gracias a la encuesta se pudo observar que se debe trabajar en mejorar la interfaz del sistema.

3.5 Conclusiones parciales del capítulo

Una vez validada la propuesta de solución al problema científico de esta investigación a partir de la metodología XP, y analizado los resultados de la misma a partir de las pruebas funcionales, se concluye que:

1. Se realizó una descripción detallada del módulo especificando cada uno de sus componentes.
2. Para garantizar una mejor calidad del software fue necesario la realización de las pruebas, las cuales se efectuaron a partir de diferentes métodos, donde los resultados arrojados por las mismas permitieron conocer algunos errores y erradicarlos para así obtener un software de mayor calidad y entregar al cliente un producto que cumpla con sus expectativas.

Conclusiones Generales

Como resultado de esta investigación se dio cumplimiento a los objetivos trazados arribando a las siguientes conclusiones:

1. El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
2. Los sistemas automatizados encontrados, vinculados al tema no le dan solución al problema planteado ya que no permiten la personalización exigida por el cliente.
3. Se realizó la estimación del costo de implementación del sistema y el estudio de factibilidad, arrojando como resultado la factibilidad de la realización del sistema informático.
4. Las pruebas aplicadas al sistema permitieron la detección de errores y la pronta corrección de los mismos.
5. La implementación del sistema y la aplicación de las pruebas de validación con resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados constatándose, mediante avales, su aporte práctico a la gestión empresarial cubana.

Referencias bibliográficas

- Batule, E. (2009). La molienda y difusión de la caña de azúcar. En E. Batule (Ed.), *Serie azucarera 20*: . San Salvador (El Salvador).
- Fox, J., & Sanford, W. (2019). *An R Companion to Applied Regression* (3ra ed.). California: SAGE
- Hugot, E. (2014). 19 - Imbibition. En E. Hugot (Ed.), *Handbook of Cane Sugar Engineering* (pp. 223-245): Elsevier. <https://www.sciencedirect.com/science/article/pii/B9781483231907500283>
- Jenkins, G. H. (2013). *Introduction to Cane Sugar Technology*. Elsevier. <https://www.sciencedirect.com/science/article/pii/B9781483231952500166>
- Kent, G. J. S. t. (2011). The effect of added water temperature on milling train operation and performance. *13*(1), 1-6.
- Montgomery, D., & Runger, G. (2018). *Applied Statistics and Probability for Engineers* (7ma ed.). Estados Unidos: Wiley
- Rein, P. (2012). *Ingeniería de la caña de azúcar*. Bartens
- Serrano, J., & Orozco, J. (2020). *Herramientas de control para el agua de imbibición en centrales azucareros de la provincia de Matanzas*. Presentado en: Conferencia provincial de la filial de la ATAC en Matanzas, Universidad de Matanzas (Matanzas, Cuba).
- Valentín-Reyes, J., García-Reyes, R. B., García-González, A., Álvarez-Valencia, L. H., Rivas-García, P., & Cerino-Córdova, F. d. J. (2018). Mathematical modelling for biohydrogen production by *Clostridium beijerinckii*. *International Journal of Hydrogen Energy*, *43*(37), 17602-17610. doi: <https://doi.org/10.1016/j.ijhydene.2018.07.200>. <http://www.sciencedirect.com/science/article/pii/S0360319918324558>
- Singh, O. K. (2019). Exergy analysis of a grid-connected bagasse-based cogeneration plant of sugar factory and exhaust heat utilization for running a cold storage. *Renewable Energy*, *143*, 149-163. doi: <https://doi.org/10.1016/j.renene.2019.05.012>
- Thaval, O. (2012). Modelling the flow of cane constituents through the milling process of a raw sugar factory. Queensland University of Technology.
- Thaval, O., & Kent, G. (2012). Modelling milling train performance. Presentado en: Proceedings of the 37th Conference of the Australian Society of Sugar Cane Technologists, Bundaberg, Queensland, Australia.

- Thaval, O. and Kent, G. A. 2013. Advanced computer simulation of the milling process. International Society of Sugar Cane Technologists 28: 1595-1607.
- Velásquez, F., Espitia, J., Mendieta, O., Escobar, S., and Rodríguez, J. (2019). Non-centrifugal cane sugar processing: A review on recent advances and the influence of process variables on qualities attributes of final products. Journal of Food Engineering, 255, 32-40. doi: <https://doi.org/10.1016/j.jfoodeng.2019.03.009>
- Villatoro, M. A. S. (2019). Evaluación de mazas perforadas, en el agotamiento del bagazo de caña de azúcar en el proceso de extracción de jugo, en ingenio PAlo Gordo, S.A., Universidad de San Carlos de Guatemala, Guatemala.
- Wickham , H., Chang, W., Henry , L., Lin, T., Takahashi, K., Wilke, C., Dunnington, D. (2021). Package "ggplot2" (Create Elegant Data Visualisations Using the Grammar of Graphics) (Version 3.3.5).
- Wienese, A. (1990). Mill settings and extraction. Presentado en: Proceedings of The South African Sugar Technologists' Association.
- Wienese, A. (1994). Imbibition optimisation at Mount Edgecombe. Proceedings of The South African Sugar Technologists' Association, 6.
- Wienese, A. (1995). The effect of imbibition and cane quality on the front end mass balance. Presentado en: Proc S Afr Sug Technol Ass.