UNIVERSIDAD TRES CULTURAS

PLANTEL CHALCO

INGENIERÍA EN SISTEMAS COMPUTACIONALES

“CONTROL DE SALIDAS DE GARRAFONES DE AGUA”

PROYECTO DEFENSA DE TÉSIS

PARA OBTENER EL TÍTULO EN INGENIERÍA EN SISTEMAS COMPUTACIONALES

P R E S E N T A N

CADENA ESPINOZA BRAULIO

Asesores. Lic. José Diego Martínez Garnelo

Rafael Ramírez Castro

Estado De México, 2024

Índice

1.CAPITULO 1 5
GENERALIDADES DEL PROYECTO 5
1.1 PLANTEAMIENTO DEL PROBLEMA 6
1.2. OBJETIVOS 7
1.2.1 OBJETIVO GENERAL: 7
1.2.1 OBJETIVO ESPECIFICO: 7
1.3. JUSTIFICACIÓN: 8
1.4. ALCANCE: 9
1.5. LIMITACIONES: 9
2.0 CAPÍTULO II 10
FUNDAMENTO TEÓRICO 10
2.1. ESTADO DEL ARTE 11
2.2 GLOSARIO DE TÉRMINOS TÉCNICOS 12
2.3 MARCO TEÓRICO 18
2.4 ESTUDIO DEL MERCADO 55
2.5 VIABILIDAD 59
2.6 FACTIBILIDAD 60
2.7 IMPACTOS 65
2.7.1 SOCIAL 65
2.7.2 ECONÓMICO 65
2.7.3TECNOLOGICO 65
3.0 METODOLOGÍA DE DESARROLLO 66
3.1 METODOLOGIA SCRUM 67
3.2 GRAFICA DE GANTT 71
3.3 FASE DE ANÁLISIS 72
3.3.1 LEVANTAMIENTO DE REQUERIMIENTOS 73
3.3.2 DESCRIPCIÓN DEL PROCESO DE NEGOCIO 76
3.3.4 HISTORIAS DE USUARIO (REQUERIMIENTOS FUNCIONALES) 76
3.3.5 REQUERIMIENTOS NO FUNCIONALES 79
3.3.6 REQUERIMIENTOS DEL SISTEMA 80
3.3.7.1. DISEÑO DE INTERFAZ DE USUARIO 80
3.3.7 ESTUDIO DE USABILIDAD (HERRAMIENTAS DE ACUERDO A LAS NECESIDADES DEL LEVANTAMIENTO) 81

3.4 FASE DISEÑO 82
3.4.1 DIAGRAMA DE PROCESOS GENERAL 83
3.4.2 DIAGRAMA DE CASOS DE USO GENERAL 85
3.4.3 ESPECIFICACIÓN DE CASOS DE USO 86
3.4.4 DIAGRAMA DE CLASES 87
3.4.5 DIAGRAMA DE SECUENCIA 87
3.4.6 DIAGRAMA DE ACTIVIDADES 89
3.4.7 DIAGRAMA ENTIDAD RELACIÓN 90
3.4.8 DIAGRAMA RELACIONAL ..... ¡ERROR! MARCADOR NO DEFINIDO.
3.4.9 SITEMAP (ARQUITECTURA DE INFORMACIÓN) 91
3.5 WILFRAMES 92
3.5.1 LOOK AND FEEL 99
3.5.2 FASE CODIFICACIÓN MÓDULOS DE PROGRAMACIÓN (ESPECIFICACIÓN DE CLASES Y MÉTODOS) 101
3.5.3 FASE PRUEBAS 112
3.5.4 PRUEBAS DEL SISTEMA 113
PRUEBAS DE VALIDACIÓN 113
PRUEBAS DE FUNCIONALIDAD 114
PRUEBAS DE USABILIDAD 116
3.5.5 PRUEBAS DE SEGURIDAD (ESTRÉS) 120
3.5.6 PRUEBAS UNITARIAS 124
3.5.7 PRUEBAS DE INTEGRIDAD 128
4.0 CAPÍTULO 4 IMPLEMENTACIÓN Y RESULTADOS 132
4.1 CONCLUSIONES 133
4.2 GLOSARIO DE TÉRMINOS 134
4.3 REFERENCIAS 136
4.4 ANEXOS 146
4.4.1 ANEXO A MANUAL TÉCNICO 147
4.4.2 ANEXO B MANUAL DE USUARIO 158
4.4.3 ANEXO C CODIFICACIÓN DEL PROGRAMA Y BD 164

INTRODUCCIÓN

Presente proyecto consiste en la creación de un sistema eficiente para el registro y control de los litros iniciales del día y las salidas con envases de 10 a 20 litros en un negocio específico. En la actualidad, se realiza el registro de litros de salida utilizando un sensor de peso, sin embargo, no se cuenta con un sistema que permita realizar un seguimiento y cuadre de los litros registrados en la venta con los litros restantes en el tanque al final del día. Esto puede generar dificultades para verificar la cantidad de agua disponible y ocasionar pérdidas económicas debido a discrepancias en los registros.

Es por ello que se plantea desarrollar un proyecto que comprenda la creación de una tabla que permita el registro adecuado y el control de las salidas de garrafones, utilizando un sensor de peso para registrar los litros de salida. Además, se llevará a cabo un proceso al final del día para verificar los litros finales en el tanque. El objetivo final del proyecto es generar un reporte final que muestre de manera clara y concisa los litros registrados en la venta y los litros restantes en el tanque, ofreciendo así una visión clara del inventario disponible.

1.CAPITULO 1

GENERALIDADES DEL PROYECTO

1.1 Planteamiento del problema

En el estado de México, Avenida paso de cortes manzana 1 lote 8, Colonia cerro del marqués, Valle de Chalco Solidaridad se ha detectado que en la purificadora “San Antonio” existe un deficiente control de registro de salida de garrafones de agua, lo cual genera que al final del día no se lleve un registro exacto de los garrafones que salieron a la venta y por consecuencia no coincidan las ganancias monetarias.

1.2. OBJETIVOS

1.2.1 Objetivo general:

Generar un control de salida de garrafones de 10 y 20 litros, usando un sensor de peso para registrar la salida de garrafones, para al final del día tener un registro exacto de los garrafones que salieron de la purificadora.

1.2.1 Objetivo Especifico:

- Crear una interfaz de registro utilizando (visual Basic, Excel, c++) con ayuda de hardware (Arduino Uno) que lleve el control de salida de garrafones de 10 y 20 litros, obteniendo los totales de garrafones vendidos.

- Analizar el control interno de la purificadora sobre los registros de salida de los garrafones de agua mediante entrevistas técnicas con los encargados directos para poder sistematizar el proceso de registro de salida de los garrafones.

- Desarrollar un sistema que permita registrar de forma digital la salida de cada garrafón, incluyendo ID, fecha, hora de salida y peso del garrafón.

- Capacitar a todo el personal involucrado en la venta de garrafones sobre el uso correcto del nuevo sistema de registro.

1.3. JUSTIFICACIÓN:

La realización de este proyecto se basa en la necesidad de llevar un buen registro de garrafones de agua en la purificadora “SAN ANTONIO”, por lo cual el negocio requiere de un sistema sistematizado que le permita el registro de los garrafones que salen de la purificadora, es por ello, que dicho negocio ha tenido dificultades para saber con exactitud cuántos garrafones salieron durante el día.

1.4. Alcance:

- Solo se registrará garrafones de 10 y 20 litros
- Solo se registra las salidas de los garrafones en la venta.
- El hardware solo será un prototipo de corta duración
- El uso de este sistema es solo con fines de llevar un registro sistemático de salida de litros de agua.

1.5. limitaciones:

- El acrílico y la madera utilizados para el soporte y la base tienen ciertas propiedades mecánicas que pueden afectar la durabilidad
- **Vida Útil del Sensor**: Con el uso continuo, la galga extensiométrica puede experimentar fatiga, lo que podría alterar sus propiedades y, en consecuencia, afectar la precisión de las mediciones.

2.0 CAPÍTULO II

FUNDAMENTO TEÓRICO

2.1. ESTADO DEL ARTE

En este apartado, se expone características de algunos sistemas web que son similares al sistema a desarrollar, dando referencias con base a funcionalidad y a características.

A continuación, se muestran algunos de los sistemas web que fueron consideradas como referencia:

- Balanza con galga extensiométrica
  https://es.scribd.com/document/442213834/PROYECTO-GALGA

  Este documento describe el diseño de una balanza que utiliza una celda de carga de 5 kg con una galga extensiométrica. Explica el principio de funcionamiento de las galgas extensiométricas, los materiales utilizados y el proceso de programar un Arduino y para mostrar los datos de la galga en una computadora. El objetivo es comprender cómo funcionan las galgas extensiométricas y crear una balanza digital programable.

- "Control de fuerza ejercida en dispositivos tensores diseñados para aplicaciones biomédicas"

  https://core.ac.uk/download/pdf/18429415.pdf

  El objetivo de este proyecto es poder medir, procesar y monitorizar la compresión que sufren una serie de tensores diseñados para aplicaciones biomédicas.

- "DISEÑO DE UN DISPOSITIVO PARA MEDIR LA FUERZA DE CORTE EN EL TORNEADO DEL ACERO INOX. 304"

  https://dspace.unitru.edu.pe/server/api/core/bitstreams/82fcc722-cf90-401c-90f9-1bc2087af1e2/content

Se realizó ante la falta de equipos sofisticados para medir esa fuerza. Se optó por realizar un dispositivo a menor costo, con los cuales se obtuvieron buenos resultados.

El trabajo fue realizado en un torno mecánico, con el apoyo de la electrónica básica utilizando un Arduino UNO R3 el cual fue conectado con un amplificador HX711 y a su vez, dicho amplificador estuvo conectado a las galgas extensiométricas las mismas que mediante un Ciano acrilato (adhesivo) se instaló.

2.2 GLOSARIO DE TÉRMINOS TÉCNICOS

A.

Arduino UNO: Arduino Uno es una placa de microcontrolador de código abierto basada en el microcontrolador ATmega328P. Está diseñada para crear prototipos de forma fácil y accesible, lo que la convierte en la herramienta perfecta tanto para principiantes como para expertos.

ATmega328: El ATmega328 es un microcontrolador basado en la arquitectura AVR de 8 bits. Posee una velocidad de reloj de hasta 20 MHz y cuenta con 32 KB de memoria Flash, lo que permite almacenar programas y datos de manera eficiente.

Automatización: La automatización es la aplicación de tecnología para reducir la participación humana en procesos, tareas o decisiones.

B.

Báscula: Una báscula es un dispositivo diseñado para medir el peso de un objeto. Funciona mediante principios de fuerza y equilibrio. Cuando colocas un objeto en la plataforma de la báscula, ejerce una fuerza hacia abajo, y esta fuerza se traduce en una lectura numérica que representa su peso

Bootloader: El bootloader (o gestor de arranque) es un componente fundamental en Android y otros sistemas operativos. Antes de cargar el sistema operativo, el bootloader realiza varias comprobaciones para asegurarse de que todo está en su sitio y da las instrucciones necesarias para el inicio.

C.

C++: C++ es un lenguaje de programación que se originó como una extensión del lenguaje C para permitir la manipulación de objetos. Aunque tiene muchos años, su gran potencia lo convierte en uno de los lenguajes de programación más demandados.

Código: Se denomina código fuente al conjunto de líneas de texto que expresan, en un lenguaje de programación determinado, los pasos que debe seguir el computador para la correcta ejecución de un programa específico.

E.

Eficiencia: La eficiencia es la capacidad de disponer de alguien o algo para conseguir el cumplimiento adecuado de una función. En otras palabras, se refiere a lograr un resultado optimizando el uso de los recursos.

Entrada serial RX: se refiere al pin de recepción en una comunicación serial. En Arduino, este pin se utiliza para recibir datos desde otros dispositivos o desde el ordenador. Por lo general, está conectado al pin digital 0 (RX) en la placa Arduino

Excel: Excel es un programa informático desarrollado por Microsoft y forma parte de Office, una suite ofimática que incluye otros programas como Word y PowerPoint. En esencia, Excel nos permite formatear, organizar y calcular datos en una hoja de cálculo.

G.

Galga extensiométrica: Una galga extensiométrica, también conocida como extensómetro, es un sensor que convierte la deformación de un material en un cambio de resistencia eléctrica. En otras palabras, mide cómo se estira o se comprime un objeto y lo traduce en una variación en su resistencia eléctrica

GND: la abreviatura GND se refiere a la conexión a tierra utilizada para las señales en un sistema electrónico. La GND de señal es una referencia común que se utiliza para establecer el nivel de voltaje cero para las señales en un circuito o dispositivo electrónico.

H.

Hardware: Hardware es la parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y

mecánicos, como circuitos, placas, memorias, discos duros y dispositivos periféricos. En resumen, es todo aquello que necesitamos para que el equipo funcione.

HX711: es un amplificador de célula de carga que permite medir con precisión el peso utilizando una celda de carga

I.

Integración: La integración en programación se refiere a la conexión o interacción entre diferentes componentes, sistemas o aplicaciones informáticas.

Interfaz: En informática, una interfaz es la conexión funcional entre dos sistemas, programas, dispositivos o componentes. Sirve para facilitar la comunicación y el intercambio de información entre ellos.

Interfaz física: Conecta dispositivos electrónicos (como puertos USB) para que se comuniquen entre sí.

Interfaz lógica: Permite la comunicación entre programas o software, como los protocolos informáticos.

Interfaz de usuario: Es el espacio donde ocurren las interacciones entre máquina y usuario. Incluye componentes físicos como teclados y ratones, así como software que permite manejar procesos complejos

M.

Memoria: la memoria es el almacén de los valores generados al ejecutar un programa. Imagina la memoria como un conjunto de celdas, cada una con una dirección asociada. Cada variable en un programa tiene un lugar específico en esta memoria.

Memoria flash: La memoria Flash en Arduino es donde se almacena el sketch (nombre que Arduino utiliza para un programa). Es la unidad de código que se sube y ejecuta en la placa Arduino. Esta memoria es no volátil, lo que

significa que los datos permanecen incluso si Arduino deja de recibir alimentación eléctrica. El tamaño de la memoria Flash varía según el microcontrolador utilizado. Por ejemplo, en el caso del ATmega328 (como el que incorpora Arduino UNO), la memoria Flash tiene un tamaño de 32 KB, de los cuales el bootloader utiliza 0.5 KB.

Microcontrolador: Un microcontrolador es un pequeño ordenador integrado en un único circuito integrado. Su función principal es controlar dispositivos electrónicos. Imagina que es el "cerebro" del dispositivo: recibe entradas de sensores u otros dispositivos, procesa esos datos y envía señales para controlar el comportamiento del dispositivo

P.

Pines analógicos/digitales (A0 a A5): Estos pines pueden funcionar tanto en modo analógico como en modo digital. Puedes usarlos para entrada o salida, según tus necesidades. Por defecto, se configuran como pines analógicos.

Pines de referencia analógico (A0 a A5): Estos pines se utilizan para medir señales analógicas. Puedes conectar sensores, potenciómetros u otros dispositivos que generen voltajes variables. La lectura en estos pines se expresa como un valor entre 0 y 1023 (10 bits de resolución).

Pines digitales (D0 a D13): Estos pines solo pueden manejar señales digitales (0 o 1). Son ideales para conectar botones, Leds, relés y otros dispositivos que requieren encendido/apagado.

Precisión: La precisión en programación se refiere a la exactitud y confiabilidad con la que se realizan los cálculos y se manejan los datos numéricos en un programa. En términos simples, se trata de la capacidad de un programa para obtener resultados correctos y consistentes

Proceso: En informática, un proceso es básicamente un programa en ejecución. Imagina que es como una secuencia de instrucciones que persigue llegar a un estado final o realizar una tarea concreta.

Proceso de registro: un registro es una forma de almacenar elementos de manera ordenada. Se utiliza para agrupar datos relacionados entre sí, aunque sean de diferentes tipos. Cada elemento dentro de un registro se denomina campo.

Programa: Se trata de aplicaciones y recursos que permiten desarrollar diferentes tareas en una computadora (ordenador), un teléfono u otros equipos tecnológicos.

Programación: Es el proceso de creación de programas informáticos. Esta definición se puede interpretar de la siguiente manera. La programación no es más que una explicación a la computadora de qué, en qué forma y cómo llegar al usuario.

Puertos de entrada/salida (E/S): Los puertos de entrada y salida (E/S) son interfaces físicas que permiten la comunicación entre una computadora y sus dispositivos periféricos. Estos puertos actúan como puntos de conexión, donde los dispositivos pueden ingresar o recibir información de la computadora.

R.

Regulador de voltaje: Un regulador de voltaje es un dispositivo electrónico que mantiene constante el valor del voltaje en su salida, independientemente de las variaciones en el voltaje de entrada.

S.

Salida serial RX: se refiere al pin de recepción en una comunicación serial. En Arduino, este pin se utiliza para recibir datos desde otros dispositivos o desde el ordenador. Por lo general, está conectado al pin digital 0 (RX) en la placa Arduino.

Sensor: Un sensor es un dispositivo capaz de medir magnitudes físicas o químicas y transformarlas en señales eléctricas. En otras palabras, los sensores pueden leer variables como temperatura, distancia, fuerza, presión, humedad, entre otras.

Sensor de peso: Un sensor de peso es un dispositivo que mide la fuerza gravitatoria que actúa sobre un objeto y la convierte en una señal eléctrica. Estos sensores se utilizan para medir la masa o el peso de objetos y pueden ser aplicados en diversas áreas, como automatización industrial, control de inventario, sistemas de pesaje y más

U.

Usuario: Un usuario es una persona que utiliza una computadora o un servicio de red. Los usuarios de sistemas informáticos y productos de software generalmente carecen de la experiencia técnica necesaria para comprender completamente cómo funcionan.

V.

Visual Basic: Visual Basic (VB) es un lenguaje de programación dirigido por eventos desarrollado por Microsoft. Fue creado por Alan Cooper en 1991 con la intención de simplificar la programación mediante un ambiente de desarrollo.

Vin: El pin Vin en Arduino es una de las entradas de energía. “Vin” significa “voltaje de entrada”, y este pin permite suministrar energía a la placa utilizando una fuente de alimentación externa.

2.3 MARCO TEÓRICO

Gestionar la investigación que va a fundamentar el proyecto, el sistema lleva como base sistematizar el registro de salidas de garrafones.

Para el desarrollo del sistema del negocio SAN ANTONIO, las herramientas a ocupar son:

- Excel
- Visual Basic
- C++
- Arduino
- Sensor de peso de 20 kg
- Concepto de purificadora
- Física de conversión de litraje

EXCEL

Imagen 1.2. Logo Excel

Excel es una potente aplicación de hoja de cálculo desarrollada por Microsoft que permite a los usuarios crear, organizar y analizar datos en formato de tabla.

detalles interesantes:

Organización de datos: Excel muestra una gran cuadrícula donde cada recuadro es llamado celda, y estas se ordenan en filas y columnas. Los datos se colocan dentro de estas celdas.

Funciones y cálculos: A diferencia de un procesador de palabras, en Excel puedes realizar cálculos aritméticos básicos y aplicar funciones matemáticas más complejas. También puedes utilizar funciones estadísticas.

Reportes y visualización: Excel facilita el análisis de datos y la generación de reportes mediante herramientas como gráficos y tablas dinámicas.

Usos comunes:

- Análisis de negocios
- Gestión de recursos humanos
- Contabilidad y elaboración de presupuestos
- Administración de proyectos
- Informes de rendimiento

Control de Peso e Indice de Masa corporal

Altura: 169 cm Peso Ideal 52,8 - 71,1

Fecha	Peso	Variación	Cintura(cm)	I.M.C	Desc IMC
01/12/2018	54 kg	--		18,9	Peso Normal
02/12/2018	55 kg	+ 1 Kg		19,2	Peso Normal
03/12/2018	51 kg	- 4 Kg		17,9	Peso insuficiente
04/12/2018	57 kg	+ 6 Kg		20,0	Peso Normal
05/12/2018	65 kg	+ 8 Kg		22,8	Peso Normal
06/12/2018	60 kg	- 5 Kg		21,0	Peso Normal
07/12/2018	55 kg	- 5 Kg		19,3	Peso Normal
08/12/2018	51 kg	- 4 Kg		17,9	Peso insuficiente
09/12/2018	50 kg	- 1 Kg		17,5	Peso insuficiente
10/12/2018	49 kg	- 1 Kg		17,2	Peso insuficiente
11/12/2018	51 kg	+ 2 Kg		17,9	Peso insuficiente
12/12/2018	55 kg	+ 4 Kg		19,3	Peso Normal
13/12/2018	77 kg	+ 22 Kg		27,0	Sobrepeso grado I
14/12/2018	89 kg	+ 12 Kg		31,2	Obesidad de tipo I
15/12/2018	85 kg	- 4 Kg		29,8	Sobrepeso grado II (preobesidad)

Imagen 2.2 Ejemplo de tabla de registro con Excel.

Las especificaciones y límites de Excel son bastante amplios y detallados.

En cuanto a los libros abiertos, están limitados por la memoria disponible y los recursos del sistema. Una hoja de cálculo puede contener hasta 1.048.576 filas y 16.384 columnas. El ancho de una columna puede ser de hasta 255 caracteres, mientras que el alto de una fila puede alcanzar los 409 puntos. Los saltos de página están limitados a 1.026 tanto horizontal como verticalmente. Una celda puede contener hasta 32.767 caracteres, y los encabezados o pies de página pueden tener hasta 255 caracteres. El número máximo de saltos de línea por celda es de 253.

En cuanto a las hojas en un libro, están limitadas por la memoria disponible, siendo el valor predeterminado una hoja. Un libro puede contener hasta 16 millones de colores (32 bits con acceso completo al espectro de colores de 24 bits). Las vistas con nombre en un libro también dependen de la memoria disponible. Los formatos o estilos de celdas distintos están limitados a 65.490, mientras que los estilos de relleno y los grosores y estilos de línea están limitados a 256 cada uno. Un libro puede tener hasta 1.024 fuentes globales disponibles, con un máximo de 512 para cada libro. Los formatos de número en un libro varían entre 200 y 250, dependiendo del idioma de la versión de Excel instalada.

Los nombres en un libro y las ventanas en un libro también dependen de la memoria disponible. Una hoja de cálculo puede contener hasta 65.530 hipervínculos y una ventana puede tener hasta 4 paneles. Las hojas vinculadas y los escenarios también están limitados por la memoria disponible, con un informe de resumen que solo muestra los primeros 251 escenarios. Las celdas cambiantes en un escenario están limitadas a 32, y las celdas ajustables en Solver a 200. Las funciones personalizadas dependen de la memoria disponible, al igual que los informes y los criterios de ordenación, que pueden combinar hasta 64 en una única operación.

El número de niveles de deshacer es de 100, y un formulario puede tener hasta 32 campos. Un libro puede tener hasta 255 parámetros y las listas de filtros desplegables pueden mostrar hasta 10.000 elementos. Las celdas discontinuas que pueden seleccionarse están limitadas a 2.147.483.648. En cuanto a los límites máximos de almacenamiento en memoria y tamaños de archivos para libros del modelo de datos, el entorno de 32 bits está sujeto a una limitación de 2 gigabytes (GB) de espacio de direcciones virtuales, compartido por Excel, el libro y los complementos que se ejecuten en el mismo proceso. La parte de dirección del modelo de datos podría alcanzar de 500 a 700 megabytes (MB), pero podría ser inferior si se cargan otros modelos de datos y complementos. El entorno de 64 bits no impone límites estrictos al tamaño del archivo, estando limitado solo por la memoria disponible y los recursos del sistema. A partir de Excel 2016, la función Detección de direcciones largas permite que Excel de 32 bits consuma el doble de memoria cuando los usuarios trabajan en un sistema operativo Windows de 64 bits.

En cuanto a los cálculos, la precisión numérica es de 15 dígitos. El número negativo más bajo permitido es -2,2251E-308 y el número positivo más bajo permitido es 2,2251E-308. El número positivo más alto permitido es 9,99999999999999E+307, mientras que el número negativo más alto permitido es -9,99999999999999E+307. El número positivo más alto permitido mediante fórmula es 1,7976931348623158e+308 y el número negativo más alto permitido mediante fórmula es -1,7976931348623158e+308. La longitud del contenido de una fórmula puede ser de hasta 8.192 caracteres y la longitud interna de la fórmula puede ser de hasta 16.384 bytes.

Excel tiene varias especificaciones y límites importantes. En cuanto a los núcleos del procesador, se permite un máximo de 64. La longitud del nombre de archivo puede ser de hasta 218 caracteres, incluyendo la ruta de acceso del archivo (por ejemplo, C:\Username\Documents\FileName.xlsx).

Especificaciones y Límites de los Cálculos

La precisión numérica en Excel es de 15 dígitos. El número negativo más bajo permitido es -2,2251E-308, mientras que el número positivo más bajo permitido es 2,2251E-308. El número positivo más alto permitido es 9,99999999999999E+307 y el número negativo más alto permitido es -9,99999999999999E+307. Mediante fórmula, el número positivo más alto permitido es 1,7976931348623158e+308 y el número negativo más alto permitido es -1,7976931348623158e+308. La longitud del contenido de una fórmula puede ser de hasta 8.192 caracteres y la longitud interna de la fórmula puede ser de hasta 16.384 bytes. Las iteraciones están limitadas a 32.767.

En cuanto a las matrices en una hoja, están limitadas por la memoria disponible. Los rangos seleccionados pueden ser hasta 2.048 y los argumentos en una función pueden ser hasta 255. Los niveles anidados de funciones están limitados a 64 y las categorías de funciones definidas por el usuario a 255. Excel tiene 341 funciones de hoja de cálculo disponibles. El tamaño de la pila de operandos es de 1.024. Hasta 64.000 hojas de cálculo pueden hacer referencia a otras hojas. La dependencia de fórmulas de matriz entre hojas de cálculo, la dependencia de áreas y la dependencia de áreas en cada hoja de cálculo están limitadas por la memoria disponible. Hasta 4.000 millones de fórmulas pueden depender de una sola celda. La longitud del contenido de celdas vinculadas de libros cerrados puede ser de hasta 32.767 caracteres. La primera fecha permitida en un cálculo es el 1 de enero de 1900 (o el 1 de enero de 1904 si se utiliza la fecha del sistema), y la última fecha permitida es el 31 de diciembre de 9999. El período de tiempo máximo que puede escribirse es 9999:59:59.

Especificaciones y Límites de los Informes de Tablas y Gráficos Dinámicos

En una hoja, los informes de tabla dinámica están limitados por la memoria disponible. Cada campo puede tener hasta 1.048.576 elementos únicos. Los campos de fila o columna en un informe de tabla dinámica están limitados por la memoria disponible, al igual que los filtros de informe, que pueden ser hasta 256. Los campos de valores en un informe de tabla dinámica pueden ser hasta 256, y las fórmulas de elementos calculados están limitadas por la memoria disponible. Los filtros de informe en un informe de gráfico dinámico también pueden ser hasta 256, al igual que los campos de valores. Las fórmulas de elementos calculados en un informe de gráfico dinámico están limitadas por la memoria disponible.

La longitud del nombre MDX para un elemento de tabla dinámica puede ser de hasta 32.767 caracteres, y la longitud de una cadena de tabla dinámica relacional también puede ser de hasta 32.767 caracteres. Las listas de filtros desplegables pueden mostrar hasta 10.000 elementos.

Libros con la Configuración "Permitir que más de un Usuario Pueda Hacer Cambios…" Activada

Si la configuración "Permitir que más de un usuario pueda hacer cambios…" está activada para un libro, se aplican los siguientes límites. Hasta 256 usuarios pueden abrir el archivo al mismo tiempo. Las vistas personales en el libro están limitadas por la memoria disponible. El historial de cambios puede mantenerse hasta por 32.767 días (el valor predeterminado es 30 días). Los libros que pueden combinarse a la vez están limitados por la memoria disponible.

VISUAL STUDIO CODE

¿Qué es Visual Studio Code?

Visual Studio Code es un editor de código fuente gratuito, de código abierto y multiplataforma desarrollado por Microsoft. Se destaca por su ligereza, velocidad y extensibilidad, lo que lo convierte en una herramienta ideal para una amplia gama de lenguajes de programación.

Imagen 3.2 Logo Visual Basic.

Características principales:

- Multiplataforma: Disponible para Windows, macOS y Linux.
- Extensible: Ampliable mediante extensiones para agregar nuevas funcionalidades y soporte para lenguajes.
- Inteligencia de código: Ofrece autocompletado, refactorización y depuración inteligentes.
- Git integrado: Permite gestionar repositorios Git directamente desde el editor.
- Personalizable: Permite configurar temas, atajos de teclado y otros aspectos de la interfaz.
- Debugger integrado: Facilita la identificación y corrección de errores en el código.

Visual Studio Code (VS Code) es un editor de código fuente muy popular y versátil. A continuación, te presento una investigación sobre sus características, ventajas y algunas referencias en formato APA:

Visual Studio Code: Un Editor de Código Potente y Personalizable

¿Qué es Visual Studio Code?Visual Studio Code es un editor de código fuente gratuito, de código abierto y multiplataforma desarrollado por Microsoft. Se destaca por su ligereza, velocidad y extensibilidad, lo que lo convierte en una herramienta ideal para una amplia gama de lenguajes de programación.

Características principales:

- Multiplataforma: Disponible para Windows, macOS y Linux.
- Extensible: Ampliable mediante extensiones para agregar nuevas funcionalidades y soporte para lenguajes.
- Inteligencia de código: Ofrece autocompletado, refactorización y depuración inteligentes.
- Git integrado: Permite gestionar repositorios Git directamente desde el editor.
- Personalizable: Permite configurar temas, atajos de teclado y otros aspectos de la interfaz.
- Debugger integrado: Facilita la identificación y corrección de errores en el código.

Ventajas de Visual Studio Code:

- Ligero y rápido: Arranca rápidamente y es eficiente en el uso de recursos.
- Gran comunidad: Cuenta con una gran comunidad de desarrolladores que crean y comparten extensiones.
- Integración con otras herramientas: Se integra fácilmente con otras herramientas de desarrollo, como Docker y Kubernetes.
- Actualizaciones frecuentes: Microsoft lanza actualizaciones regulares con nuevas características y mejoras.

Usos comunes:

- Desarrollo web (JavaScript, TypeScript, HTML, CSS)
- Desarrollo de aplicaciones móviles (React Native, Flutter)
- Desarrollo de aplicaciones de escritorio (Electron)
- Desarrollo de juegos (C++)
- Ciencia de datos (Python, R)

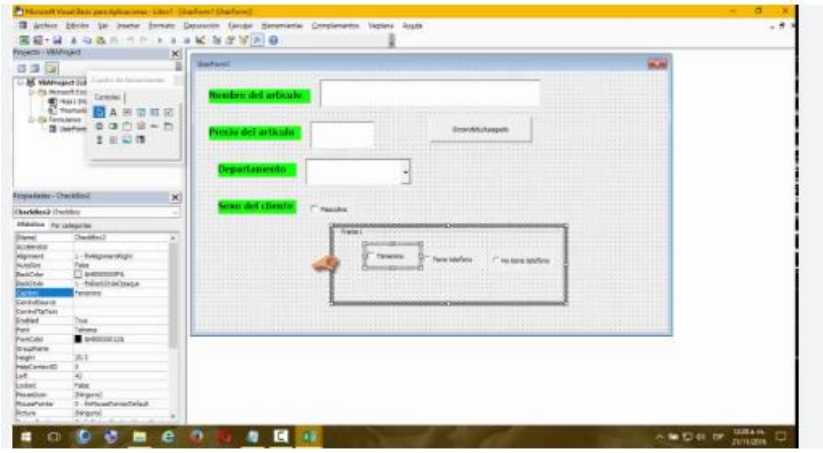


Imagen 4.2 Ejemplo de interfaz de Visual Basic con Excel.

El círculo cromático

Esto quizás lo hayas visto en clase de arte, o si no, al menos quizás estés familiarizado con su forma: los colores primarios rojo, azul y amarillo. Para este punto, usaremos el círculo tradicional de 12 colores, el que usan la mayoría de los pintores y otros artistas, pues es una forma sencilla de visualizar y de entender la relación que tienen los colores entre sí.

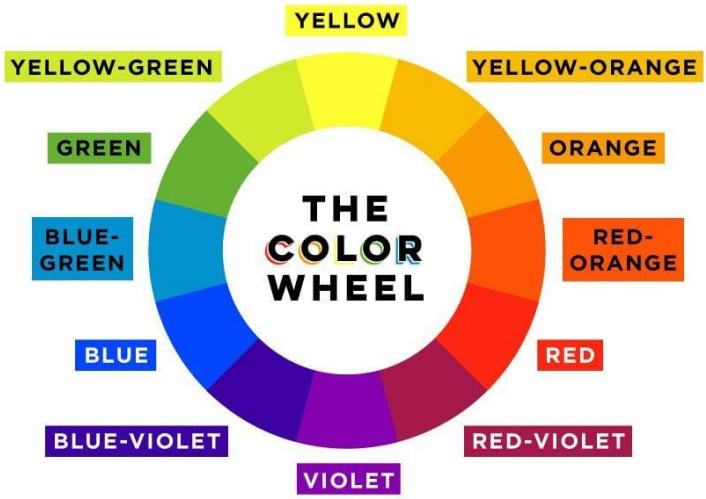


Imagen 5.2 rueda cromática.

En sí la rueda de colores o círculo cromático es una herramienta que se basa, o cuyo objetivo, es la combinación de los colores. Al mezclar los colores base o primarios mencionados anteriormente, obtienes los secundarios: anaranjado, verde y violeta y si mezclas esos con un color primario, obtienes el tercer nivel de colores o mejor dicho, colores terciarios. Estos incluyen rojo-anaranjado, amarillo-anaranjado, amarillo verdoso, azul verdoso, azul y rojo violáceos. Los colores primarios y secundarios (agregando además el índigo) también son parte del espectro visible de la luz o los “colores del arcoíris”.

Esta forma de entender los colores se conoce como síntesis sustractiva de colores, que comprende mezclar colores pigmento como pinturas o tintes y es la categoría en la que se clasifican tanto la rueda tradicional de colores como el sistema de colores CMYK que los equipos de impresión usan.

El opuesto de este modelo es la síntesis aditiva de colores, el cual comprende la mezcla del color luz (así como los colores que ves en la pantalla de tu TV o computadora) y utiliza un conjunto distinto de colores primarios: rojo, verde y azul, abreviado frecuentemente RGB, por el nombre de los colores en inglés (red, green, blue).

En sí la rueda de colores o círculo cromático es una herramienta que se basa, o cuyo objetivo, es la combinación de los colores. Al mezclar los colores base o primarios mencionados anteriormente, obtienes los secundarios: anaranjado, verde y violeta y si mezclas esos con un color primario, obtienes el tercer nivel de colores o mejor dicho, colores terciarios. Estos incluyen rojo-anaranjado, amarillo-anaranjado, amarillo verdoso, azul verdoso, azul violáceo y rojo violáceo. Los colores primarios y secundarios (agregando además el índigo) también son parte del espectro visible de la luz o los “colores del arcoíris”.
Esta forma de entender los colores se conoce como síntesis sustractiva de colores, que comprende mezclar colores pigmento como pinturas o tintes y es la categoría en la que se clasifican tanto la rueda tradicional de colores como el sistema de colores CMYK que los equipos de impresión usan. El opuesto de este modelo es la síntesis aditiva de colores, el cual comprende la mezcla del color luz (así como los colores que ves en la pantalla de tu TV o computadora) y utiliza un conjunto distinto de colores primarios: rojo, verde y azul, abreviado frecuentemente RGB, por el nombre de los colores en inglés (red, green, blue).

La psicología del color

El color nos rodea y juega un gran papel en nuestras, nos demos cuenta de ello o no. ¿Esa luz del amarillo, a veces anaranjada, del semáforo? Te llamo la atención por una razón. ¿La caja de cereal que compraste en el mercado, aun cuando era más cara que otras? Quizás te viste atraído por los colores del envase. Incluso, hasta en el lenguaje encontramos al color... ¿o es qué las personas dicen que “ven rojo” o se “sienten azul” sin razón alguna? No, porque el color tiene una conexión única con nuestros ánimos y emociones.

El significado de los colores

El significado del color azul

El color del cielo y el mar, el azul es un color que con frecuencia transmite sensaciones de paz y pureza, y a diferencia de otros colores más energéticos, el color azul es más calmante. Por otro lado, en algunos contextos puede representar tristeza o depresión.

***Significados alternativos**:* Tradicionalmente, en algunas culturas del medio oriente, el azul ha sido la representación de la protección contra el mal, así como también por su asociación con el cielo, el azul es un símbolo de inmortalidad y/o espiritualidad en muchas culturas.

***En branding**:* El azul es uno de los colores más comunes y versátiles. Generalmente se usa para transmitir confianza, seguridad y estabilidad, mientras que el azul marino u oscuro es muy popular en ambientes corporativos, dado que se percibe como un color serio, conservador y que representa cualidades profesionales.

El significado del color blanco

El color de la luz y la nieve, el color blanco con frecuencia representa pureza, inocencia, bondad o perfección (y es el color tradicional de las novias). Sin embargo, el blanco también puede verse como estéril o austero.

***Significados alternativos**:* En China, el blanco es el color del duelo. En muchas culturas representa la paz y universalmente, una bandera blanca significa tregua o rendición.

***En branding**:* Lo que el blanco comúnmente transmite es simpleza o atributos modernos y limpios. Los diseñadores, cuando quieren una estética minimalista, usan mucho blanco.

C++

C++ es un lenguaje de programación de propósito general que se utiliza principalmente para el desarrollo de software de sistemas, aplicaciones de escritorio, juegos y software de alto rendimiento. Es conocido por su potencia, eficiencia y flexibilidad, lo que lo convierte en una opción popular para una amplia gama de aplicaciones.

Imagen 6.2 Logo C++

¿Cuáles son las características principales de C++?

- **Programación orientada a objetos (POO):** C++ es un lenguaje de POO maduro que permite a los desarrolladores crear programas organizados en módulos reutilizables llamados clases. Esto facilita el desarrollo y mantenimiento de código complejo.
- **Alto rendimiento:** C++ ofrece un control cercano al hardware, lo que lo hace ideal para aplicaciones que requieren un rendimiento rápido y eficiente. Esto lo convierte en una buena opción para juegos, software de finanzas y otras aplicaciones sensibles al tiempo.
- **Flexibilidad:** C++ admite una amplia gama de estilos de programación, desde la programación estructurada hasta la genérica y funcional. Esto lo hace adaptable a una variedad de proyectos y necesidades.
- **Amplia biblioteca estándar:** C++ viene con una biblioteca estándar extensa que proporciona funciones integradas para tareas comunes como entrada/salida, manipulación de cadenas y algoritmos.
- **Portabilidad:** C++ es un lenguaje de programación portátil, lo que significa que el código escrito en C++ se puede compilar y ejecutar en diferentes plataformas con pocos o ningún cambio.

¿Para qué se utiliza C++?

Aquí hay algunos ejemplos específicos de cómo se usa C++:

- **Sistemas operativos:** C++ es un lenguaje fundamental para el desarrollo de sistemas operativos como Windows, Linux y macOS.
- **Aplicaciones de escritorio:** C++ se utiliza para crear una amplia gama de aplicaciones de escritorio, como navegadores web, editores de texto y suites ofimáticas.
- **Juegos:** C++ es uno de los lenguajes de programación más populares para el desarrollo de videojuegos, desde juegos simples hasta títulos AAA de alta gama.
- **Software de alto rendimiento:** C++ se utiliza en software de alto rendimiento como aplicaciones científicas, de ingeniería y financieras.
- **Aplicaciones embebidas:** C++ es una buena opción para desarrollar software para sistemas embebidos, como microcontroladores y dispositivos de Internet de las cosas (IoT).
- **Bibliotecas y compiladores:** C++ incluso se utiliza para escribir bibliotecas y compiladores para otros lenguajes de programación.

En resumen, C++ es un lenguaje de programación poderoso y versátil que se utiliza para crear una amplia gama de aplicaciones. Su rendimiento, flexibilidad y portabilidad lo convierten en una opción popular para desarrolladores de todo el mundo.

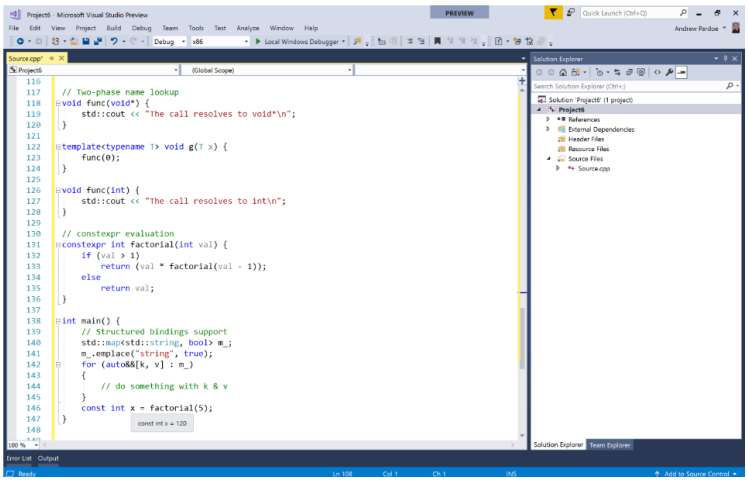


Imagen 7.2 Ejemplo de C++ en Visual Basic.

Arduino uno

Arduino Uno es una placa de microcontroladores de código abierto basada en el microcontrolador ATmega328P. Es una de las placas Arduino más populares y fáciles de usar, lo que la convierte en una opción ideal para principiantes y aficionados a la electrónica. Arduino Uno se puede utilizar para crear una amplia gama de proyectos, desde simples controles de luces hasta robots complejos.

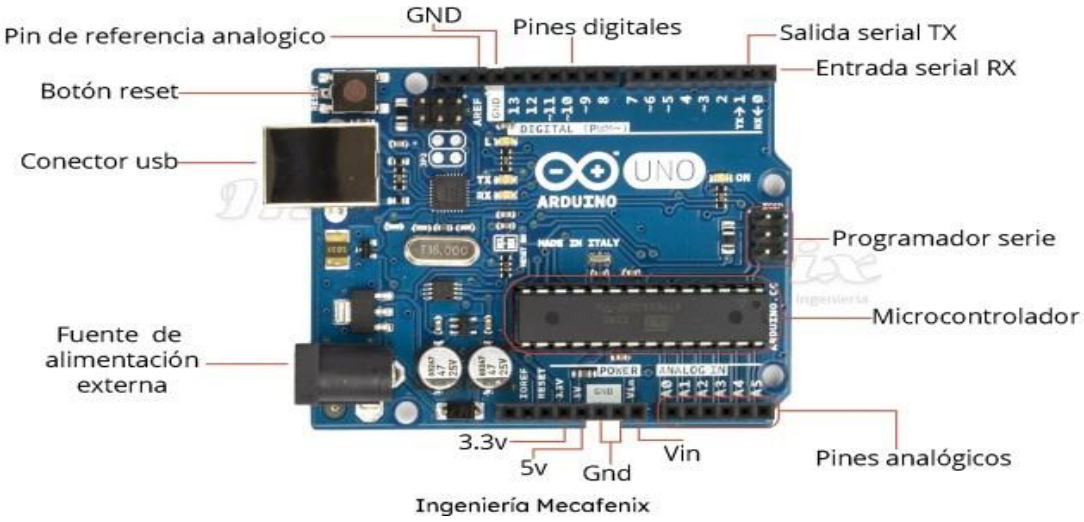


Imagen 8.2 Arduino Uno

Características principales:

- Microcontrolador ATmega328P
- 14 pines de entrada/salida digital
- 6 entradas análogas
- Un resonador cerámico de 16 MHz
- Un conector USB tipo hembra
- Un conector ICSP
- Un botón de reinicio
- Alimentación por USB o fuente de alimentación externa

¿Qué se puede hacer con Arduino Uno?

Arduino Uno se puede utilizar para crear una amplia gama de proyectos, incluyendo:

- **Electrónica básica:** Controles de luces, interruptores, sensores, etc.
- **Robótica:** Robots simples, brazos robóticos, vehículos controlados a distancia, etc.
- **Domótica:** Sistemas de control de temperatura, iluminación y seguridad para el hogar.
- **Proyectos interactivos:** Instrumentos musicales, juegos, obras de arte interactivas, etc.
- **Prototipado rápido:** Desarrollar rápidamente ideas para nuevos productos o proyectos.

¿Por qué usar Arduino Uno?

Hay muchas razones por las que Arduino Uno es una plataforma popular para el desarrollo de electrónica:

- **Fácil de usar:** Arduino Uno es fácil de aprender y usar, incluso para principiantes sin experiencia previa en programación o electrónica.
- **De código abierto:** El hardware y el software de Arduino Uno son de código abierto, lo que significa que son gratuitos de usar y modificar.
- **Gran comunidad:** Hay una gran comunidad de usuarios de Arduino que comparten información, tutoriales y proyectos.
- **Amplia gama de recursos:** Hay una amplia gama de recursos disponibles para aprender a usar Arduino Uno, incluyendo libros, tutoriales en línea y cursos.
- **Precio accesible:** Arduino Uno es una placa de microcontroladores relativamente barata, lo que la hace accesible para una amplia gama de usuarios.

Cómo empezar con Arduino Uno:

Para empezar con Arduino Uno, necesitarás lo siguiente:

- Una placa Arduino Uno
- Un cable USB
- Un ordenador con software Arduino instalado
- Algunos componentes electrónicos básicos, como Leds, resistencias y sensores

```
Public Sub New()
     InitializeComponent()
     ' Abrir puerto mientras se ejecute la aplicación
     If Not SerialPort1.IsOpen Then
         Try
             SerialPort1.Open()
         Catch ex As System.Exception
             MessageBox.Show(ex.ToString())
         End Try
     End If
 End Sub
```

Imagen 9.2 Ejemplo de comunicación Arduino 1 con Visual Basic

Sensor de peso de 20 kg

Un sensor de peso de 20 kilogramos, también conocido como celda de carga de 20 kg, es un dispositivo electrónico que convierte una fuerza aplicada (en este caso, el peso) en una señal eléctrica proporcional. Esta señal eléctrica puede ser luego procesada por un microcontrolador como Arduino para obtener una lectura precisa del peso.

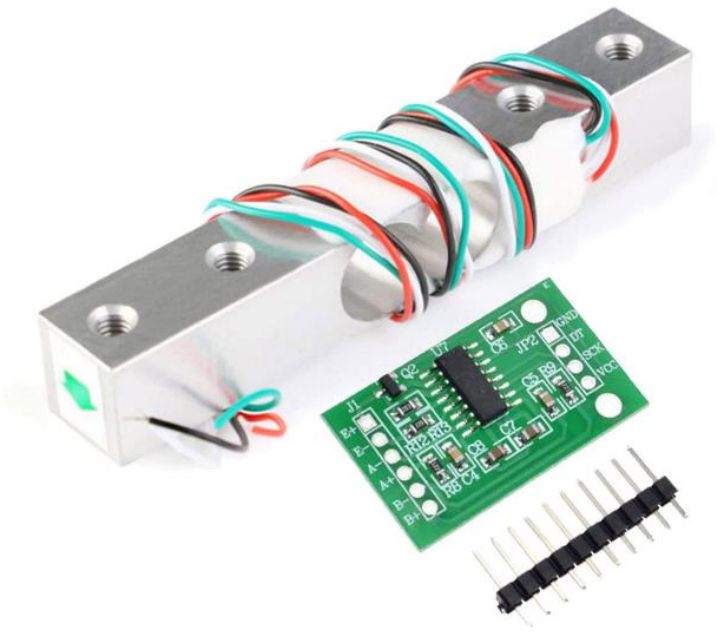


Imagen 10.2 sensor de peso de 20 kg

¿Cómo funciona?

La mayoría de los sensores de peso de 20 kg utilizan galgas extensiométricas. Estas son pequeñas resistencias eléctricas que cambian su valor cuando se deforman. Cuando se aplica una fuerza a la celda de carga, esta se deforma ligeramente, lo que provoca un cambio en la resistencia de las galgas extensiométricas. Este cambio de resistencia es detectado y amplificado por un circuito electrónico, y luego convertido en una señal digital que puede ser leída por un microcontrolador.

Componentes principales:

- Celda de carga: La parte física que soporta el peso y contiene las galgas extensiométricas.

- Galgas extensiométricas: Pequeños sensores que miden la deformación de la celda de carga.
- Módulo amplificador: Circuito electrónico que amplifica la débil señal de las galgas extensiométricas para que pueda ser leída por un microcontrolador.
- Microcontrolador: (Por ejemplo, Arduino) Procesa la señal digital del módulo amplificador y realiza cálculos para determinar el peso.

Aplicaciones:

Los sensores de peso de 20 kg tienen una amplia gama de aplicaciones, incluyendo:

- Básculas: Para pesar objetos de hasta 20 kg.
- Sistemas de dosificación: Para controlar la cantidad de un producto en un proceso industrial.
- Robots industriales: Para controlar la fuerza aplicada por un brazo robótico.
- Proyectos DIY: Para construir básculas personalizadas, dispensadores de alimentos, etc.
- Precisión: La precisión del sensor depende de varios factores, como la calidad de las galgas extensiométricas, el módulo amplificador y la calibración.
- Rango de medición: En este caso, el rango es de 0 a 20 kg.
- Tasa de muestreo: La frecuencia con la que se toman las medidas.
- Salida: La mayoría de los sensores de peso ofrecen una salida analógica (voltaje o corriente) que debe ser convertida a digital por un microcontrolador.

Concepto de purificadora

¿Qué es una purificadora de agua?

Una purificadora de agua es un dispositivo que elimina las impurezas y contaminantes del agua, haciéndola segura para beber. Las purificadoras de agua pueden utilizar una variedad de métodos para purificar el agua, como la filtración, la ebullición, la desinfección con cloro o la ósmosis inversa.

Imagen 11.2 Purificadora de agua.

¿Por qué es importante purificar el agua?

El agua contaminada puede contener una variedad de bacterias, virus y parásitos que pueden causar enfermedades graves. La purificación del agua mata o elimina estos microorganismos, lo que la hace segura para beber. Además, el agua purificada puede tener un mejor sabor y olor que el agua del grifo.

¿Qué tipos de purificadoras de agua hay?

Existen diferentes tipos de purificadoras de agua, cada una con sus propias ventajas y desventajas. Algunos de los tipos más comunes incluyen:

- **Purificadoras de filtro:** Estas purificadoras utilizan filtros para eliminar las impurezas del agua. Los filtros pueden ser de diferentes materiales, como carbón activado, arena o cerámica.
- **Purificadoras de ebullición:** Estas purificadoras matan las bacterias y virus hirviendo el agua. La ebullición es un método simple y efectivo para purificar el agua, pero puede ser lento y consumir mucha energía.
- **Purificadoras de desinfección con cloro:** Estas purificadoras añaden cloro al agua para matar las bacterias y virus. El cloro es un desinfectante eficaz, pero puede dejar un sabor desagradable en el agua.

En general, las purificadoras de agua son una herramienta importante para garantizar que tengas acceso a agua potable segura. Al elegir una purificadora de agua adecuada y usarla correctamente, puedes protegerte a ti mismo y a tu familia de enfermedades transmitidas por el agua.

Física de conversión de litros

La conversión de litros a kilogramos es una operación común en física e ingeniería, ya que permite relacionar el volumen de un fluido con su masa. Pero es importante considerar que esta conversión no es directa, ya que depende de la densidad del fluido en cuestión.

$$\text{Masa (kg)} = \text{Densidad (kg/L)} * \text{Volumen (L)}$$

Imagen 12.2 Fórmula general para convertir litros a kilogramos

Densidad:

La densidad de un fluido se define como la masa por unidad de volumen. Se expresa comúnmente en unidades de kg/L o g/cm³. La densidad de un fluido depende de su composición y temperatura. Por ejemplo, la densidad del agua a temperatura ambiente es de aproximadamente 1000 kg/L o 1 g/cm³. Esto significa que 1 litro de agua a temperatura ambiente tiene una masa de 1 kilogramo.

Masa (kg) = Densidad (kg/L) * Volumen (L)

Ejemplo:

Calcule la masa de 2 litros de aceite de oliva, cuya densidad es de 0.915 kg/L.

Solución:

Masa (kg) = 0.915 kg/L * 2 L = 1.83 kg

Casos especiales:

- **Agua:** Para el agua a temperatura ambiente, la conversión es simple: 1 litro = 1 kilogramo.

La conversión de litros a kilogramos requiere considerar la densidad del fluido en cuestión. La fórmula general es Masa (kg) = Densidad (kg/L) * Volumen (L). Es importante recordar que esta conversión no es directa y que la densidad puede variar dependiendo de las condiciones.

Pliego de corcho

es un material interesante con propiedades útiles. Permíteme explicarte por qué el papel de corcho es antihumedad y no es conductor de electricidad:

Aislamiento térmico eficiente: El corcho tiene una estructura celular única que proporciona un excelente aislamiento térmico. Esto ayuda a mantener la temperatura interior de los edificios de manera más eficiente, lo que ahorra energía a largo plazo1.

Material sostenible y ecológico: El corcho se extrae de la corteza del alcornoque, un árbol que se regenera y se puede volver a cosechar sin dañarlo. Es una alternativa sostenible a otros materiales de aislamiento1.

Resistencia al fuego: El corcho contiene sustancias naturales que actúan como retardantes de llama, lo que lo hace más seguro en caso de incendio1.

No conductor de electricidad: El corcho, al ser no metálico, no contiene electrones libres que puedan moverse para conducir la electricidad. Por lo tanto, actúa como un aislante, impidiendo el flujo de corriente eléctrica

Imagen 13.2 Pliego de papel corcho

Madera

Un material que tradicionalmente asociamos con la construcción y la carpintería, está demostrando ser una opción innovadora y prometedora en el campo de la electrónica. A pesar de su naturaleza orgánica, la madera presenta una serie de ventajas que la hacen atractiva para diversas aplicaciones. Su amplia disponibilidad y bajo costo la convierten en un material accesible para prototipos y proyectos a pequeña escala. Además, al ser un recurso renovable y biodegradable, su uso en electrónica contribuye a una producción más sostenible y respetuosa con el medio ambiente. La madera también destaca por su flexibilidad y ligereza, lo que permite la creación de dispositivos con formas ergonómicas y adaptables, como wearables o sensores flexibles. A pesar de su naturaleza orgánica, la madera ha demostrado ser un material estable y capaz de soportar componentes electrónicos y conexiones de manera eficiente, ampliando así las posibilidades de diseño en el campo de la electrónica.

Imagen 14.2 Madera resistente

Plástico

Un material sintético omnipresente en nuestra vida cotidiana ha revolucionado la industria electrónica. Sus propiedades versátiles lo convierten en un material ideal para la fabricación de componentes electrónicos. La plasticidad del plástico permite crear formas complejas y miniaturizadas, lo que ha dado lugar a dispositivos electrónicos cada vez más pequeños y potentes. Además, su bajo costo y facilidad de procesamiento hacen que sea un material muy atractivo para la producción a gran escala. Pero, a pesar de sus ventajas, el plástico también presenta desafíos en cuanto a sostenibilidad, ya que muchos tipos de plástico no son biodegradables y contribuyen a la contaminación ambiental. Actualmente, se están desarrollando alternativas más ecológicas, como los bioplásticos, que buscan combinar las ventajas del plástico con un menor impacto ambiental.

Imagen 15.2 Cajitas de plástico

Metal inoxidable

En metalurgia, el acero inoxidable se define como una aleación de acero con un mínimo del 10 % al 12 % de cromo contenido en masa.12nota 1 También puede contener otros metales, como molibdeno, níquel y tungsteno.

El acero inoxidable es un acero de elevada resistencia a la corrosión, dado que el cromo u otros metales aleantes que contiene, poseen gran afinidad por el oxígeno y reacciona con él formando una capa pasivadora, evitando así la corrosión del hierro (los metales puramente inoxidables, que no reaccionan con oxígeno son oro y platino, y de menor pureza se llaman resistentes a la corrosión, como los que contienen fósforo). Sin embargo, esta capa puede ser afectada por algunos ácidos, dando lugar a que el hierro sea atacado y oxidado por mecanismos intergranulares o picaduras generalizadas. Algunos tipos de acero inoxidable contienen además otros elementos aleantes; los principales son el níquel y el molibdeno.

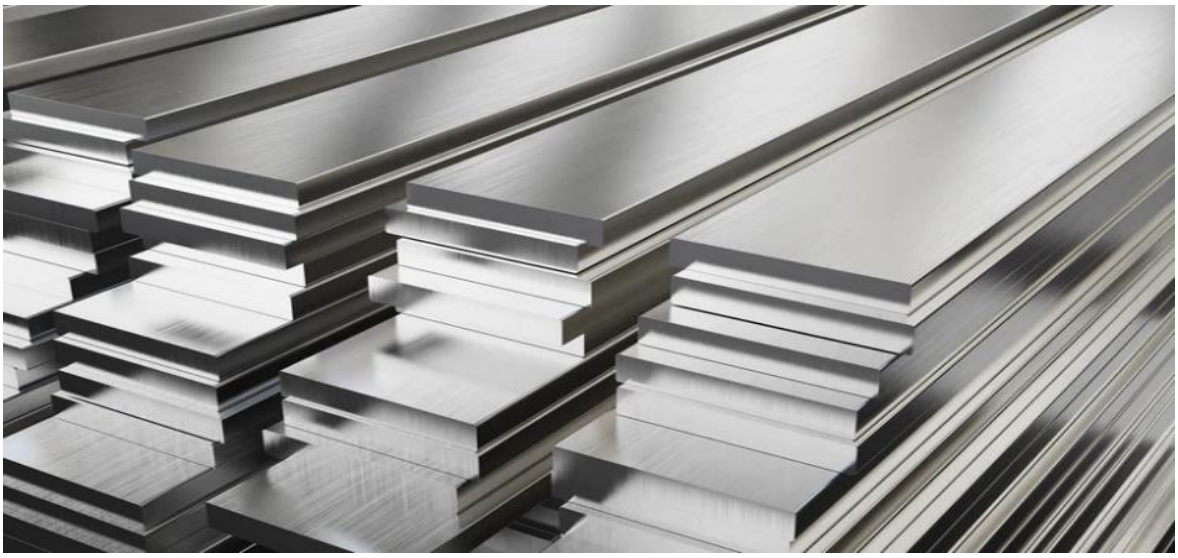


Imagen 16.2 Metal inoxidable

Cinta Aislante

La cinta aislante (conocida también como cinta aisladora o cinta de aislar) es un tipo de cinta adhesiva de presión usada principalmente para aislar empalmes de hilos y cables eléctricos. Este tipo de cinta es capaz de resistir condiciones de temperaturas extremas, corrosión, humedad y altos voltajes. La cinta está fabricada en material de PVC delgado, con un ancho generalmente de 14 mm; uno de los lados de la cinta está impregnado con un adhesivo. El PVC ha sido elegido por ser un material de bajo costo, flexible y tener excelentes propiedades de aislante eléctrico, aunque posee la desventaja de endurecerse con el tiempo y el calor.

Imagen 17.2 Cinta aislante

Tornillo o pija de metal

Los tornillos, también conocidos como pijas, son excelentes para fijar cosas debido a su fuerte sujeción mecánica gracias a las roscas, su versatilidad en diversos materiales como madera, metal y plástico, su facilidad de instalación y remoción con herramientas comunes, su durabilidad y resistencia a las vibraciones, y su capacidad para proporcionar una fijación precisa y ajustable, lo cual es crucial en proyectos que requieren exactitud en la alineación y ensamblaje.

Imagen 18.2 Tornillos o pijas

Acrílico

El acrílico, también conocido como polimetilmetacrilato (PMMA), es un tipo de plástico transparente que se destaca por su alta resistencia al impacto, ligereza y durabilidad[1]. Este material es una excelente alternativa al vidrio, ya que ofrece una claridad óptica similar pero con mayor resistencia y menor peso. Además, el acrílico es fácil de moldear y cortar, lo que lo hace ideal para una amplia variedad de aplicaciones, desde ventanas y techos hasta letreros y exhibidores. Su capacidad para resistir las condiciones climáticas y su facilidad de mantenimiento lo convierten en una opción popular tanto en el ámbito doméstico como industrial.

Imagen 19.2 Tornillos o pijas

¿Qué es un prototipo?

Un prototipo se refiere a una representación temprana y funcional de un producto o sistema, diseñada para proporcionar una vista preliminar y tangible de cómo se verá y se comportará el producto final. Se trata de un modelo inicial que encapsula las características clave, funcionalidades y aspectos visuales del producto, permitiendo a los diseñadores, desarrolladores y equipos de proyecto explorar conceptos, probar ideas y recopilar retroalimentación antes de avanzar hacia la producción completa.

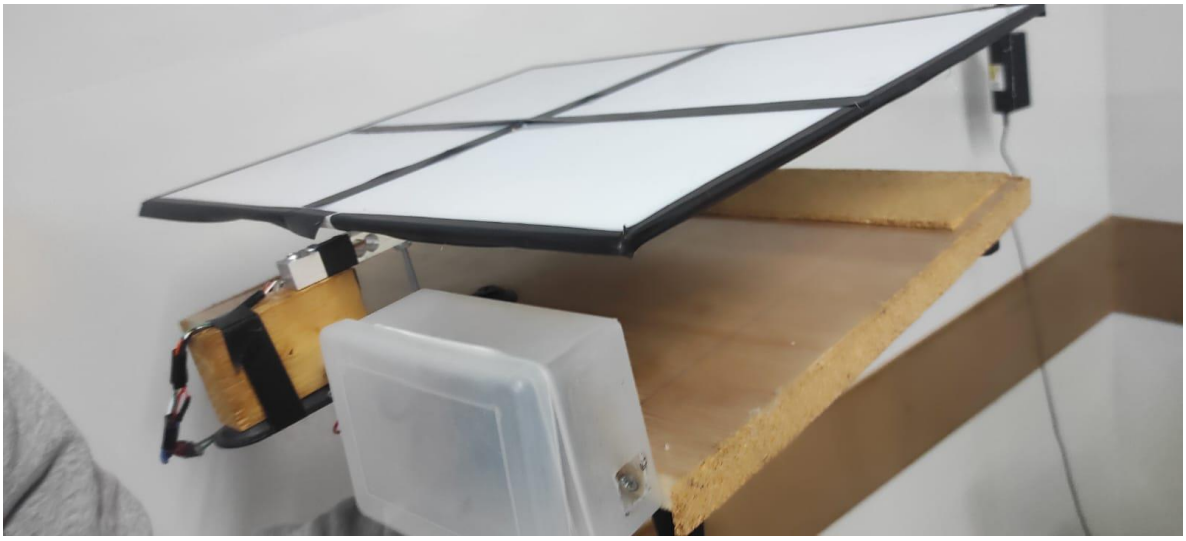


Imagen 20.2 Prototipo de báscula.

¿Por qué es importante desarrollar un prototipo de nuestro producto?

- **Validación de conceptos:** Los prototipos permiten a los equipos validar y corregir conceptos de diseño, asegurándose de que el producto cumpla con los objetivos y necesidades del usuario antes de invertir recursos significativos en el desarrollo completo. Conoce también qué es una prueba de concepto.

- **Retroalimentación temprana**: Al proporcionar una representación tangible, los prototipos facilitan la obtención de retroalimentación temprana de los usuarios, clientes y otras partes interesadas. Esta retroalimentación temprana es crucial para identificar y abordar problemas potenciales antes de que el producto final esté completamente desarrollado.

- **Ahorro de tiempo y recursos**: Identificar y solucionar problemas en una etapa temprana del proceso de desarrollo a través de prototipos puede evitar costosas implementaciones más adelante. Esto contribuye a un uso más eficiente de los recursos y acelera el tiempo de llegada al mercado.
- **Comunicación efectiva**: Los prototipos sirven como herramientas de comunicación efectivas entre los diferentes miembros del equipo, permitiendo una comprensión clara de la visión del producto. Además, facilitan la alineación entre los stakeholders al proporcionar una referencia visual y funcional.

- **Desarrollo frecuente**: La naturaleza iterativa de la creación de prototipos permite a los equipos perfeccionar y evolucionar gradualmente el producto a medida que avanzan en el proceso de diseño. Esto fomenta la adaptabilidad y la mejora continua.

Prototipos de Alta Fidelidad

Los prototipos de alta fidelidad son representaciones avanzadas y detalladas de un producto, que se aproximan mucho a la versión final en términos de diseño, interacción y funcionalidad. Estos tipos de prototipos son esenciales para entender la experiencia del usuario final y para realizar pruebas de usabilidad antes del lanzamiento del producto.

Modelos Funcionales: Los modelos funcionales son prototipos que no solo imitan el diseño del producto final, sino que también replican su funcionalidad en gran medida. Son especialmente útiles para simular la interacción del usuario con el producto, permitiendo evaluar la usabilidad y la efectividad del diseño.

Prototipos Digitales: Los prototipos digitales son creados utilizando software avanzado de diseño y simulación, ofreciendo un nivel de detalle y precisión muy cercano al producto final. Estos prototipos son indispensables para visualizar el producto en su contexto final, incluyendo detalles visuales, interactividad y transiciones.

Aplicaciones: Los prototipos de alta fidelidad son particularmente beneficiosos en industrias y proyectos donde la precisión del diseño y la experiencia del usuario son críticas. Algunas de las aplicaciones más comunes incluyen:
– Desarrollo de Software y Aplicaciones Móviles: Donde la interacción del usuario y la interfaz son fundamentales para el éxito del producto.

– Diseño de Sitios Web y Comercio Electrónico: Para asegurar que la disposición de elementos, la navegación y la interactividad cumplan con las expectativas de los usuarios y fomenten conversiones.

– Dispositivos y Tecnología Wearable: Donde la ergonomía y la usabilidad son esenciales para la adopción del producto por parte del usuario.

Prototipos de Experiencia de Usuario (UX)

Los prototipos de experiencia de usuario son herramientas esenciales en el diseño de interfaces y en la optimización de la interacción del usuario con el producto. Permiten a los diseñadores experimentar y probar diversas aproximaciones para mejorar la usabilidad y la accesibilidad.

Wireframes y Prototipos Interactivos: Los wireframes son esquemas visuales que representan la estructura básica de una página web o aplicación, mientras que los prototipos interactivos simulan la funcionalidad y la interacción en un entorno cercano al producto final.

Características Principales:

– Enfoque en la Usabilidad: Permiten evaluar la eficacia de la disposición de elementos, la navegación y la interacción general.

– Iteración Rápida: Facilitan el proceso de iteración, permitiendo realizar cambios rápidos basados en el feedback de los usuarios o del equipo.

Herramientas de Prototipado UX: Existen diversas herramientas disponibles para el prototipado UX, cada una con características únicas que las hacen adecuadas para diferentes fases del proceso de diseño. Algunas de las más populares incluyen Sketch, Adobe XD, Figma y Axure.

Características Principales:

– Flexibilidad y Funcionalidad: Ofrecen una amplia gama de funciones, desde la creación de wireframes básicos hasta prototipos interactivos altamente detallados.
– Colaboración y Compartición: Muchas herramientas facilitan la colaboración en tiempo real y la compartición de prototipos con equipos y stakeholders.
Pruebas de Usabilidad: Las pruebas de usabilidad son fundamentales para validar la eficacia de un diseño desde la perspectiva del usuario final. Implican observar a los usuarios mientras interactúan con el prototipo, recopilando información valiosa sobre la experiencia de usuario.

Prototipos Físicos
Los prototipos físicos son representaciones tangibles de un producto, esenciales en el proceso de diseño y desarrollo, especialmente en áreas como la ingeniería, la electrónica y el diseño industrial.

Prototipos 3D: Los prototipos 3D utilizan tecnologías como la impresión 3D para crear modelos físicos detallados del diseño de un producto.

Características Principales:
– Visualización Realista: Ofrecen una representación física y tangible que facilita la comprensión del diseño y la funcionalidad.
– Pruebas de Concepto: Permiten validar la viabilidad del diseño y realizar ajustes antes de la producción en masa.

Prototipos de Ingeniería:
Los prototipos de ingeniería se enfocan en probar los aspectos mecánicos y técnicos del diseño, asegurando que el producto funcione según las especificaciones y cumpla con los estándares de calidad.

Características Principales:
– Validación Técnica: Prueban la viabilidad técnica del diseño y la integración de componentes.
– Identificación de Mejoras: Permiten identificar áreas de mejora en términos de rendimiento, durabilidad y costos de producción.

Prototipos de Pruebas

Prototipos para Pruebas de Concepto: Los prototipos para pruebas de concepto son esenciales en las etapas iniciales del desarrollo del producto. Su principal objetivo es validar la viabilidad y el atractivo de la idea central del producto antes de invertir significativamente en su desarrollo.

Características Principales:
– Validación Temprana: Permiten a los equipos validar la idea del producto con usuarios potenciales, stakeholders o inversores, asegurando que el concepto cumple con las expectativas y resuelve una necesidad real.
– Eficiencia en Costos y Tiempo: Minimizan el riesgo de invertir en un producto que no tiene un mercado viable o no satisface una necesidad del usuario.
Prototipos para Pruebas de Campo: Los prototipos para pruebas de campo se utilizan para evaluar el producto en el entorno en el que se usará. Estos prototipos son más detallados y funcionales, permitiendo a los equipos recolectar datos precisos y feedback del usuario en condiciones reales.

Características Principales:
– Realismo en las Pruebas: Ofrecen una comprensión profunda de cómo el producto se desempeña en situaciones reales, fuera del entorno controlado de laboratorio o diseño.

– Feedback Directo del Usuario: Permiten la recolección de observaciones y comentarios directos de los usuarios finales, proporcionando insights valiosos para la mejora del producto.
Importancia de las Pruebas: Las pruebas son fundamentales en el proceso de desarrollo del producto. No solo reducen la incertidumbre y el riesgo asociados con el lanzamiento de un nuevo producto, sino que también aseguran que el producto final cumpla con las necesidades y expectativas del usuario.
Beneficios Claves:
– Reducción de Riesgos: Ayudan a identificar y solucionar problemas antes de que el producto llegue al mercado, evitando costos elevados de corrección post-lanzamiento.
– Optimización del Producto: A través de las pruebas, se pueden hacer ajustes y mejoras continuas, asegurando que el producto final sea de la más alta calidad y relevancia para el usuario.

Prototipos Rápidos

Técnicas de Prototipado Rápido: Las técnicas de prototipado rápido incluyen una variedad de métodos que permiten la creación rápida de modelos físicos a partir de diseños digitales, utilizando principalmente tecnologías de fabricación aditiva como la impresión 3D y el corte por láser.

Impresión 3D

– Flexibilidad de Diseño: Permite la creación de prototipos complejos y detallados que serían difíciles o imposibles de fabricar con métodos tradicionales.
– Iteración Rápida: Facilita la realización de cambios rápidos en el diseño y la producción de múltiples versiones en un corto período de tiempo.

Corte por Láser:

– Precisión y Detalle: Ofrece cortes precisos y detallados, ideal para prototipos que requieren alta precisión en las dimensiones y los detalles.
– Versatilidad de Materiales: Apto para una amplia gama de materiales, incluyendo plásticos, maderas y metales.

Ventajas del Prototipado Rápido: El prototipado rápido ofrece múltiples ventajas que pueden acelerar significativamente el proceso de desarrollo de productos, haciendo que sea una opción atractiva para empresas y diseñadores.

Reducción de Tiempos de Desarrollo:

– Producción Acelerada: Reduce significativamente el tiempo entre el diseño inicial y la evaluación física del prototipo.
– Feedback y Mejoras Rápidas: Permite una iteración rápida basada en el feedback, agilizando el proceso de refinamiento del diseño.

Costo-Eficiencia:

– Menor Gasto en Materiales: Reduce el desperdicio de material al utilizar solo lo necesario para cada prototipo.
– Reducción de Costos de Fabricación: Minimiza los costos asociados con la configuración y fabricación de moldes o herramientas especiales.

Selección de Materiales: La selección de materiales es un aspecto crítico en el prototipado rápido, ya que el material utilizado puede influir significativamente en la funcionalidad, apariencia y durabilidad del prototipo.

Consideraciones Clave:

Propiedades del Material: Evaluar la resistencia, flexibilidad, durabilidad y estética del material para asegurarse de que cumple con los requisitos del prototipo.

Compatibilidad con la Tecnología de Prototipado: Asegurarse de que el material es compatible con la técnica de prototipado elegida (por ejemplo, no todos los plásticos son adecuados para todas las impresoras 3D).

Costo y Disponibilidad: Considerar el costo del material y su disponibilidad en el mercado, especialmente para prototipos que se producirán en grandes cantidades.

Prototipos Virtuales

Realidad Aumentada (AR) y Realidad Virtual (VR): Las tecnologías de AR y VR están revolucionando la forma en que los prototipos se diseñan, desarrollan y presentan. Estas tecnologías permiten a los diseñadores y usuarios interactuar con prototipos virtuales de una manera altamente inmersiva.

Características Principales:

– Experiencia Inmersiva: La AR y la VR ofrecen una experiencia realista y envolvente que permite a los usuarios visualizar y interactuar con los prototipos en un entorno controlado.

– Iteración y Modificación en Tiempo Real: Permiten realizar cambios en el diseño de manera instantánea y visualizar los efectos de esos cambios de manera inmediata.

Simulaciones Computarizadas: Las simulaciones computarizadas utilizan modelos matemáticos y físicos para predecir cómo un producto reaccionará bajo ciertas condiciones. Son herramientas valiosas para anticipar problemas de rendimiento y optimizar el diseño antes de la fabricación física.

Características Principales:

Predicción del Comportamiento del Producto: Ayudan a entender cómo se comportará un producto en situaciones del mundo real, evaluando su resistencia, durabilidad y otras propiedades críticas.

Optimización del Diseño: Permiten a los diseñadores y ingenieros realizar ajustes y mejoras en el diseño para maximizar el rendimiento y la eficiencia del producto.

Aplicaciones en Industrias: Los prototipos virtuales son ampliamente utilizados en una variedad de sectores, incluyendo la arquitectura, la ingeniería, la medicina y el entretenimiento, debido a su capacidad para proporcionar visualizaciones detalladas y realizar pruebas precisas.

En la Arquitectura:

Visualización de Proyectos: Permiten a los arquitectos y clientes recorrer edificaciones antes de su construcción, proporcionando una comprensión clara del diseño y la estética del espacio.

En la Ingeniería:

Pruebas y Análisis: Facilitan la realización de pruebas de estrés, análisis de flujo de fluidos y otros estudios técnicos, lo que es esencial para validar la seguridad y funcionalidad de estructuras y maquinarias.

Evaluación del Rendimiento de su Prototipo

Comprensión de la Evaluación de la Funcionalidad

La evaluación de la funcionalidad implica un análisis sistemático de las características, capacidades y comportamiento de su prototipo. Este proceso va más allá de las simples pruebas de funcionalidad, considerando aspectos como la usabilidad, la confiabilidad y la satisfacción del usuario. A continuación, se presentan ideas desde diferentes perspectivas:

Evaluación Centrada en el Usuario:

Pruebas de Usabilidad: Realice pruebas de usabilidad con usuarios representativos. Observe cómo interactúan con su prototipo, identifique los puntos débiles y recopile comentarios. Por ejemplo, si está diseñando una aplicación móvil, observe a los usuarios mientras navegan por las pantallas, realizan tareas y brindan información.

Encuestas de Usuarios: Cree encuestas para recopilar datos cuantitativos y cualitativos. Pregunte a los usuarios sobre su experiencia, facilidad de uso y satisfacción general. Utilice escalas Likert o .s abiertas. Por ejemplo, “En una escala del 1 al 5, ¿qué tan intuitiva fue la navegación del prototipo?”

Entrevistas con Usuarios: Realice entrevistas en profundidad con los usuarios. Comprenda sus necesidades, expectativas y puntos débiles. Haga .s como: “¿Qué funciones le parecieron más útiles?” o “¿Qué le frustró durante la interacción?”

Evaluación Técnica:

Pruebas Funcionales: Verifique que cada característica funcione según lo previsto. Cree casos de prueba que cubran diferentes escenarios. Por ejemplo, si está creando una aplicación meteorológica, pruebe cómo maneja las actualizaciones de ubicación, muestra pronósticos y maneja errores.

Pruebas de Rendimiento: Evalúe los tiempos de respuesta, la utilización de recursos y la escalabilidad. Mida qué tan bien se desempeña su prototipo bajo

estrés (por ejemplo, alta carga de usuarios o cálculos complejos). Por ejemplo, pruebe la rapidez con la que un sitio web de comercio electrónico carga páginas de productos durante las horas pico.

Pruebas de Compatibilidad: Asegúrese de que su prototipo funcione en diferentes dispositivos, navegadores y sistemas operativos. Pruebe en varias plataformas (Windows, macOS, iOS, Android) y tamaños de pantalla. Considere las experiencias tanto de escritorio como móviles.

Perspectiva Empresarial:

Priorización de Características: Evalúe qué características son esenciales para el éxito del prototipo. Priorice en función de los objetivos comerciales, las necesidades de los usuarios y los recursos de desarrollo. Por ejemplo, un producto mínimo viable (MVP) puede centrarse primero en las funciones principales.

Análisis de Costo-Beneficio: Evalúe las compensaciones entre agregar funciones y los costos de desarrollo. Considere el tiempo de desarrollo, el mantenimiento y los ingresos potenciales. A veces, la simplicidad y la funcionalidad optimizada conducen a mejores resultados.

Market Fit: Analice qué tan bien se alinea su prototipo con las demandas del mercado. Investigue a los competidores, las tendencias de los usuarios y las tecnologías emergentes. Por ejemplo, si está diseñando una aplicación de fitness, explore si aborda las tendencias actuales de salud y bienestar.

¿Qué es MD5?

MD5, o Message-Digest Algorithm 5 (Algoritmo de Resumen del Mensaje 5), es una función hash criptográfica de 128 bits. Esto significa que toma un conjunto de datos de cualquier tamaño y lo transforma en una cadena de caracteres única de 32 dígitos hexadecimales. Esta cadena es conocida como "hash" o "resumen".

¿Para qué se utiliza?

- **Verificación de integridad de datos:** Al comparar el hash de un archivo original con el hash de una copia, se puede determinar si el archivo ha sido modificado. Cualquier cambio, por pequeño que sea, resultará en un hash completamente diferente.
- **Almacenamiento de contraseñas:** Aunque ya no se recomienda por razones de seguridad, MD5 se usó ampliamente para almacenar contraseñas de forma hash en bases de datos. En lugar de almacenar la contraseña en texto plano, se almacenaba su hash. Sin embargo, debido a su vulnerabilidad a ataques de fuerza bruta y colisiones, esta práctica ha sido reemplazada por algoritmos más seguros como bcrypt o Argon2.

¿Cómo funciona?

El algoritmo MD5 toma un mensaje de cualquier tamaño y lo procesa a través de una serie de operaciones matemáticas para generar un hash de tamaño fijo. Incluso un pequeño cambio en el mensaje original resultará en un hash completamente diferente. Esta propiedad es lo que hace que MD5 sea útil para la verificación de integridad de datos.

Limitaciones y Vulnerabilidades

- **Colisiones:** A lo largo de los años, se han descubierto varias colisiones en MD5, lo que significa que es posible encontrar dos mensajes diferentes que produzcan el mismo hash. Esto compromete su seguridad para aplicaciones criptográficas más exigentes.
- **Ataques de fuerza bruta:** Aunque generar un hash MD5 es rápido, invertir el proceso (es decir, encontrar el mensaje original a partir de un hash dado) es computacionalmente muy costoso. Sin embargo, con suficiente poder de cómputo, es posible realizar ataques de fuerza bruta en ciertos escenarios.

¿Por qué no se recomienda usar MD5 hoy en día?

- **Vulnerabilidades:** Como se mencionó anteriormente, las colisiones y la susceptibilidad a ataques hacen que MD5 sea inseguro para muchas aplicaciones.
- **Algoritmos más seguros:** Existen algoritmos de hash más modernos y seguros, como SHA-256 y SHA-3, que ofrecen una mayor resistencia a ataques y colisiones.

MD5 fue un algoritmo hash popular en su tiempo, pero sus limitaciones han llevado a su reemplazo en muchas aplicaciones. Si bien todavía se utiliza en algunos contextos, es crucial comprender sus debilidades y elegir los algoritmos adecuados para garantizar la seguridad de los datos.

2.4 ESTUDIO DEL MERCADO

Este apartado, permite hacer la integración de la opinión del encargado y la de los trabajadores para el desarrollo del sistema de registro automático, la encuesta se aplica al encargado del negocio y a los trabajadores, dando un total de 3 personas encuestadas. Las .s que se aplicaron son las siguientes:

Primera Entrevista: Conociendo al Usuario y sus Necesidades

1.: ¿Con qué frecuencia utiliza un sistema para registrar las salidas de garrafones?

Respuesta varias veces al día.

2.: ¿Qué información considera esencial registrar al momento de realizar una salida de garrafón?

Número de garrafón en el día, cantidad, fecha y hora de salida

3.: ¿Cómo realiza actualmente el seguimiento de las salidas de garrafones? En hojas de cálculo o Excel

Respuesta (ejemplo): Actualmente lo hago en una hoja de cálculo de Excel.

4.: ¿Qué dificultades encuentra en su método actual de registro?

A veces se pierden datos, es difícil generar reportes detallados y lleva mucho tiempo actualizar la información.

5.: ¿Qué tipo de reportes le gustaría poder generar con un sistema automatizado?

Registrar salidas

6.: ¿Qué tan importante es para usted tener acceso a la información de las salidas de garrafones desde cualquier lugar?

Es muy importante, ya que a veces necesito consultar la información

7.: ¿Qué nivel de conocimientos en tecnología tiene? ¿Se siente cómodo utilizando software nuevo?

Tengo conocimientos básicos de informática y me adapto fácilmente a nuevas herramientas.

8.: ¿Qué características considera indispensables en un sistema de gestión de salidas de garrafones?

Facilidad de uso, seguridad de los datos, rapidez

Segunda Entrevista: Profundizando en la Experiencia del Usuario

1.: ¿Qué tan importante es para usted la seguridad de los datos almacenados en el sistema? Es fundamental, ya que la información de los clientes es confidencial.

2.: ¿Qué tipo de notificaciones o alertas le gustaría recibir del sistema?

Alertas de bajo stock, recordatorios de mantenimiento de equipos y notificaciones de nuevos registros.

3.: ¿Cómo le gustaría que el sistema le ayudara a optimizar la gestión de su inventario?

Solo quiero registrar salidas

4.: ¿Qué tipo de integración le gustaría que tuviera el sistema? (Con otro software, dispositivos, etc.)

Me gustaría que sea un sistema único

5.: ¿Qué tan importante es para usted poder personalizar la interfaz del sistema?

Sería útil la visualización de los datos.

6.: ¿Qué nivel de soporte técnico espera recibir al utilizar este sistema?

Espero que haya un servicio de atención al cliente disponible para resolver cualquier duda o problema.

7.: ¿Qué otros sistemas o herramientas utiliza actualmente en su negocio?

ninguno

8.: ¿Cuál es su mayor preocupación al implementar un nuevo sistema de gestión?

La curva de aprendizaje

9.: ¿Qué beneficios espera obtener al utilizar un sistema de gestión de salidas de garrafones automatizado?

Ahorrar tiempo, mejorar la precisión de los registros, tomar mejores decisiones basadas en datos y ofrecer un mejor servicio a los clientes.

10.: ¿En una escala del 1 al 10, cuán satisfecho estaría si el sistema cumpliera con todas sus expectativas?

10, estaría muy satisfecho.

11.: ¿Hay alguna otra característica o funcionalidad que considere importante y que no hayamos mencionado?

Me gustaría poder pesar los garrafones.

¿Por qué las Preguntas Técnicas son Clave?

Un estudio de mercado tradicional a menudo se basa en encuestas con preguntas generales sobre preferencias y opiniones. Sin embargo, para obtener una visión verdaderamente profunda de las necesidades y expectativas de los usuarios, es fundamental recurrir a preguntas técnicas. Estas preguntas permiten:

Identificar necesidades específicas: Al indagar en los procesos y desafíos actuales de los usuarios, se pueden identificar necesidades específicas y oportunidades de mejora que podrían no ser evidentes en una encuesta general.

Evaluar la viabilidad técnica: Las preguntas técnicas permiten evaluar la compatibilidad de una solución propuesta con los sistemas y procesos existentes del usuario.

Comparar soluciones alternativas: Al comprender las necesidades técnicas, se pueden comparar diferentes soluciones y seleccionar la que mejor se adapte a las circunstancias del usuario.

Establecer requisitos claros: Las respuestas a las preguntas técnicas ayudan a establecer requisitos claros para el desarrollo de un producto o servicio.

Primera Entrevista: Conociendo el Contexto Técnico

Infraestructura actual:

¿Qué hardware y software utilizan actualmente para gestionar las salidas de garrafones?

¿Cuál es la capacidad de almacenamiento de sus sistemas actuales?

¿Qué sistemas operativos utilizan los equipos involucrados?

Procesos actuales:

¿Cuál es el flujo de trabajo actual desde que se registra un pedido hasta que se entrega el garrafón?

¿Cómo se realiza el inventario de garrafones?

¿Qué métricas se utilizan actualmente para evaluar la eficiencia del proceso?

Integraciones:

¿Qué otros sistemas se integran con el sistema de gestión de garrafones (contabilidad, CRM, etc.)?

¿Qué protocolos de comunicación se utilizan en estas integraciones?

Desafíos técnicos:

¿Cuáles son los principales desafíos técnicos que enfrentan actualmente en la gestión de garrafones?

¿Qué problemas técnicos recurrentes experimentan?

Requisitos de seguridad:

¿Qué medidas de seguridad se implementan actualmente para proteger los datos de los clientes y los registros de ventas?

¿Existen regulaciones específicas que deban cumplirse en términos de seguridad de datos?

Segunda Entrevista: Profundizando en las Necesidades Específicas

Funcionalidades deseadas:

¿Qué funcionalidades específicas le gustaría que tuviera un nuevo sistema de gestión de garrafones (móvil, web, ambas)?

¿Qué tipo de reportes personalizados le serían útiles?

Escalabilidad:

¿Cómo prevé que crecerá su negocio en los próximos años?

¿El sistema debe ser capaz de adaptarse a un aumento en el volumen de datos y usuarios?

Experiencia de usuario:

¿Qué tipo de interfaz de usuario prefiere (simple, detallada, personalizable)?

¿Qué dispositivos utiliza principalmente para gestionar su negocio?

Mantenimiento y soporte:

¿Qué nivel de soporte técnico espera recibir?

¿Cómo le gustaría que se realicen las actualizaciones del sistema?

Presupuesto:

¿Cuál es su presupuesto estimado para la implementación y mantenimiento de un nuevo sistema?

2.5 Viabilidad

La viabilidad es un concepto que se utiliza para evaluar si un proyecto, idea o negocio tiene la capacidad de realizarse con éxito. En otras palabras, es una medida de la posibilidad de que algo se pueda llevar a cabo de manera realista y sostenible. La viabilidad implica un análisis profundo de diversos factores, tanto internos como externos, que pueden influir en el éxito o fracaso de una iniciativa.

La viabilidad según diferentes perspectivas:

Ingeniería: En el ámbito de la ingeniería, la viabilidad se refiere a la posibilidad técnica de construir o desarrollar un producto o sistema. Se evalúan factores como la disponibilidad de materiales, la tecnología necesaria, los recursos humanos y la capacidad de producción..

Un estudio de viabilidad en el marco de la gestión de proyectos normalmente evalúa las siguientes áreas:

• Capacidad técnica: ¿tiene la organización las capacidades y los recursos técnicos para emprender el proyecto?

• Presupuesto: ¿tiene la organización los recursos financieros para llevar a cabo el proyecto y es suficiente el análisis de costes/beneficios del proyecto para justificar el avance del proyecto?

• Riesgo: ¿cuál es el riesgo asociado con la realización de este proyecto?

• Viabilidad operativa: el proyecto, en su alcance propuesto, ¿satisface las necesidades de la organización resolviendo problemas y/o aprovechando las oportunidades identificadas?

• Tiempo: ¿se puede completar el proyecto en un plazo razonable que sea ventajoso para la entidad?

2.6 Factibilidad

En esta sección se indican los posibles riesgos que puedan presentarse en el sistema de registro automático (Tabla 1.2.5. Riesgos), ya sea durante el desarrollo o en la manipulación del mismo. A continuación, se muestra una matriz de riesgos, donde se relacionará la probabilidad y la consecuencia.

Matriz de riesgos

			CONSECUENCIA			
			INSIGNIFICANTE A	MENOR B	MODERADO C	PELIGROSO D
PROBABILIDAD	Frecuente	5				
	Probable	4				
	Ocasional	3				
	Posible	2				
	Improbable	1				

Tabla 1.2.5. Riesgos

Tabla de riesgos

No.	Riesgo	Probabilidad	Consecuencia	Probabilidad
1	Falla del servicio de luz (Problema externo)	Posible	Peligroso	2D
2	Fallo eléctrico del sensor de peso por contacto con agua (corto circuito)	Posible	Peligroso	2D
3	Falla en las herramientas de trabajo	Posible	Moderado	2C
4	Fallo en el marcaje del sensor	Posible	Peligroso	2D
5	Robo del equipo de computo	Posible	Peligroso	2D

Tabla 2.2.5. Tabla de riesgos

En este apartado, se tiene contemplado los riesgos que puedan suceder en el sistema de registro automático, ya sea durante el desarrollo o en su manipulación de dicho sistema. A continuación, se presentan las contenciones y las contingencias.

No.	Riesgo	Contención	Contingencia
1	Falla del servicio de luz (Problema externo)	Instalación de un NoBreake	Verificar el flujo de la energía eléctrica
2	Fallo eléctrico del sensor de peso por contacto con agua (corto circuito)	Usar material aislante al agua	Verificar material resistente al agua
3	Falla en las herramientas de trabajo	Consultar óptimo rendimiento del equipo de trabajo (Computadora / Laptop)	Verificar posible falla del equipo, y solucionarlo lo más pronto posible
4	Fallo en el marcaje del sensor	Reemplazar el sensor de peso por el de respaldo	Contar con un sensor de peso de respaldo
5	Robo del equipo de computo	Verificar si en algún dispositivo de almacenamiento (USB/Disco duro) existe un respaldo de información	Todos los integrantes contarán con un respaldo del proyecto

Tabla 3.2.5. Plan de Contención y Contingencia

Recurso Humano

En el siguiente apartado se hace las observaciones de los recursos que son necesarios para la elaboración del sistema en curso, se determinará el monto de pago de cada uno de los involucrados en el desarrollo del proyecto. En la siguiente tabla se muestran los ingresos de cada miembro del equipo en cuestión.

No. De encargados	Cargo	Mensual	Tiempo (Meses)	Total
1	Dirección del proyecto	$25,000	3	$75,000
1	Programador	$15,000	3	$45,000
1	Diseñador de prototipo	$30,000	3	$90,000
Costo total		$210,000		

Tabla 4.2.6. Recurso Humano

Recursos Financieros

Con el estudio de factibilidad, permite llevar a cabo la evaluación socioeconómica del proyecto o análisis costo y beneficio, la cual permitirá concluir si el proyecto es socioeconómicamente rentable. La inversión corresponde a la etapa en donde se llevan a cabo las obras, los componentes, el equipamiento, el diseño de procesos y procedimientos, la liberación de afectaciones y la gestión de los recursos humanos, es decir, es la etapa en donde se desarrollan todas aquellas acciones necesarias que garanticen, en su momento, la adecuada operación del proyecto.

Recursos Materiales

Son los bienes tangibles o concretos que disponen una empresa u organización con el fin de cumplir y lograr sus objetivos como: instalaciones, materia prima, equipos, herramientas, entre otros. A continuación, se muestran los siguientes equipos que se utilizaran para el desarrollo del sistema de automatización.

MATERIALES	DETALLES	PRECIO	TOTAL
Máquina de cómputo (escritorio)	6 RAM, Windows 10, procesador de 64 bits, disco solido 500 GB	$ 6,000.00 MXN	$ 6,000.00 MXN
Sensor de peso	20 kg (2)	$ 100.00 MXN	$ 100.00 MXN
Arduino Uno	Microcontrolador de código abierto basado en el microchip ATmega328P y desarrollado por Arduino.cc.	$ 250.00 MXN	$ 250.00 MXN
Cautín	Es una herramienta eléctrica que se utiliza para difundir y derretir metales	$ 100.00 MXN	$ 100.00 MXN
Pasta para soldar	Se utiliza para tener una soldadura de mayor calidad	$ 50.00 MXN	$ 50.00 MXN
Estaño	Es un material importante para soldar circuitos eléctricos	$ 100.00 MXN	$ 100.00 MXN
Galga extensiométrica	Una galga extensiométrica es un sensor que mide la deformación, presión, carga, par, posición, etc.	$ 70.00 MXN	$ 70.00 MXN
Placa de ADC (XFW-HX711)	Una placa de ADC es una placa que contiene un conversor analógico-digital (ADC).	$ 40.00 MXN	$ 40.00 MXN
Base Madera	Una base de madera resistente al agua	$100	$100
		Total:	$ 6,910.00 MXN

Tabla 5.2.6. Recursos Tecnológicos

Recurso para el desarrollo

A continuación, en la siguiente tabla se muestra los costos generados por los servicios necesarios para el desarrollo del sistema de registro.

Servicio	Cantidad	Periodo (Mensual)	Monto (Mensual)	Precio
luz	3	3	$250	$750
Internet	3	3	$500	$1500
transporte	3	3	$800	$2400
			Total:	$4,650

Tabla 6.2.6. Recurso para el desarrollo

Resumen de costos

En la siguiente tabla (Tabla 7.2.6. Resumen de costos) presenta un aproximado del presupuesto para desarrollar el sistema de registro automático.

RECURSO	SUB TOTAL
Recurso humano	$210,000
Recursos materiales	$6,870
Recurso para el desarrollo	$4,650
Total:	221,520

Tabla 7.2.6. Resumen de costos

2.7 Impactos

Con la creación de este sistema, se generará más confianza al dueño de la purificadora, ya que, pueden realizar dichos registros sin necesidad de intervenir manualmente y se registran todos los garrafones que saldrán a la venta

2.7.1 Social

Con la creación de este sistema de registro podremos reducir el uso de papel para el proceso de registro manual. Igualmente, los litros podrán registrarse sistemáticamente sin necesidad de hacerlo manualmente. Con estas acciones tendrán beneficios para el cuidado del medio ambiente con la reducción del desperdicio de papel.

2.7.2 Económico

Por parte del cliente, existe un ahorro sustancial monetario, ya que a comparación de otros programas muchos de ellos requieren una inversión mucho mayor para su implementación. Por otro lado, la Purificadora San Antonio tendrá un ahorro económico significativo, debido a que es un proyecto para la acreditación de la titulación de los alumnos de la carrera de Ingeniería en Sistemas Computacionales, dicho esto, el desarrollo del proyecto tiene poca o nula inversión.

2.7.3Tecnologico

La purificadora San Antonio reflejara el impacto tecnológico al implementar un sistema de registro sistemático de salida de litraje que tiene hardware y software que le permita registrar los garrafones que salen del negocio, agilizando tiempos y mejorara la organización para un mejor conteo de garrafones vendidos durante el día.

3.0 Metodología de desarrollo

3.1 METODOLOGIA SCRUM

La metodología Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes. Dicho en otras palabras: Scrum sirve para mejorar el trabajo colaborativo entre equipos

Al implementar dicha metodología, se deberá de cubrir las siguientes fases:

A. Fase de Análisis

En esta fase se llevan a cabo reuniones con el cliente, para determinar la problemática y con base a ello determinar los requerimientos que necesita el cliente y se adecuen al sistema que solicita, se estima el tiempo de entrega del proyecto y los diferentes riesgos que puedan surgir en el sistema.

B. Fase Diseño

En esta fase, se diseña el prototipo, se realiza la diagramación lógica para la ejecución del sistema de registro automático.

C. Fase Codificación

En esta fase se lleva a cabo la programación lógica del sistema de registro automático, la cual consiste en la implementación de los diferentes elementos visuales que con tendrá el sistema de registro automático bajo un lenguaje de programación C++ así mismo se usará.

En la parte lógica programable, se implementará la funcionalidad del sistema de registro automático, bajo un esquema de programación C++ y visual basic.

D. Fase Pruebas

Tal como su nombre lo indica, es la fase final en la que se realizan las pruebas de funcionamiento para verificar el rendimiento óptimo del sistema de registro sistemático, y corregir los posibles errores.

Justificación de la implementación de la Metodología Scrum

La metodología Scrum permite agilizar los procesos de entrega de cada apartado que abarca al proyecto, cumpliendo con las fechas establecidas y visualizar los resultados de cada actividad para avanzar progresivamente en el desarrollo del sistema de registro automático.

Dichos resultados se ven reflejados ya que la metodología contiene distintos roles que conforman los integrantes del equipo, cumpliendo cada uno con sus respectivas actividades y proponer ideas para el siguiente Sprint. Scrum permite la comunicación constante y la interacción con todos los integrantes del equipo para poder determinar lo que se está realizando, las actividades que aún faltan por hacer y las actividades que ya han sido concluidas.

Además, Scrum permite el intercambio de roles para agilizar los procesos y no estancarse durante el desarrollo del proyecto.

Profundizando en los roles y responsabilidades:

- El rol del Product Owner: "El Product Owner es el responsable de definir el producto y maximizar su valor. Actúa como el vínculo entre el equipo de desarrollo y los stakeholders, priorizando las funcionalidades y asegurando que el producto se alinea con las necesidades del negocio."
- El rol del Scrum Master: "El Scrum Master es el facilitador del equipo Scrum. Se asegura de que el equipo siga los principios de Scrum, elimina los obstáculos que impidan el progreso y fomenta un ambiente de trabajo colaborativo."
- El equipo de desarrollo: "El equipo de desarrollo es autoorganizado y responsable de crear el producto incrementable. Los miembros del equipo colaboran estrechamente para completar las tareas asignadas."

Destacando la importancia de la comunicación:

- Reuniones diarias: "Las reuniones diarias, o Daily Scrum, son fundamentales para sincronizar al equipo y mantener a todos informados sobre el progreso. Estas reuniones cortas y enfocadas ayudan a identificar y resolver cualquier impedimento que pueda surgir."
- Retrospectivas: "Las retrospectivas son eventos regulares donde el equipo reflexiona sobre lo que ha funcionado bien, lo que podría mejorar y cómo hacerlo. Estas reuniones son clave para la mejora continua."

La metodología Scrum ha demostrado ser una herramienta invaluable para agilizar el desarrollo de sistemas de registro automático. Al dividir el proyecto en Sprints cortos e iterativos, Scrum permite entregar valor al cliente de manera incremental y adaptarse rápidamente a los cambios en los requisitos.

Dichos resultados se obtienen gracias a la estructura y los roles definidos en Scrum. El Product Owner establece la visión del producto y prioriza las funcionalidades, mientras que el Scrum Master facilita el proceso y asegura que el equipo siga las prácticas de Scrum. El equipo de desarrollo, autoorganizado, trabaja de manera colaborativa para crear incrementos de producto funcionales al final de cada Sprint.

La comunicación constante y abierta es un pilar fundamental de Scrum. Las reuniones diarias, las retrospectivas y la transparencia en el trabajo permiten a todos los miembros del equipo estar alineados y tomar decisiones informadas. Además, la posibilidad de intercambiar roles fomenta la versatilidad y el aprendizaje continuo.

Los pilares de la transparencia y la adaptabilidad

Los **artefactos Scrum** son elementos clave que proporcionan transparencia y una visión compartida del trabajo a realizar en un proyecto. Estos artefactos son como las piezas de un rompecabezas que, al encajar de manera correcta, permiten al equipo Scrum tener una comprensión clara del progreso y los objetivos del proyecto.

Los tres artefactos principales de Scrum son:

1. Product Backlog

- Definición: Es una lista ordenada de todas las características, funcionalidades y requisitos que se desean en el producto final.
- Características:
    - Priorizado: Las items del Product Backlog están ordenados por prioridad, lo que permite al equipo enfocarse en las características más valiosas.
    - Evolutivo: El Product Backlog es un documento vivo que se actualiza constantemente para reflejar los cambios en los requisitos del cliente y las prioridades del negocio.
    - Detallado: Cada item del Product Backlog debe ser lo suficientemente detallado para que el equipo de desarrollo pueda estimar el esfuerzo necesario para implementarlo.

2. Sprint Backlog

- Definición: Es un conjunto de items seleccionados del Product Backlog que el equipo se compromete a completar durante un Sprint específico.
- Características:
    - Trabajo a realizar: El Sprint Backlog detalla las tareas específicas que deben realizarse para completar los items seleccionados del Product Backlog.
    - Estimado: Cada tarea del Sprint Backlog está estimada en términos de esfuerzo.
    - Comprometido: El equipo se compromete a completar todo lo que está en el Sprint Backlog al final del Sprint.

3. Incremento

- Definición: Es el resultado tangible de un Sprint. Es un producto potencialmente entregable que aumenta el valor del producto final.
- Características:
    - Funcional: El incremento debe ser funcional y usable, aunque puede no ser completamente terminado.
    - Potencialmente entregable: El incremento está listo para ser entregado al cliente si así se desea.
    - Valor: El incremento debe proporcionar valor al cliente, aunque sea incremental.

3.2 Grafica de Gantt

El diagrama de Gantt es una herramienta de representación de actividades o tareas que pretende dar una visión generalizada sobre el tiempo dedicado a cada actividad contemplada de forma independiente dentro de un proceso.

En el siguiente diagrama de Gantt, se visualiza el tiempo que corresponde a cada una de las fases al implementar la metodología Scrum.

Capitulo 1					
Planteamiento del problema	Riesgo bajo	Eder	80%	01/08/2024	3
Onjetivo General	Hito	Braulio	95%	11/01/2024	1
Onbjetivo Especifico	Riesgo medio	Jhonathan	100%	29/07/2024	10
Justificación	Riesgo bajo	Braulio	95%	21/08/2024	1
Alcance y limitaciones	Riesgo bajo	eder	70%	07/08/2024	6
Capitulo 2 Fundamento Teorico					
Estado del arte	Riesgo alto		98%	07/08/2024	3
Definición terminos bsicos	Riesgo medio		100%	03/03/2024	5
Marco teorico	Riesgo medio		100%	14/08/2024	4
Estudio del mercado	egún lo previsto		100%	16/08/2024	1
Viablidad	Riesgo bajo		100%	08/03/2024	3
Factibilidad	Riesgo bajo		100%	09/03/2024	2
Inpactos	Riesgo medio		95%	10/03/2024	24
Capitulo 3 Metodologia de desarrollo					
Grafica de Gant	egún lo previsto		92%	15/03/2024	4
Fase de analisis	Riesgo medio		92%	18/03/2024	14
Levantamiento de requerimientos	egún lo previsto		100%	01/03/2024	6

3.3 Fase de Análisis

3.3.1 Levantamiento de Requerimientos

Levantamiento de Requerimientos

Se llevaron a cabo entrevistas técnicas con el dueño del negocio, para determinar la problemática existente, conocer sus necesidades, con la finalidad de compartir ideas y determinar una posible solución al problema, para agilizar el registro de salida de garrafones.

Sprint Backlog

Metodologías ágiles: listas de tareas de la iteración					
(Sprint backlog)					
Identificador (ID) de item de product backlog	Enunciado del item de product Backlog	Tarea	Dueño / Voluntario	Estatus	Horas Estima-das totales
EBJ-001	**Braulio:** Documentador, diseñador de la interfaz, Programador; Documentar datos y antecedentes con la finalidad de generar un documento que conforme las partes del sistema y funcionalidades , y implantar el diseño correctamente	Investigar los impactos ambientales, y sociales plantear la metodología scrum y los diagramas de Gantt	Dueño	Hecho	20
		Establecer los requerimientos del proyecto con base al protocolo de Tesis de Ingeniería	Dueño/ voluntario	Hecho	15

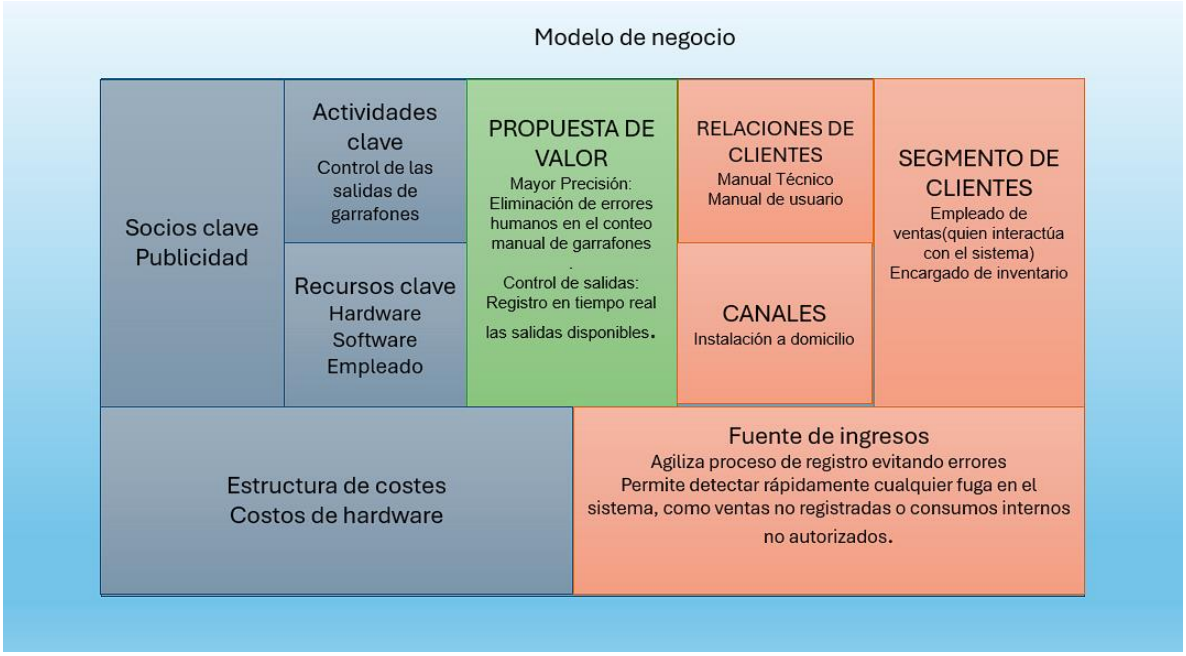
		Plantear y documentar los daily scrum			8
		Programar la interfaz	Dueño / voluntario	Hecho	50
	Eder: Documentador, intermediario con el cliente; Documentar para una mejor implementación del proyecto y portador de la información al respecto con el cliente y sus intereses, investigación de viabilidad del proyecto	Planteamiento del problema	Dueño	Hecho	5
		Redacción del objetivo general	Dueño	Hecho	5
		Redacción y investigación de los impactos económicos	Dueño	Hecho	5
		Redacción y investigación de marco teórico	Dueño / voluntario	Hecho	8
		Redacción de los objetivos específicos	Dueño	Hecho	5
		Implementación de sprint	Dueño	Hecho	10
EBJ-003	**Jonathan:** Documentador; Para mejorar la implementación del proyecto y un mejor orden, investigación	Investigar y complementar Marco teórico	Dueño / voluntario	Hecho	6

	de factibilidad del proyecto				
		Investigación y redacción de los impactos tecnológicos	Dueño	Hecho	5
		Diagramas de casos de uso	Dueño / voluntario	Hecho	8
		Diagramas de flujo	Dueño / voluntario	Hecho	8
		Investigación de complementos (c++, Arduino uno, Visual Basic)	Dueño / voluntario	Hecho	6
		sketch	Dueño/voluntario	Hecho	15
		wireframes	Dueño	Hecho	15

3.3.2 Descripción del Proceso de Negocio

Descripción del Proyecto

El sistema de registro estará desarrollado en C++ este siendo el lenguaje intermediario entre Arduino y la interfaz de usuario, Arduino Uno, Visual Basic versión más reciente, Excel versión 24-06 HTML. Lo que nos ayuda a tener un mejor diseño y una mayor interacción en el sistema para el cliente. Unas de sus características principales:



Actividades (Azul)

Control de calidad: Asegurar que los productos o servicios cumplan con los estándares establecidos.

Estudios de mercado: Investigar y analizar el mercado para entender las necesidades y preferencias de los clientes.

Publicidad: Promocionar productos o servicios para atraer a más clientes.

Reclamación de personal: Proceso de atraer y seleccionar nuevos empleados.

Modelo de Negocio (Verde)

Mayor Precisión: Eliminación de errores humanos en el conteo.

Control de salidas: Registro en tiempo real de las salidas de productos.

Costos de hardware reducidos: Minimización de los costos asociados con el hardware necesario para el sistema.

Relaciones con Clientes (Naranja)

Manual Técnico: Proporcionar documentación técnica para los clientes.

Seguimiento de Ventas: Registro y seguimiento de las ventas en el sistema.

Compromiso en la entrega del inventario: Asegurar que el inventario se entregue de manera oportuna y precisa.

Sección de Pie de Página (Gris)

Estructura de costes:

Permitir acceso remoto: Facilitar el acceso al sistema desde ubicaciones remotas.

Sistema como vendedor: Detectar rápidamente cualquier fuga en el algoritmo o proceso no registrado, así como errores no autorizados en el conteo.

3.3.4 Historias de Usuario (Requerimientos Funcionales)

Para que el usuario pueda acceder y utilizar el sistema de registro, requerirá lo siguiente:

• ordenador (Laptop, computador de escritorio).

• paquetería office (Excel).

• Para hacer uso del sistema de registro, el usuario deberá ingresar su contraseña.

3.3.5 Requerimientos No Funcionales

Requerimientos No Funcionales

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. No hablan de “lo que” hace el sistema, sino de “cómo” lo hace.

Especifican los criterios que debe cumplir para que sea adecuado para su uso (Fitness for-use).

- El usuario debe proporcionar una contraseña para su acceso.
- La interfaz gráfica será de fácil comprensión para el usuario.
- El sistema deberá estar alojado en una hoja de Excel que pueda manejar con facilidad el usuario.
- El sistema debe funcionar en cualquier ordenador.

Debe adaptarse a la resolución de pantalla de la laptop

3.3.6 Requerimientos del Sistema

Al desarrollar el sistema, se requerirá las siguientes herramientas tecnológicas para determinar su estructura y su funcionalidad, dichas herramientas son las siguientes:

• Arduino UNO (Hardware)

• Visual Basic (Diseño de interfaz).

• Excel (hoja de registro).

• C++ (Programación lógica).

Procesador: Intel Core i5 de 11ª o 12ª generación, o AMD Ryzen 5 de la serie 5000.

Memoria RAM: 6 GB como mínimo.

Almacenamiento: 256 GB SSD (Unidad de Estado Sólido).

Pantalla: Resolución de 1920 x 1080 píxeles (Full HD).

Puertos: Al menos un puerto USB-C, varios puertos USB-A, y un puerto HDMI.

Sistema Operativo: Windows 10 o la última versión o posterior

Batería: Duración mínima de 8 horas.

3.3.7. Estudio de Usabilidad

Para que el usuario pueda acceder y utilizar el sistema, requerirá lo siguiente:

- ordenador (Laptop, computador de escritorio).
- paquetería office (Excel).
- Para hacer uso del sistema de registro, el usuario deberá ingresar su contraseña.

3.3.7.1. Diseño de interfaz de usuario

3.3.7 Estudio de Usabilidad (Herramientas de acuerdo con las necesidades del levantamiento)

Para que el usuario pueda acceder y utilizar el sistema, requerirá lo siguiente:

- ordenador (Laptop, computador de escritorio).
- paquetería office (Excel).
- Para hacer uso del sistema de registro, el usuario deberá ingresar su contraseña.

3.4 Fase Diseño

La Fase de Diseño en una Tesis de Sistema: Un Puente Entre la Teoría y la Práctica

La fase de diseño en una tesis que desarrolla un sistema es crucial, ya que en ella se define la estructura, la funcionalidad y la apariencia del sistema que se va a implementar. Es el momento de transformar los requerimientos y análisis realizados en etapas anteriores en una representación tangible del sistema.

¿Qué implica la fase de diseño?

En esta etapa se abordan aspectos como:

Arquitectura del sistema: Se define la estructura general del sistema, incluyendo los componentes principales, su relación entre sí y cómo se comunican.

Diseño de la interfaz de usuario: Se crea una interfaz intuitiva y fácil de usar para que los usuarios interactúen con el sistema de manera eficiente.

Diseño de la base de datos: Se define la estructura de la base de datos que almacenará la información del sistema, incluyendo las tablas, campos y relaciones.

Diseño de los módulos: Se descompone el sistema en módulos más pequeños y manejables, cada uno con una función específica.

Diseño de los algoritmos: Se diseñan los algoritmos que realizarán los cálculos y procesos necesarios para el funcionamiento del sistema.

En el caso específico de un sistema de registro con los requerimientos mencionados, el diseño podría incluir:

Diseño de la interfaz de inicio de sesión: Un formulario donde el usuario ingrese su nombre de usuario y contraseña.

Diseño de la interfaz principal: Un menú o dashboard que permita al usuario navegar por las diferentes funcionalidades del sistema.

Diseño de la base de datos: Una tabla para almacenar los usuarios, con campos como nombre de usuario, contraseña y otros datos relevantes.

Diseño de los módulos de registro: Módulos para registrar nuevas entradas de datos, modificar registros existentes y eliminar registros.

Diseño de los reportes: Módulos para generar reportes personalizados con los datos almacenados en el sistema.

3.4.1 Diagrama de Procesos General

Un **Diagrama de Procesos General** es una herramienta visual que representa la secuencia de actividades, decisiones y flujos de información dentro de un proceso. Este tipo de diagrama es esencial para entender cómo se desarrollan los procesos en una organización, identificar ineficiencias y mejorar la eficiencia operativa. Los componentes clave incluyen actividades, decisiones, flujos de información, y entradas y salidas. Los beneficios de utilizar estos diagramas son múltiples: ayudan a documentar procesos para mejorar la comprensión y la capacitación, facilitan la estandarización de procesos para lograr eficiencia y repetibilidad, y permiten identificar cuellos de botella y pasos innecesarios para optimizar el proceso.

INICIO

Administrador

Recibe el garrafón vacío de 10 o 20 litros.

Realiza el lavado del garrafón.

Realiza el secado del garrafón.

Coloca el garrafón sobre la base del prototipo.

Realiza el llenado del garrafón:

Sistema

Recibe el peso del garrafón lleno.

Realiza el pesaje del garrafón.

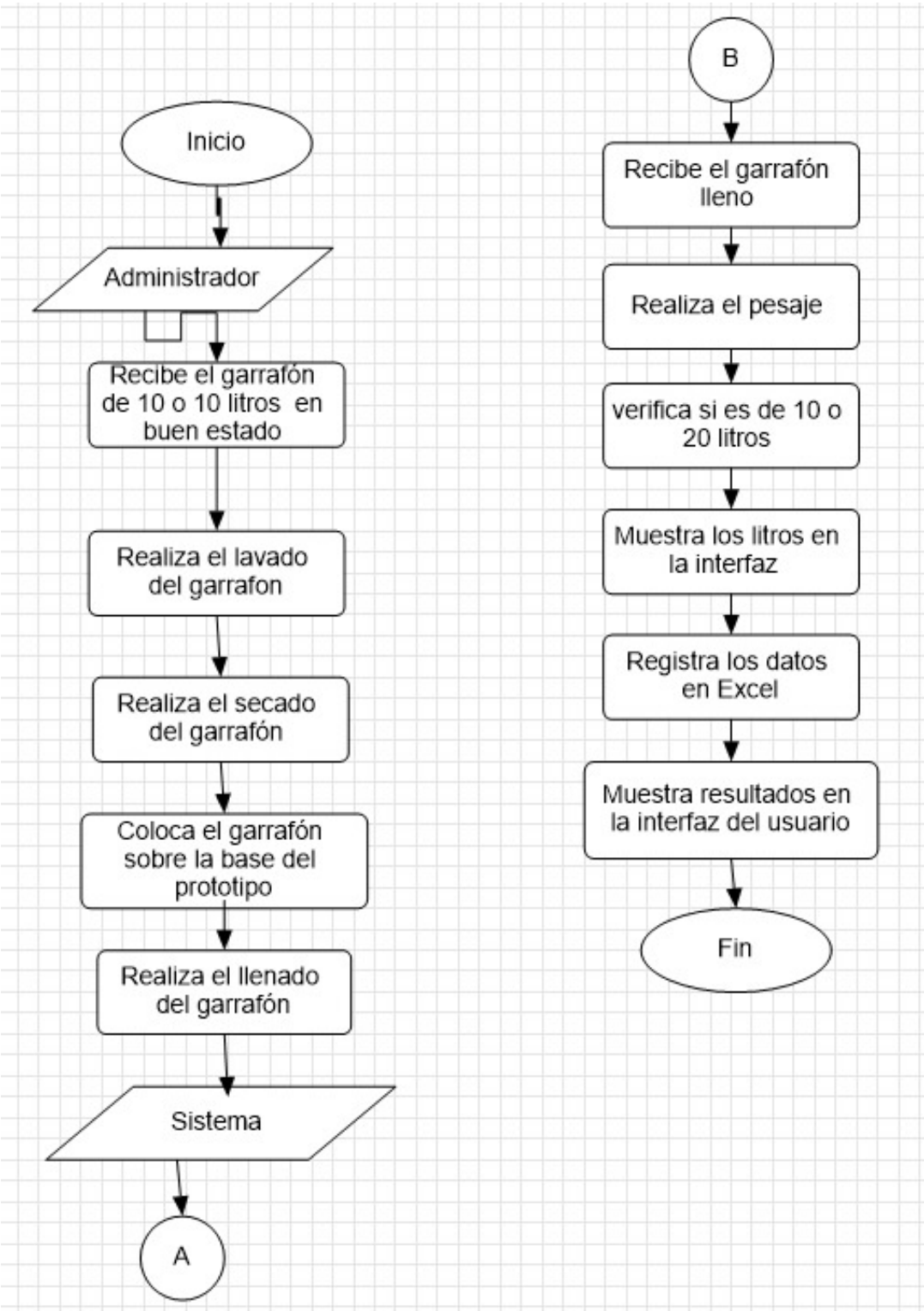
Verifica si es de 10 o 20 litros.

Muestra los litros en la interfaz.

Registra los datos en Excel.

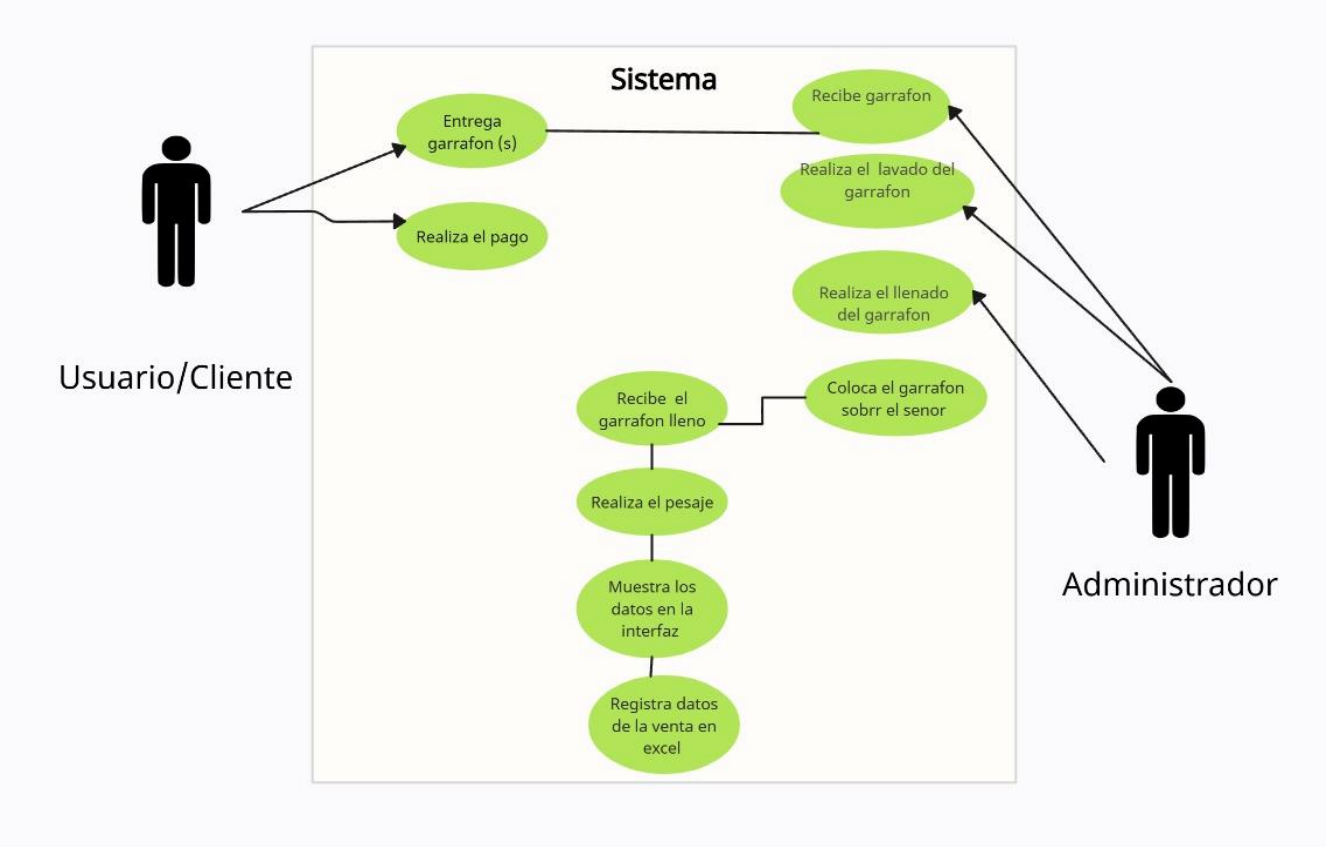
Muestra los resultados en la interfaz del usuario.

FIN


Inicio
Administrador
Recibe el garrafón de 10 o 10 litros en buen estado
Realiza el lavado del garrafon
Realiza el secado del garrafón
Coloca el garrafón sobre la base del prototipo
Realiza el llenado del garrafón
Sistema
A
B
Recibe el garrafón lleno
Realiza el pesaje
verifica si es de 10 o 20 litros
Muestra los litros en la interfaz
Registra los datos en Excel
Muestra resultados en la interfaz del usuario
Fin

3.4.2 Diagrama de Casos de Uso General

Un **Diagrama de Casos de Uso General** es una herramienta visual utilizada en el análisis y diseño de sistemas para representar las interacciones entre los usuarios (actores) y el sistema. Este tipo de diagrama es parte del Lenguaje de Modelado Unificado (UML) y es fundamental para entender las funcionalidades del sistema desde la perspectiva del usuario final.



3.4.3 Especificación de Casos de Uso

Usuario/Cliente:

1. **Entrega el paquete**: El usuario entrega el paquete al sistema.
2. **Recibe el paquete**: El usuario recibe el paquete del sistema.
3. **Realiza el pago**: El usuario efectúa el pago correspondiente.
4. **Muestra los datos en la interfaz**: El usuario puede ver los datos en la interfaz del sistema.

Sistema:

1. **Realiza el pago**: El sistema procesa el pago realizado por el usuario.
2. **Recibe garantía**: El sistema recibe y gestiona la garantía del paquete.
3. **Realiza el lavado del garrafón**: El sistema se encarga de lavar el garrafón.

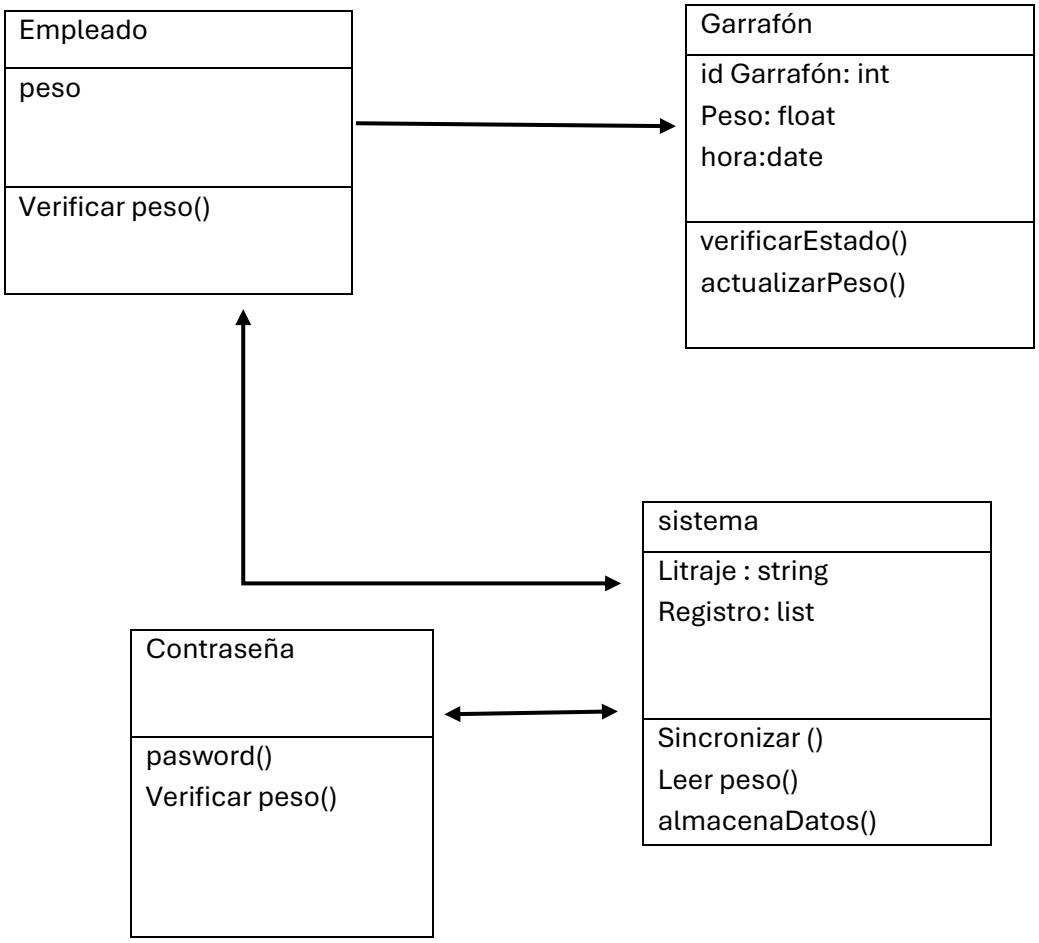
Administrador:

1. **Coloca el garrafón sobre la estación de llenado**: El administrador coloca el garrafón en la estación de llenado para su procesamiento.

Acciones Comunes:

1. **Registra datos del evento actual**: Tanto el usuario como el sistema registran los datos del evento actual.
2. **Realiza el pesaje**: El sistema realiza el pesaje del paquete o garrafón.

3.4.4 Diagrama de Clases

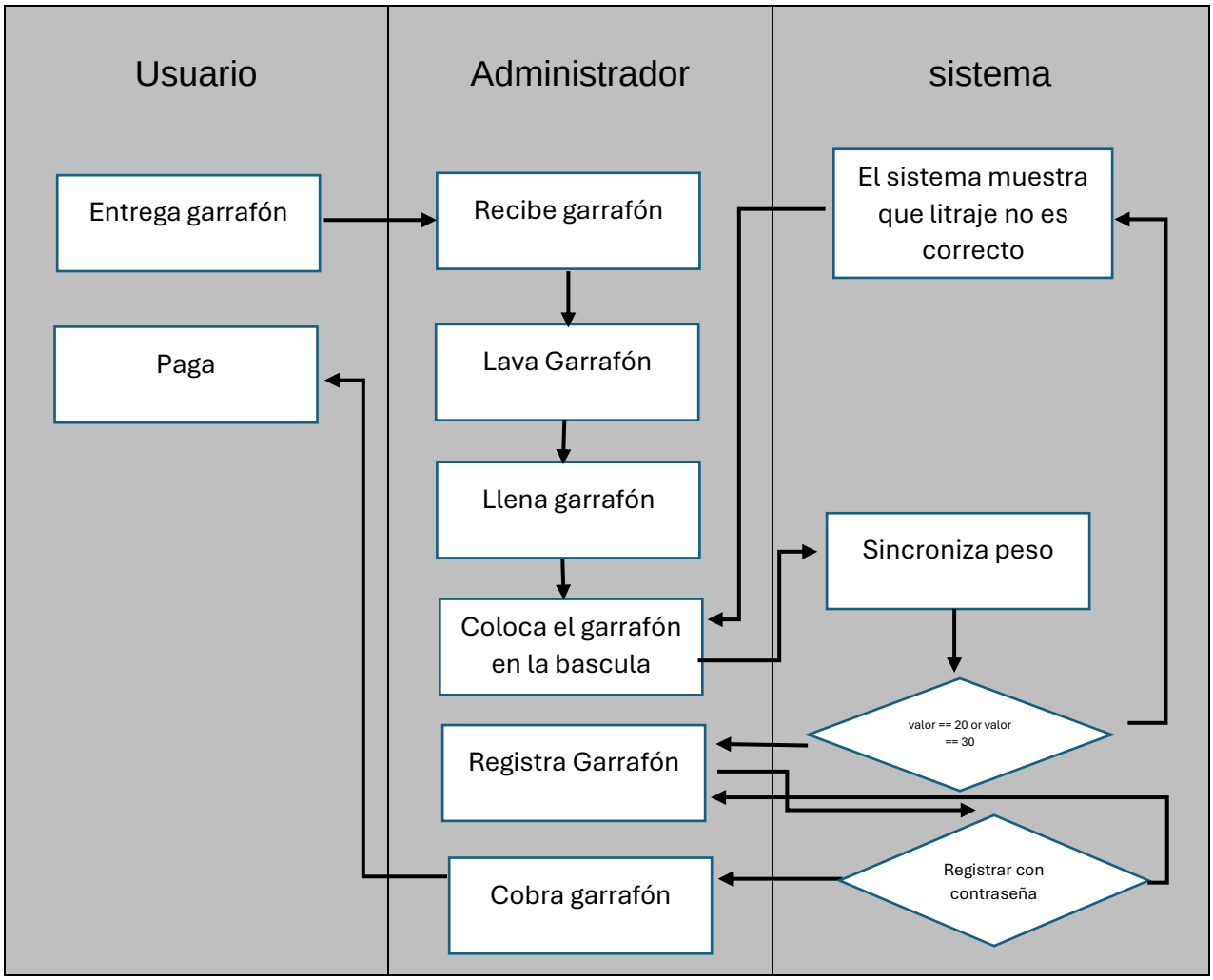


3.4.5 Diagrama de Secuencia

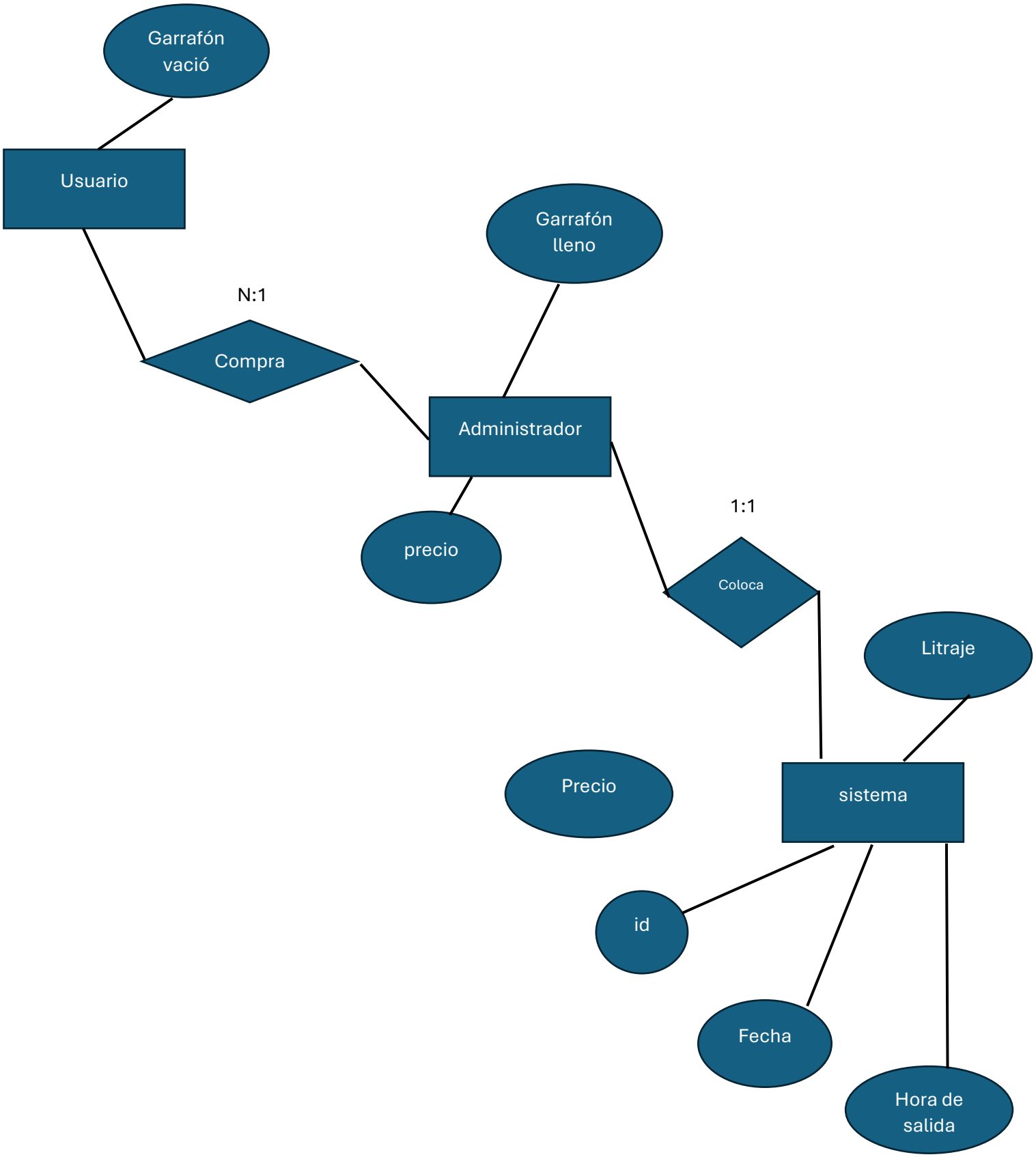
Un **diagrama de secuencia** es un tipo de diagrama utilizado en el modelado de sistemas, particularmente en el lenguaje de modelado unificado (UML, por sus siglas en inglés). Representa cómo interactúan los objetos en un sistema a lo largo del tiempo para llevar a cabo un proceso o cumplir con un requisito

USUARIO	INTERFAZ	PESAJE	GALGAS	EXCEL
Interactúa con la interfaz				
	Recibe las entradas del usuario y muestra los resultados.			
		Procesa las señales de las galgas.		
			Sensores que miden el peso.	
				Almacena los datos de pesaje.

3.4.6 Diagrama de Actividades

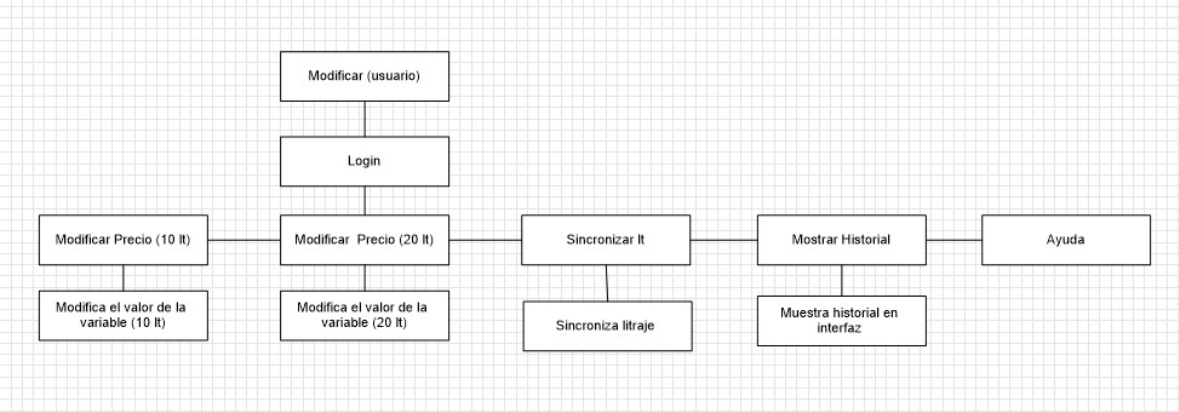


3.4.7 Diagrama Entidad Relación

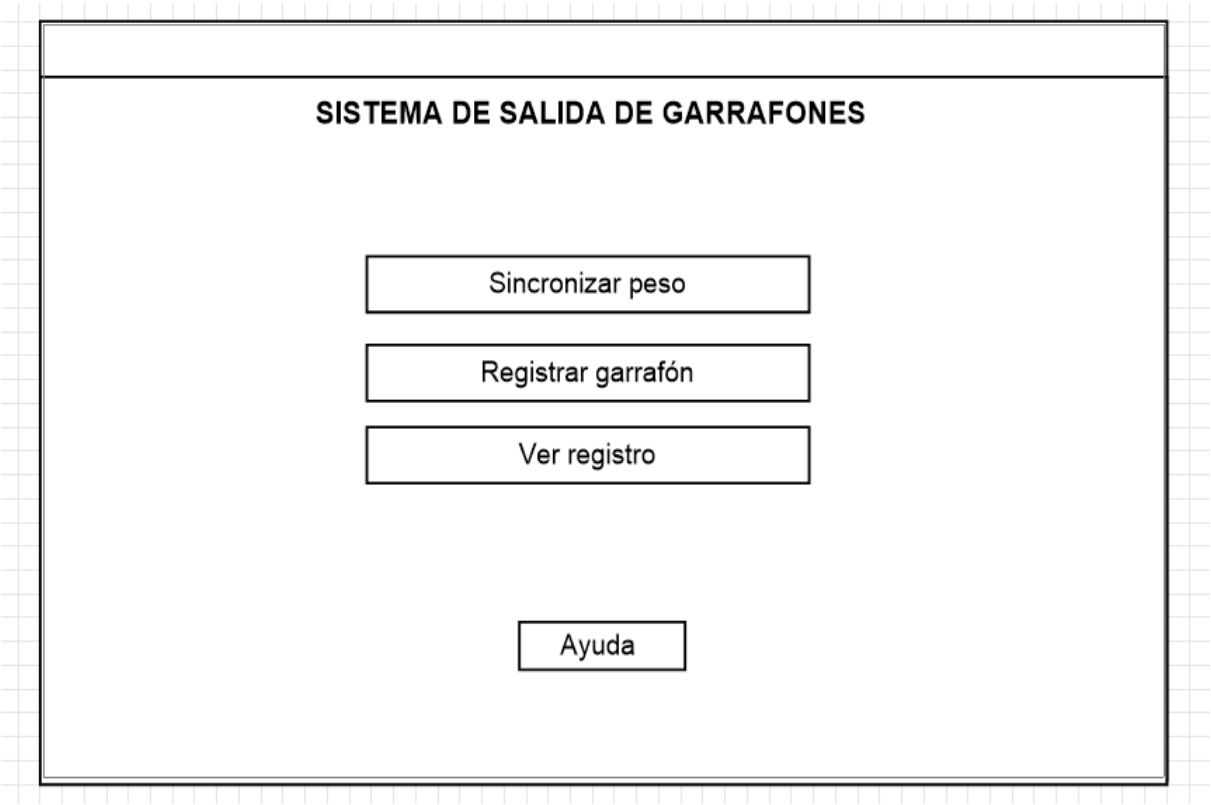


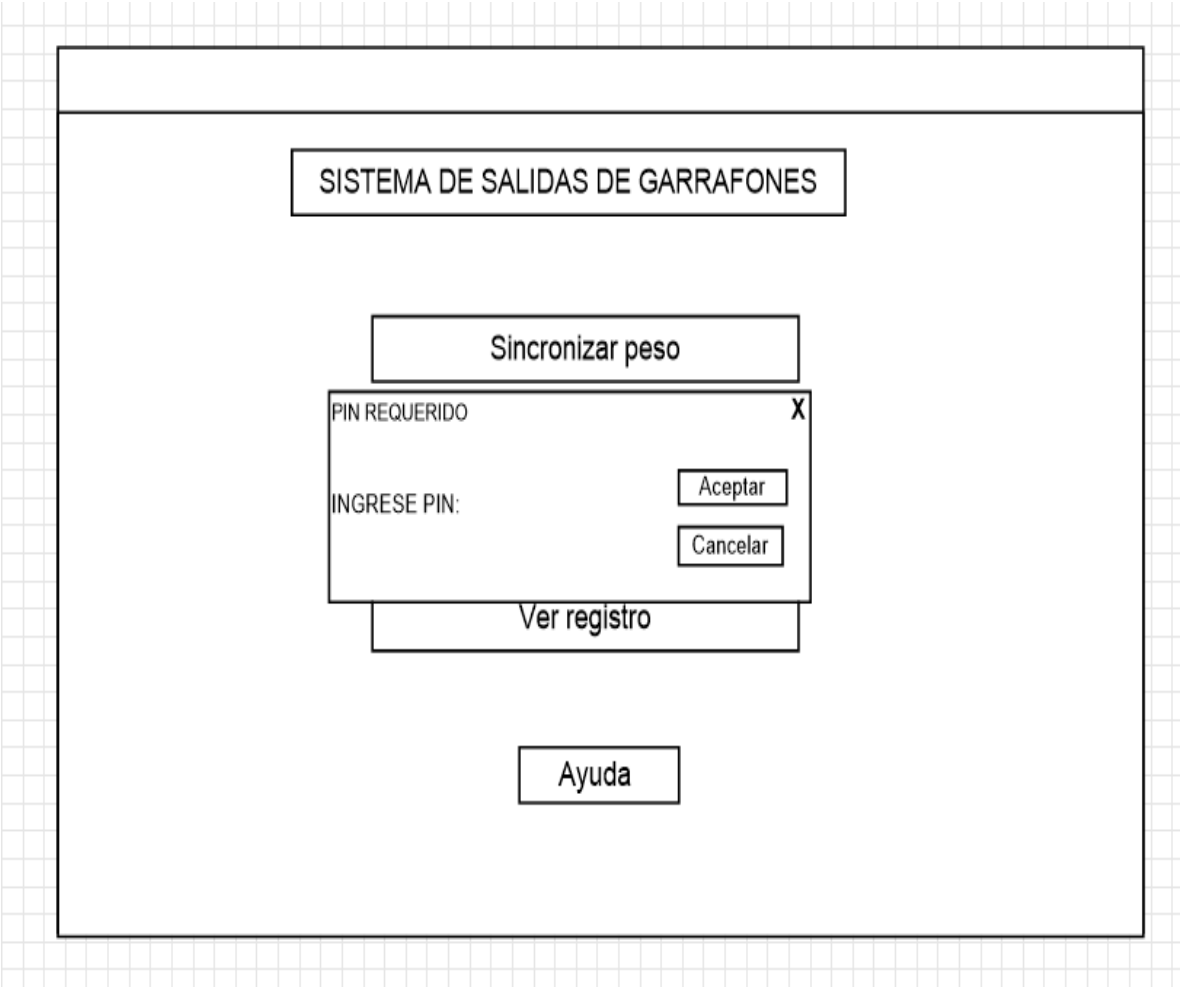
3.4.9 Sitemap (Arquitectura de información)

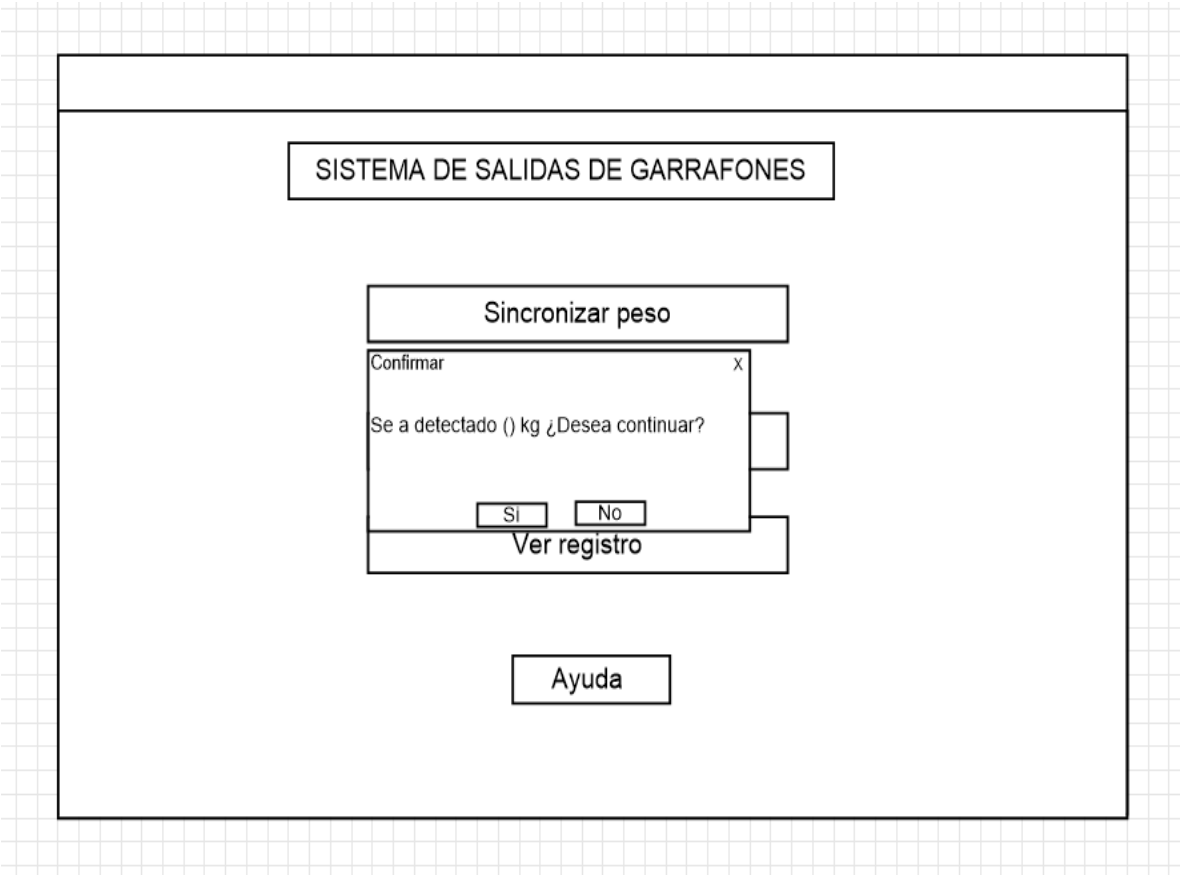
Un **sitemap** (o mapa del sitio) es una herramienta crucial en el desarrollo y mantenimiento de sitios web. Sirve como un esquema o lista jerárquica de las páginas que componen un sitio web. Existen dos tipos principales de sitemaps: sitemaps para usuarios y sitemaps para motores de búsqueda, y cada uno tiene un propósito específico



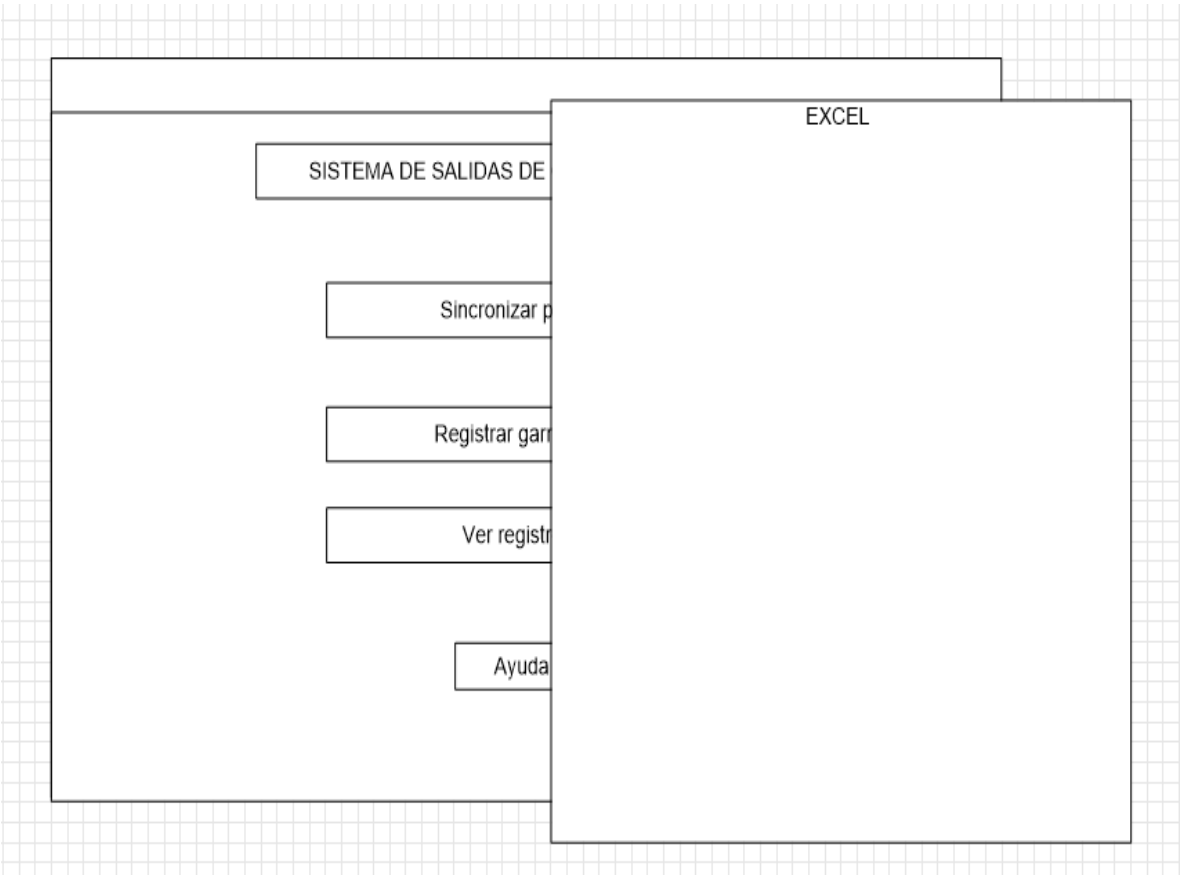
3.5 Wilframes

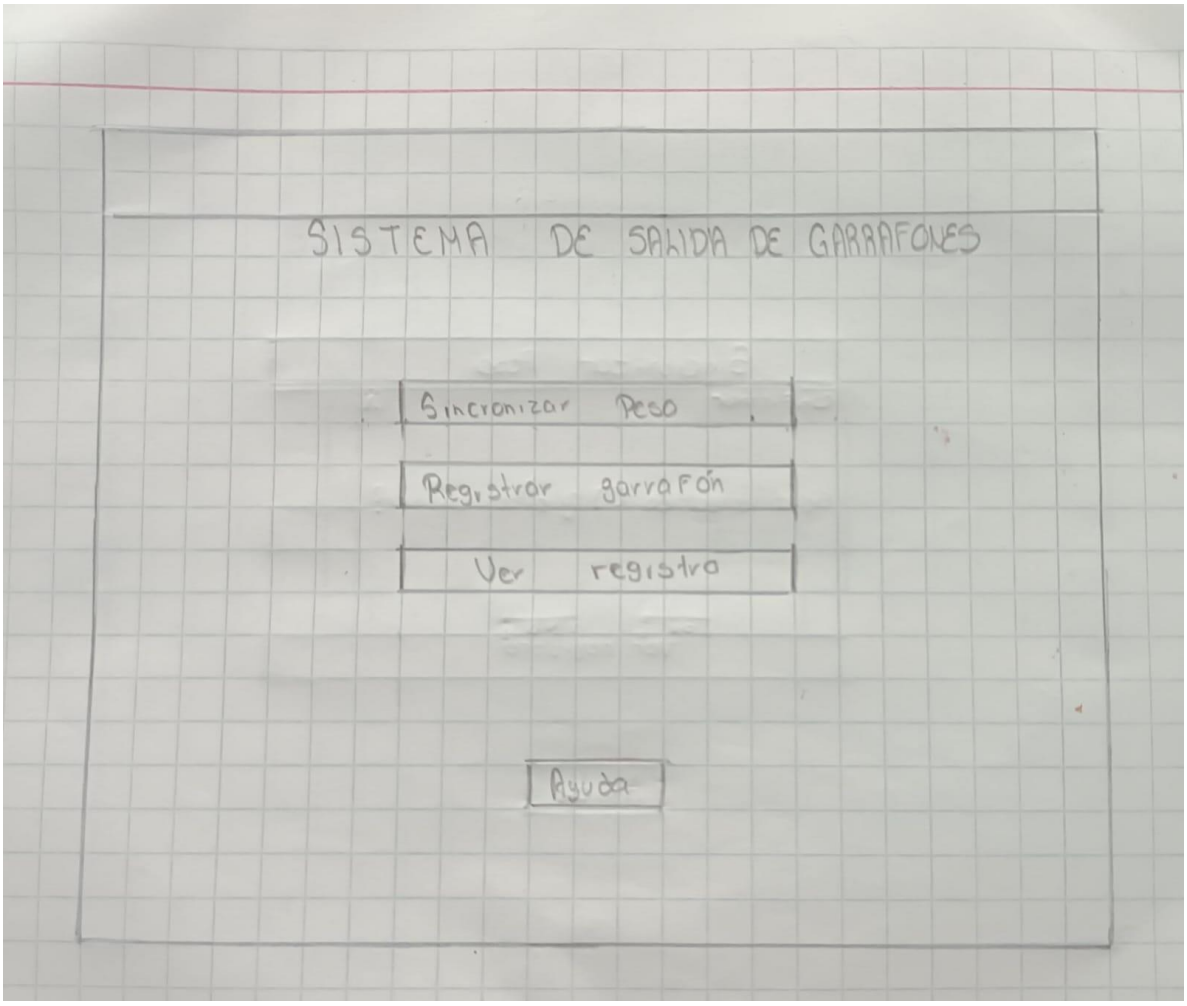


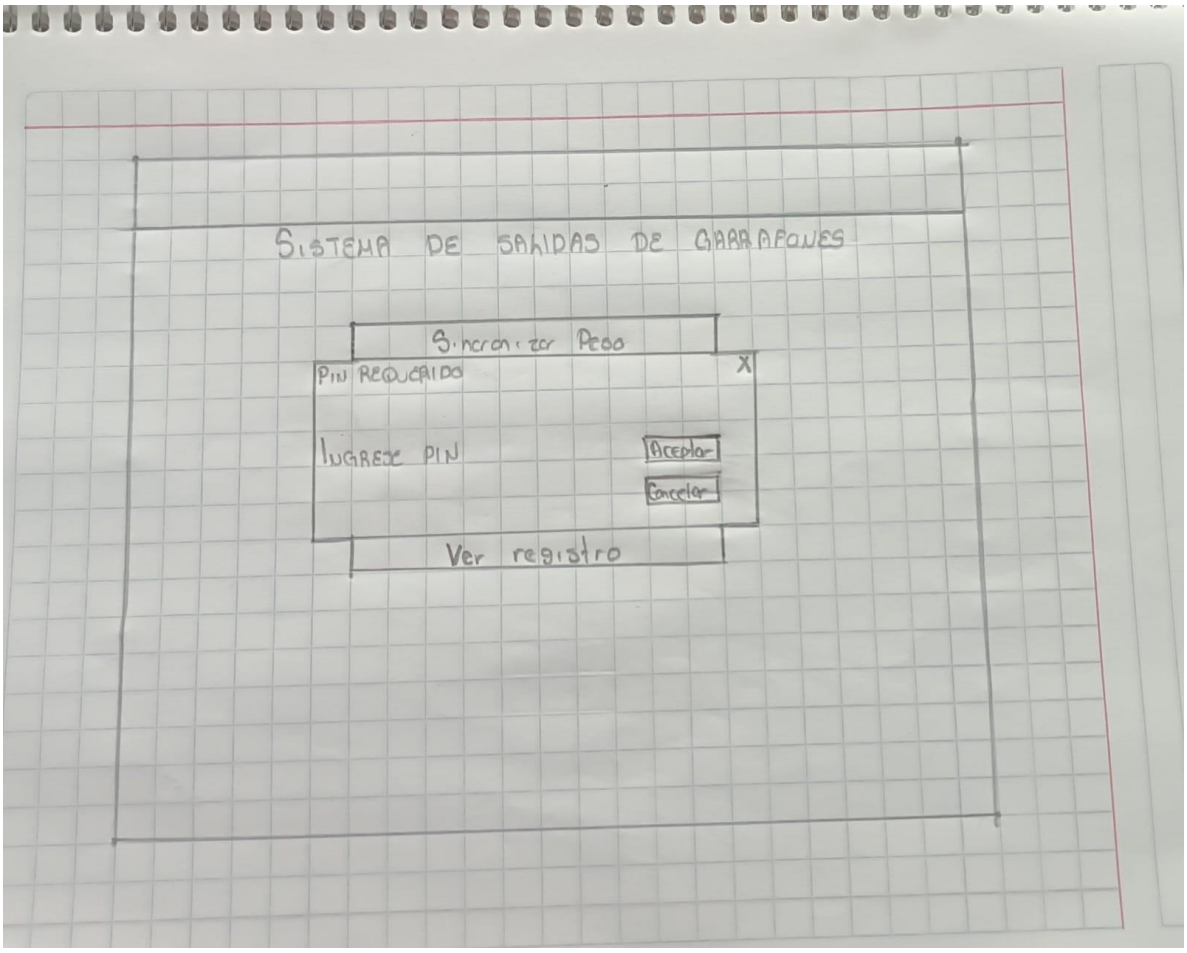

SISTEMA DE SALIDAS DE GARRAFONES
Sincronizar peso
PIN REQUERIDO
X
INGRESE PIN:
Aceptar
Cancelar
Ver registro
Ayuda

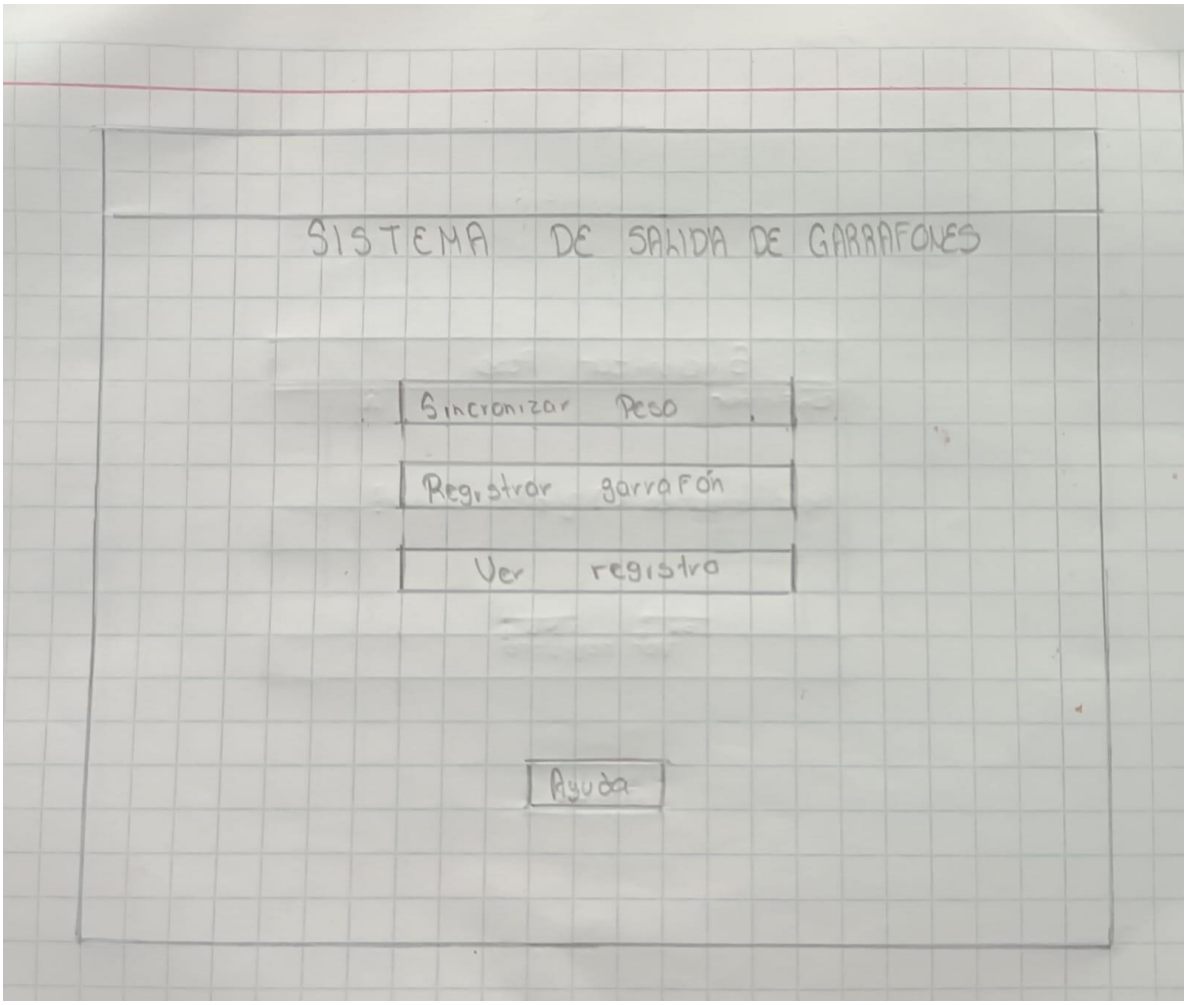

SISTEMA DE SALIDAS DE GARRAFONES
Sincronizar peso
Confirmar
X
Se a detectado () kg ¿Desea continuar?
Si
No
Ver registro
Ayuda

( )


EXCEL
SISTEMA DE SALIDAS DE
Sincronizar p
Registrar garr
Ver registr
Ayuda


SISTEMA DE SALIDA DE GARRAFONES
Sincronizar Peso
Registrar garrafón
Ver registro
Ayuda


SISTEMA DE SALIDAS DE GARRAFONES
Sincronizar Pedo
PIN REQUERIDO
X
INGRESE PIN
Aceptar
Cancelar
Ver registro


SISTEMA DE SALIDA DE GARRAFONES
Sincronizar Peso
Registrar garrafón
Ver registro
Ayuda

3.5.1 Look and Feel

Look and Feel del Sistema de Salidas de Garrafones

- **Fondo:** Azul claro, evocando la sensación de agua y frescura.
  - **Justificación:** Según estudios de psicología del color, los tonos azules están asociados con la calma, confiabilidad y frescura, atributos importantes para un sistema que maneja productos relacionados con agua (Smith, 2019).
- **Botones:** Azul oscuro con texto blanco para un buen contraste y legibilidad.
  - **Justificación:** El contraste alto entre el azul oscuro y el blanco mejora la legibilidad, lo cual es crucial en interfaces que requieren interacción rápida y precisa (Nielsen, 2021).
- **Texto:** Negro o gris oscuro para el contenido principal, blanco para los botones.
  - **Justificación:** Los colores neutros como el negro y el gris oscuro permiten que el contenido sea fácil de leer y no distraiga al usuario (Brown, 2020).
- **Tipografía:**
- **Fuente principal:** Sans-serif, como Arial o Helvetica, para una apariencia limpia y moderna.
  - **Justificación:** Las fuentes sans-serif, como Arial y Helvetica, son consideradas modernas y fáciles de leer en pantallas digitales debido a su diseño limpio y sin adornos (Lupton, 2018). Estas fuentes son ampliamente utilizadas en interfaces de usuario debido a su legibilidad y simplicidad.
- **Tamaño de fuente:**
- **Títulos:** 24px
  - **Justificación:** El tamaño de 24px para los títulos asegura que la información principal destaque sin ser abrumadora (Krug, 2020).
- **Subtítulos:** 18px
  - **Justificación:** Un tamaño ligeramente menor para subtítulos mantiene la jerarquía visual clara (Garrett, 2019).
- **Texto de botones:** 16px
  - **Justificación:** Un tamaño de 16px en botones es adecuado para la interacción, según las pautas de accesibilidad web (W3C, 2020).
- **Texto normal:** 14px

- **Justificación:** Este tamaño es comúnmente recomendado para el texto del cuerpo en interfaces de usuario, facilitando la lectura prolongada (Norman, 2021).

- **Elementos de la Interfaz:**
- **Logo:** En la parte superior, centrado, con el texto “BYRAU”.
  - **Justificación:** Colocar el logo en la parte superior y centrado es una convención de diseño que asegura que la marca sea visible y fácilmente reconocible al cargar la página (Lidwell, 2019).
- **Botones:**
  - “Sincronizar garrafón”
  - “Registrar Garrafón”
  - “Ver registro”
  - “AYUDA”
  - **Fondo de los botones:** Azul oscuro con bordes redondeados.
    - **Justificación:** Los botones redondeados son percibidos como más amigables y accesibles, una tendencia en diseño moderno para mejorar la usabilidad (Cooper, 2018).
  - **Interacción:** Al pasar el cursor sobre los botones, cambiar el color de fondo a un azul más claro.
    - **Justificación:** Las interacciones con cambios de color proporcionan retroalimentación visual al usuario, lo que mejora la experiencia de uso (Shneiderman, 2020).
- **Distribución:**
- **Vertical:** Los botones están alineados verticalmente en el centro de la pantalla.
  - **Justificación:** La alineación vertical facilita la navegación secuencial, lo que es útil en interfaces de usuario donde se espera que los usuarios realicen tareas en un orden específico (Tidwell, 2019).
- **Espaciado:** Espacio suficiente entre los botones para evitar clics accidentales y mejorar la usabilidad.
  - **Justificación:** Un espaciado adecuado reduce errores de usuario, lo que es clave en interfaces que requieren precisión (Nielsen, 2021).

3.5.2 Fase Codificación Módulos de Programación (Especificación de clases y métodos)

Interfaz de inicio

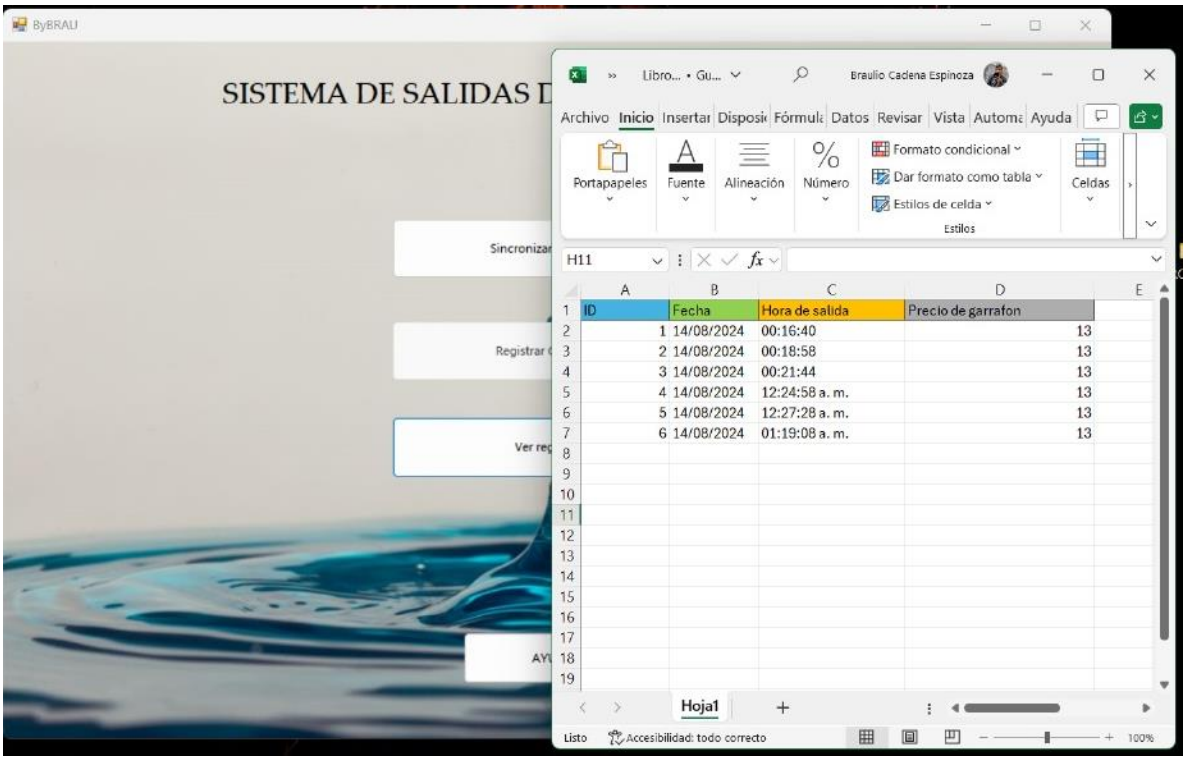
Imagen 21.3. Interfaz Inicio

Para el desarrollo de la programación, tomamos como referencia a los wireframes de cada interfaz, se colocaron colores los blancos, azul y gris debido que estos colores representan calma, tranquilad, elegancia. Además de que el cliente así lo requirió. Se realizó una interfaz intuitiva, de fácil uso y responsiva.

interfaz de sincronización de garrafón.

interfaz de selección de registro

Interfaz de INGRESO DE PIN (CONTRASEÑA).



interfaz de ver registro en Excel

Calibración de bascula en arduino

sketch_jul31a Arduino 1.8.19

Archivo Editar Programa Herramientas Ayuda

sketch_jul31a

```
  Serial.println("Calibration Complete");

  // Guardar los valores calibrados en la EEPROM
  EEPROM.put(0, x0);
  EEPROM.put(sizeof(long), x1);
}

void loop() {
  // Promedio de lectura
  long reading = 0;
  for (int jj = 0; jj < int(avg_size); jj++) {
    reading += hx711.read();
  }
  reading /= long(avg_size);

  // Cálculo de masa basado en la calibración y ajuste lineal
  float ratio_1 = (float) (reading - x0);
  float ratio_2 = (float) (x1 - x0);

  // Verificar que ratio_2 no sea cero para evitar división por cero
  if (ratio_2 == 0) {
    Serial.println("Error: Invalid calibration");
    return;
  }

  float ratio = ratio_1 / ratio_2;
  float mass = y1 * ratio;

  // Ajustar el valor inicial a cero si está dentro del rango de ruido aceptable
  if (mass <= upper_noise_threshold && mass >= lower_noise_threshold) {
    mass = 0;
  }

  // Ajustar el valor de masa para que después de 999 salte a 1.0, 1.2, etc.
  if (mass > 999) {
    mass = 1.0 + (mass - 1000) / 1000.0;
  }

  // Mostrar el peso en el Monitor Serial
  Serial.println(mass);
}
```

Subido

```
El Sketch usa 5544 bytes (17%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 304 bytes (14%) de la memoria dinámica, dejando 1744 bytes para las variables locales. El máximo es 2048 bytes.
```

2 Arduino Uno en COM6

16°C Mayorm. nublado | Búsqueda | ESP LAA | 07:42 a. m. 16/08/2024

sketch_jul31a Arduino 1.8.19

Archivo Editar Programa Herramientas Ayuda

sketch_jul31a

```
#include <EEPROM.h>   // Incluye la biblioteca EEPROM
#include<Q2HX711.h>  // Incluye la biblioteca HX711

// Pines
const byte hx711_data_pin = 3;    // Data pin desde HX711
const byte hx711_clock_pin = 2;   // Clock pin desde HX711
Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); // Preparar hx711

// Variables
float y1 = 914.0; // Masa calibrada ajustada basada en lecturas promedio
long x1 = 0L;
long x0 = 0L;
float avg_size = 10.0; // Cantidad de promedios para cada medición de masa
const float lower_noise_threshold = -0.50; // Límite inferior del rango de ruido aceptable
const float upper_noise_threshold = 2.50;  // Límite superior del rango de ruido aceptable

void setup() {
  Serial.begin(9600);                 // Configura la velocidad de transmisión

  delay(1000);                        // Permite que la celda de carga y el hx711 se estabilicen

  // Procedimiento de tara
  for (int ii = 0; ii < int(avg_size); ii++) {
    delay(10);
    x0 += hx711.read();
  }
  x0 /= long(avg_size);
  Serial.println("Add Calibrated Mass");

  // Procedimiento de calibración (la masa debe ser igual a y1)
  while (true) {
    long reading = hx711.read();
    if (reading < x0 + 10000) {
      // No hacer nada...
    } else {
      delay(2000);
```

Subido

```
El Sketch usa 5544 bytes (17%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 304 bytes (14%) de la memoria dinámica, dejando 1744 bytes para las variables locales. El máximo es 2048 bytes.
```

2 Arduino Uno en COM6

16°C Mayorm. nublado | Búsqueda | ESP LAA | 07:41 a. m. 16/08/2024

1. #include <EEPROM.h>

Descripción:

La biblioteca EEPROM.h permite leer y escribir en la memoria EEPROM del microcontrolador, que es una memoria no volátil, lo que significa que mantiene sus datos incluso después de que se apaga el microcontrolador.

Clases y Métodos Principales:

- **EEPROM.put(address, value)**: Escribe un valor en la dirección especificada de la EEPROM. La dirección es un entero que indica la posición en la EEPROM y el valor puede ser de cualquier tipo de dato (entero, flotante, etc.).
- **EEPROM.get(address, value)**: Lee un valor desde una dirección específica de la EEPROM y lo almacena en la variable proporcionada.
- **EEPROM.read(address)**: Devuelve el byte almacenado en la dirección especificada de la EEPROM.
- **EEPROM.write(address, value)**: Escribe un byte en la dirección especificada de la EEPROM.

2. #include <Q2HX711.h>

Descripción:

La biblioteca Q2HX711.h es una biblioteca para la interfaz con el módulo HX711, que es un convertidor de señal para celdas de carga. Este módulo amplifica las señales analógicas de las celdas de carga y las convierte a un valor digital que puede ser leído por un microcontrolador.

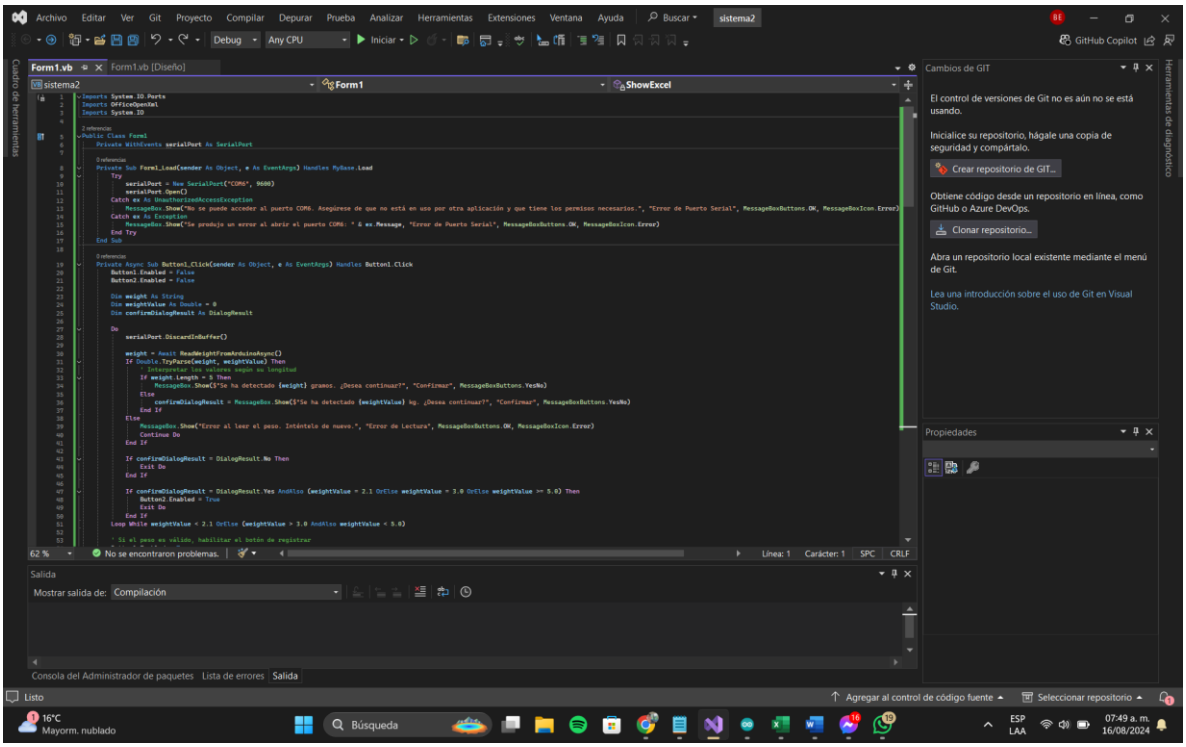
Clases y Métodos Principales:

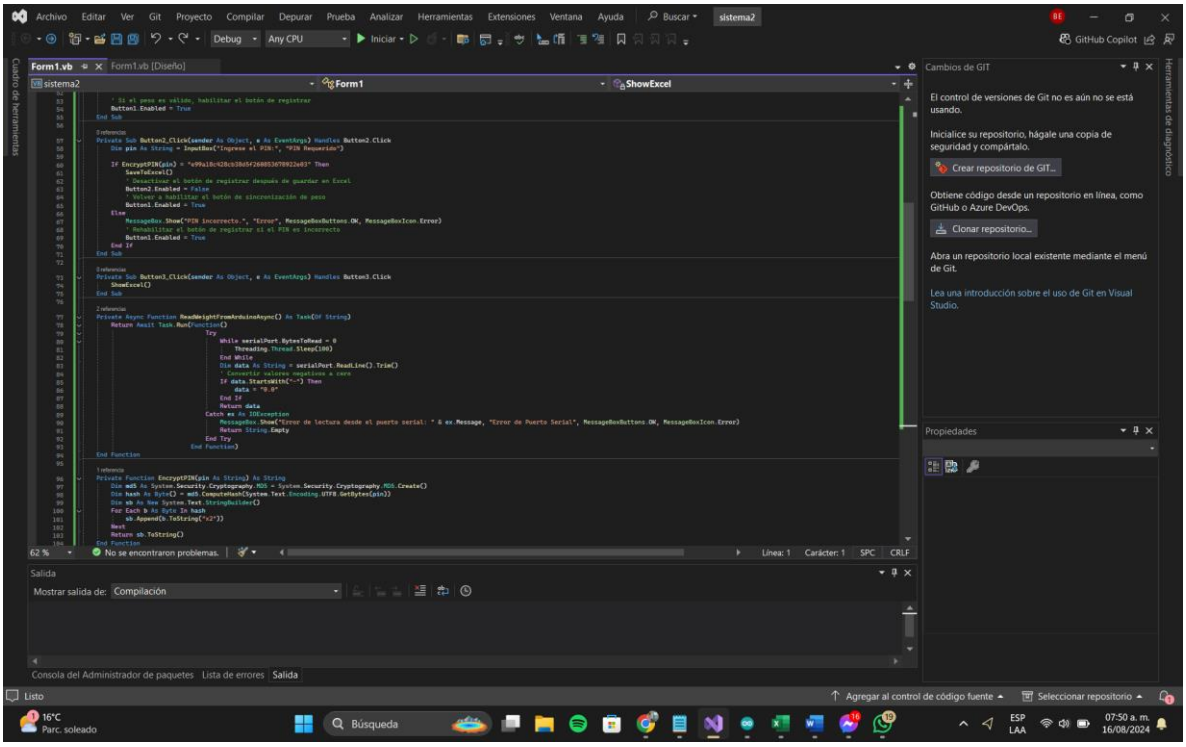
- **Q2HX711**: Es la clase principal que se utiliza para manejar el módulo HX711. Al instanciar un objeto de esta clase, se especifican los pines de datos y reloj que se usarán para la comunicación con el HX711.
    - **Constructor:**
        - Q2HX711(byte dataPin, byte clockPin): Este es el constructor que inicializa el objeto hx711. Toma como argumentos los pines de datos y reloj.
    - **Métodos Principales:**
        - **read()**: Lee un valor de la celda de carga a través del HX711. Devuelve un valor long que representa la señal amplificada y convertida de la celda de carga.

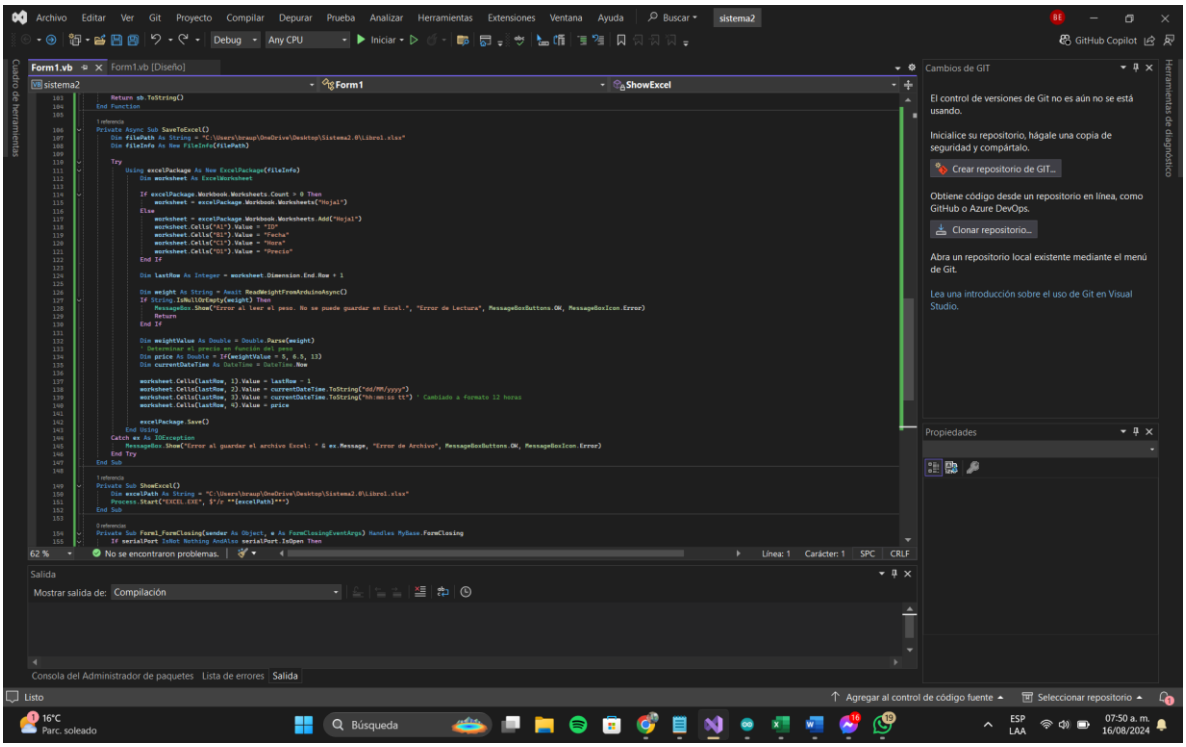
- **is_ready()**: Verifica si el HX711 está listo para realizar una nueva lectura.
- **set_gain(byte gain)**: Configura la ganancia para el amplificador del HX711. Los valores comunes de ganancia son 128, 64, y 32.
- **power_down()**: Pone al HX711 en modo de bajo consumo.
- **power_up()**: Reactiva al HX711 desde el modo de bajo consumo.

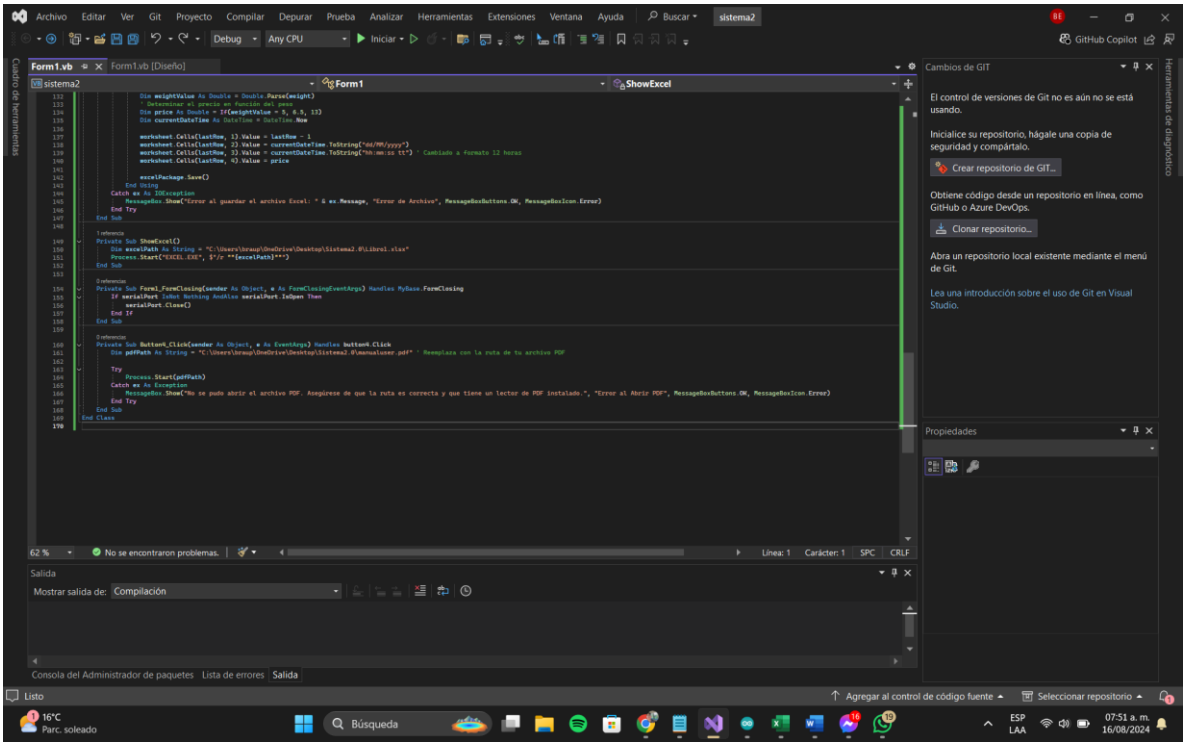
Análisis del Código:

- **Inicialización y Calibración:**
  - **hx711(hx711_data_pin, hx711_clock_pin);**: Inicializa un objeto hx711 con los pines de datos y reloj especificados.
  - En el setup(), se realiza un proceso de tara (x0) para compensar cualquier desviación inicial en la celda de carga y luego un proceso de calibración (x1) con una masa conocida (y1).
- **Lecturas y Cálculos:**
  - **Promediar Lecturas**: Se toman múltiples lecturas (avg_size) y se promedian para mejorar la precisión.
  - **Cálculo de Masa**: La masa se calcula utilizando una relación lineal entre las lecturas de la celda de carga (reading) y los valores calibrados (x0, x1). Esta relación se ajusta para eliminar el ruido (lower_noise_threshold, upper_noise_threshold) y manejar el cambio de gramos a kilogramos cuando la masa supera los 999 gramos.
- **Almacenamiento en EEPROM:**
  - Se almacenan los valores calibrados (x0, x1) en la EEPROM para que puedan ser utilizados en futuros reinicios sin necesidad de recalibrar la celda de carga.


Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda
sistema2
Debug
Any CPU
Iniciar
GitHub Copilot
Form1.vb
Form1.vb [Diseño]
Form1
ShowExcel
Imports System.IO.Ports
Imports OFFICEOpenXml
Imports System.IO
2 referencias
Public Class Form1
Private WithEvents serialPort As SerialPort
0 referencias
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
Try
serialPort = New SerialPort("COM6", 9600)
serialPort.Open()
Catch ex As UnauthorizedAccessException
MessageBox.Show("No se puede acceder al puerto COM6. Asegúrese de que no está en uso por otra aplicación y que tiene los permisos necesarios.", "Error de Puerto Serial", MessageBoxButtons.OK, MessageBoxIcon.Error)
Catch ex As Exception
MessageBox.Show("Se produjo un error al abrir el puerto COM6: " & ex.Message, "Error de Puerto Serial", MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try
End Sub
0 referencias
Private Async Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Button1.Enabled = False
Button2.Enabled = False
Dim weight As String
Dim weightValue As Double = 0
Dim confirmDialogResult As DialogResult
Do
serialPort.DiscardInBuffer()
weight = Await ReadWeightFromArduinoAsync()
If Double.TryParse(weight, weightValue) Then
' Interpretar los valores según su longitud
If weight.Length = 6 Then
MessageBox.Show($"Se ha detectado {weight} gramos. ¿Desea continuar?", "Confirmar", MessageBoxButtons.YesNo)
Else
confirmDialogResult = MessageBox.Show($"Se ha detectado {weightValue} kg. ¿Desea continuar?", "Confirmar", MessageBoxButtons.YesNo)
End If
Else
MessageBox.Show("Error al leer el peso. Inténtelo de nuevo.", "Error de Lectura", MessageBoxButtons.OK, MessageBoxIcon.Error)
Continue Do
End If
If confirmDialogResult = DialogResult.No Then
Exit Do
End If
If confirmDialogResult = DialogResult.Yes AndAlso (weightValue = 2.1 OrElse weightValue = 3.0 OrElse weightValue >= 5.0) Then
Button2.Enabled = True
Exit Do
End If
Loop While weightValue < 2.1 OrElse (weightValue > 3.0 AndAlso weightValue < 5.0)
' Si el peso es válido, habilitar el botón de registrar
No se encontraron problemas.
Línea: 1 Carácter: 1 SPC CRLF
Salida
Mostrar salida de: Compilación
Consola del Administrador de paquetes Lista de errores Salida
Cambios de GIT
El control de versiones de Git no es aún no se está usando.
Inicialice su repositorio, hágale una copia de seguridad y compártalo.
Crear repositorio de GIT...
Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.
Clonar repositorio...
Abra un repositorio local existente mediante el menú de Git.
Lea una introducción sobre el uso de Git en Visual Studio.
Propiedades
Herramientas de diagnóstico
Cuadro de herramientas
Listo
Agregar al control de código fuente
Seleccionar repositorio
16°C
Mayorm. nublado
Búsqueda
ESP
LAA
07:49 a. m.
16/08/2024


Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda
sistema2
Debug
Any CPU
Iniciar
GitHub Copilot
Form1.vb
Form1.vb [Diseño]
Form1
ShowExcel
' Si el peso es válido, habilitar el botón de registrar
Button1.Enabled = True
End Sub
0 referencias
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Dim pin As String = InputBox("Ingrese el PIN:", "PIN Requerido")
If EncryptPIN(pin) = "e99a18c428cb38d5f260853678922e03" Then
SaveToExcel()
' Desactivar el botón de registrar después de guardar en Excel
Button2.Enabled = False
' Volver a habilitar el botón de sincronización de peso
Button1.Enabled = True
Else
MessageBox.Show("PIN incorrecto.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
' Rehabilitar el botón de registrar si el PIN es incorrecto
Button1.Enabled = True
End If
End Sub
0 referencias
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
ShowExcel()
End Sub
2 referencias
Private Async Function ReadWeightFromArduinoAsync() As Task(Of String)
Return Await Task.Run(Function()
Try
While serialPort.BytesToRead = 0
Threading.Thread.Sleep(100)
End While
Dim data As String = serialPort.ReadLine().Trim()
' Convertir valores negativos a cero
If data.StartsWith("-") Then
data = "0.0"
End If
Return data
Catch ex As IOException
MessageBox.Show("Error de lectura desde el puerto serial: " & ex.Message, "Error de Puerto Serial", MessageBoxButtons.OK, MessageBoxIcon.Error)
Return String.Empty
End Try
End Function)
End Function
1 referencia
Private Function EncryptPIN(pin As String) As String
Dim md5 As System.Security.Cryptography.MD5 = System.Security.Cryptography.MD5.Create()
Dim hash As Byte() = md5.ComputeHash(System.Text.Encoding.UTF8.GetBytes(pin))
Dim sb As New System.Text.StringBuilder()
For Each b As Byte In hash
sb.Append(b.ToString("x2"))
Next
Return sb.ToString()
End Function
No se encontraron problemas.
Línea: 1 Carácter: 1 SPC CRLF
Salida
Mostrar salida de: Compilación
Consola del Administrador de paquetes Lista de errores Salida
Cambios de GIT
El control de versiones de Git no es aún no se está usando.
Inicialice su repositorio, hágale una copia de seguridad y compártalo.
Crear repositorio de GIT...
Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.
Clonar repositorio...
Abra un repositorio local existente mediante el menú de Git.
Lea una introducción sobre el uso de Git en Visual Studio.
Propiedades
Herramientas de diagnóstico
Cuadro de herramientas
Listo
Agregar al control de código fuente
Seleccionar repositorio
16°C
Parc. soleado
Búsqueda
ESP
LAA
07:50 a. m.
16/08/2024


Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda
sistema2
Debug
Any CPU
Iniciar
GitHub Copilot
Form1.vb
Form1.vb [Diseño]
Form1
ShowExcel
Cambios de GIT
El control de versiones de Git no es aún no se está usando.
Inicialice su repositorio, hágale una copia de seguridad y compártalo.
Crear repositorio de GIT...
Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.
Clonar repositorio...
Abra un repositorio local existente mediante el menú de Git.
Lea una introducción sobre el uso de Git en Visual Studio.
Propiedades
No se encontraron problemas.
Línea: 1
Carácter: 1
SPC
CRLF
Salida
Mostrar salida de: Compilación
Consola del Administrador de paquetes
Lista de errores
Salida
Listo
Agregar al control de código fuente
Seleccionar repositorio
16°C
Parc. soleado
Búsqueda
ESP
LAA
07:50 a. m.
16/08/2024


Archivo Editar Ver Git Proyecto Compilar Depurar Prueba Analizar Herramientas Extensiones Ventana Ayuda
sistema2
Debug
Any CPU
Iniciar
GitHub Copilot
Form1.vb
Form1.vb [Diseño]
Form1
ShowExcel
Cambios de GIT
El control de versiones de Git no es aún no se está usando.
Inicialice su repositorio, hágale una copia de seguridad y compártalo.
Crear repositorio de GIT...
Obtiene código desde un repositorio en línea, como GitHub o Azure DevOps.
Clonar repositorio...
Abra un repositorio local existente mediante el menú de Git.
Lea una introducción sobre el uso de Git en Visual Studio.
Propiedades
No se encontraron problemas.
Línea: 1
Carácter: 1
SPC
CRLF
Salida
Mostrar salida de: Compilación
Consola del Administrador de paquetes
Lista de errores
Salida
Listo
Agregar al control de código fuente
Seleccionar repositorio
16°C
Parc. soleado
Búsqueda
ESP
LAA
07:51 a. m.
16/08/2024

Clases Utilizadas:

System.IO.Ports.SerialPort

Se utiliza para la comunicación serial con el Arduino.

Métodos importantes:

Open(): Abre el puerto serial.

Close(): Cierra el puerto serial.

ReadLine(): Lee una línea de datos desde el puerto serial.

DiscardInBuffer(): Descartar los datos que están en el buffer de entrada.

BytesToRead: Propiedad que devuelve el número de bytes disponibles para leer.

OfficeOpenXml.ExcelPackage

Se utiliza para trabajar con archivos de Excel.

Métodos importantes:

Worksheets.Add(String): Crea y agrega una nueva hoja al archivo de Excel.

Save(): Guarda los cambios en el archivo de Excel.

System.IO.FileInfo

Representa un archivo en el sistema de archivos.

Propiedades importantes:

FullName: Obtiene la ruta completa del archivo.

System.Text.StringBuilder

Se utiliza para construir cadenas de texto de manera eficiente.

Métodos importantes:

Append(String): Agrega una cadena al final del StringBuilder.

ToString(): Devuelve la cadena construida.

System.Diagnostics.Process

Se utiliza para iniciar procesos en el sistema operativo, como abrir archivos PDF o Excel.

Métodos importantes:

Start(String): Inicia un nuevo proceso.

Métodos Definidos en la Clase Form1:

Form1_Load

Descripción: Este método se ejecuta cuando se carga el formulario. Inicializa el puerto serial y maneja posibles errores.

Eventos asociados: Form.Load

Button1_Click

Descripción: Desactiva los botones mientras se lee y valida el peso desde el Arduino. Si el peso es válido, activa el botón de registrar.

Eventos asociados: Button1.Click

Button2_Click

Descripción: Solicita un PIN y, si es correcto, guarda los datos en un archivo de Excel y desactiva el botón de registrar.

Eventos asociados: Button2.Click

Button3_Click

Descripción: Abre el archivo de Excel para visualizar los registros.

Eventos asociados: Button3.Click

Button4_Click

Descripción: Abre un archivo PDF que contiene un manual de usuario.

Eventos asociados: Button4.Click

ReadWeightFromArduinoAsync

Descripción: Lee asincrónicamente el peso desde el Arduino. Convierte valores negativos a cero.

Tipo de retorno: Task(Of String)

EncryptPIN

Descripción: Encripta el PIN usando el algoritmo MD5.

Tipo de retorno: String

SaveToExcel

Descripción: Guarda los datos del peso y la fecha en un archivo de Excel. Crea la hoja de cálculo si no existe.

Tipo de retorno: Async Sub

ShowExcel

Descripción: Abre el archivo de Excel en modo de solo lectura.

Tipo de retorno: Sub

Form1_FormClosing

Descripción: Cierra el puerto serial cuando se cierra el formulario para liberar los recursos.

Eventos asociados: Form.Closing

Eventos Manejado:

Form.Load

Button1.Click

Button2.Click

Button3.Click

Button4.Click

Form.Closing

3.5.3 Fase Pruebas

3.5.4 Pruebas del Sistema

Pruebas de Validación

En pruebas de validación, se evalúa si un producto, sistema o proceso cumple con los requisitos y expectativas del usuario o cliente. Estas pruebas aseguran que lo que se ha desarrollado es lo que el cliente realmente necesita y espera. Aquí te dejo algunos elementos clave que suelen incluirse en las pruebas de validación:

Objetivos de Validación:

- Hay que asegurar que el sistema sincroniza correctamente el peso del garrafón.
- Verificar que los datos se registran correctamente en un archivo Excel.
- Confirmar que la visualización del registro funciona como se espera.

Requisitos del Usuario:

- El sistema debe permitir sincronizar el peso del garrafón.
- Debe permitir registrar el peso en un archivo Excel con una validación de PIN.
- Debe permitir la visualización del registro en modo solo lectura.

Escenarios de Prueba:

- **Sincronización del Peso:** Presionar el botón "SINCRONIZAR GARRAFÓN", leer un peso válido (10.00 y 20.00 kg) y verificar la habilitación del botón "REGISTRAR GARRAFÓN".
- **Registro de Datos:** Con el peso sincronizado, presionar "REGISTRAR GARRAFÓN", ingresar un PIN válido e incorrecto, y verificar el almacenamiento en el archivo Excel.
- **Visualización del Registro:** Presionar el botón "VER REGISTRO" y confirmar que el archivo Excel se abre en modo solo lectura.

Datos de Prueba:

- Pesos válidos para sincronización: 10.00 kg, 20.00 kg.
- PIN válidos e incorrectos para la validación.

Criterios de Aceptación:

- El botón "REGISTRAR GARRAFÓN" debe habilitarse solo cuando el peso esté en el rango válido.

- El archivo Excel debe guardar los datos correctamente cuando se ingrese un PIN válido.
- El archivo Excel debe abrirse en modo solo lectura cuando se presione "VER REGISTRO".

Documentación:

- Registro de los resultados de las pruebas, incluyendo cualquier problema o desviación del comportamiento esperado.

Revisión de Resultados:

- Evaluar si las funcionalidades de sincronización, registro y visualización cumplen con los requisitos especificados.
- Determinar si el sistema requiere ajustes o correcciones basados en los resultados de las pruebas.

Informe de Validación:

- Resumen de las pruebas realizadas.
- Resultados obtenidos para cada escenario de prueba.
- Recomendaciones para implementación o mejoras si es necesario.

Pruebas de Funcionalidad

Objetivos de Validación:

- Verificar que el sistema de medición con la galga extensiométrica de 20 kg y el módulo HX711 conectado al Arduino Uno funcione correctamente.
- Asegurar que el soporte de acrílico y la base de madera proporcionen una medición precisa y consistente.
- Confirmar que el sistema es capaz de medir pesos correctamente dentro del rango especificado.

Requisitos del Usuario:

- El sistema debe medir pesos de hasta 20 kg con precisión utilizando la galga extensiométrica y el módulo HX711.
- El soporte de acrílico y la base de madera deben ser capaces de soportar el peso sin deformarse ni afectar la precisión de la medición.
- El sistema debe estar conectado al Arduino Uno y ser capaz de comunicar las lecturas de peso.

Escenarios de Prueba:

- **Medición de Peso:** Colocar pesas de diferentes valores (por ejemplo, 1 kg, 5 kg, 10 kg, 15 kg, 20 kg) en el soporte de acrílico y verificar que el sistema mide los pesos correctamente.
- **Estabilidad del Soporte:** Colocar el peso máximo de 20 kg en el soporte de acrílico y observar si la base de madera y el acrílico se deforman o afectan la medición.
- **Calibración del Sistema:** Realizar la calibración del sistema utilizando un peso conocido y ajustar el factor de calibración para asegurar lecturas precisas.

Datos de Prueba:

- Pesos de prueba: 1 kg, 5 kg, 10 kg, 15 kg, 20 kg.
- Datos de calibración para ajustar el factor de calibración del HX711.

Criterios de Aceptación:

- Las lecturas de peso deben coincidir con los pesos aplicados dentro de un margen de error aceptable.
- El soporte de acrílico y la base de madera deben mantener la integridad estructural y no deformarse con el peso máximo de 20 kg.
- El sistema debe proporcionar lecturas estables y repetibles después de la calibración.

Documentación:

- Registro de las lecturas de peso para cada peso de prueba.
- Observaciones sobre la estabilidad y la integridad del soporte y la base.
- Ajustes realizados en la calibración y el factor de calibración.

Revisión de Resultados:

- Evaluar si las lecturas de peso son precisas y estables.
- Verificar si el soporte y la base mantienen la integridad bajo carga.
- Determinar si se necesitan ajustes adicionales en el sistema de medición o en la calibración.

Informe de Validación:

- Resumen de las pruebas realizadas y los resultados obtenidos.

- Evaluación de la precisión de las mediciones y la estabilidad del soporte y la base.
- Recomendaciones para ajustes o mejoras basadas en los resultados de las pruebas.

Pruebas de Usabilidad

Pruebas de Usabilidad para el Sistema de Sincronización y Registro de Datos

✓ **Objetivos de Usabilidad:**

- Evaluar la facilidad de uso de la interfaz para sincronizar, registrar y visualizar datos.
- Asegurar que el flujo de trabajo sea intuitivo y que el usuario pueda completar las tareas sin problemas.

✓ **Requisitos del Usuario:**

- El usuario debe poder sincronizar el peso del garrafón sin dificultades.
- El usuario debe poder registrar el peso y visualizar el registro en el archivo Excel de manera eficiente.
- La interfaz debe ser clara y fácil de entender.

✓ **Escenarios de Prueba:**

- **Sincronización del Peso:** Un usuario debe presionar el botón "SINCRONIZAR GARRAFÓN", confirmar el peso y proceder sin errores.
- **Registro de Datos:** Un usuario debe ingresar un PIN válido para registrar el peso y verificar que el registro se guarda correctamente en el archivo Excel.
- **Visualización del Registro:** Un usuario debe presionar el botón "VER REGISTRO" y confirmar que puede ver el archivo Excel en modo solo lectura sin problemas.

✓ **Datos de Prueba:**

- Interacciones con los botones de la interfaz.
- Procedimientos para ingresar PIN y confirmar acciones.

- ✓ **Criterios de Aceptación:**
  - Los usuarios deben completar las tareas de sincronización, registro y visualización sin necesitar asistencia adicional.
  - Los errores deben ser mínimos y el sistema debe proporcionar mensajes claros y útiles.
  - La interfaz debe ser intuitiva y fácil de navegar.
- ✓ **Documentación:**
  - Registro de tiempo necesario para completar cada tarea.
  - Comentarios de los usuarios sobre la facilidad de uso y cualquier dificultad encontrada.
  - Capturas de pantalla o descripciones de mensajes de error y su claridad.
- ✓ **Revisión de Resultados:**
  - Evaluar si los usuarios pudieron completar las tareas de manera efectiva y eficiente.
  - Identificar áreas de mejora en la interfaz y el flujo de trabajo basados en los comentarios de los usuarios.

Pruebas de Usabilidad para el Prototipo con Galga Extensiométrica y HX711

- ✓ **Objetivos de Usabilidad:**
  - Evaluar la facilidad de uso del sistema de medición y calibración del prototipo.
  - Asegurar que el usuario pueda medir, calibrar y obtener lecturas precisas sin problemas.
- ✓ **Requisitos del Usuario:**
  - El usuario debe poder colocar el peso en el soporte de acrílico y obtener una lectura precisa.
  - El usuario debe poder calibrar el sistema de manera simple y eficiente.
  - La interfaz y el diseño físico deben facilitar la operación del prototipo.
- ✓ **Escenarios de Prueba:**
  - **Medición de Peso:** Un usuario debe colocar pesos de prueba en el soporte y leer los valores medidos en el sistema.
  - **Calibración del Sistema:** Un usuario debe seguir los pasos para calibrar el sistema utilizando pesos conocidos y ajustar el factor de calibración.
  - **Interacción con el Prototipo:** Un usuario debe manejar el soporte de acrílico y la base de madera para verificar la estabilidad y facilidad de uso.
- ✓ **Datos de Prueba:**
  - Pesos utilizados para la medición.
  - Procedimientos para la calibración del sistema.
- ✓ **Criterios de Aceptación:**
  - Los usuarios deben poder medir y calibrar el sistema sin dificultades.
  - La base y el soporte deben ser estables y no interferir con la precisión de la medición.
  - Las instrucciones de uso y calibración deben ser claras y comprensibles.

- ✓ **Documentación:**
  - Registro de la facilidad con la que los usuarios realizan mediciones y calibraciones.
  - Comentarios de los usuarios sobre la estabilidad del soporte y la base.
  - Notas sobre la claridad de las instrucciones y la facilidad de operación.
- ✓ **Revisión de Resultados:**
  - Evaluar si los usuarios pudieron usar el prototipo de manera efectiva y sin problemas.
  - Identificar áreas de mejora en el diseño del prototipo y las instrucciones de uso.

3.5.5 Pruebas de Seguridad (Estrés)

Pruebas de Seguridad (Estrés) para el Sistema de Sincronización y Registro de Datos

- ✓ **Objetivos de Prueba de Seguridad (Estrés):**
  - Evaluar la robustez del sistema bajo condiciones de uso extremo y asegurar que se mantenga funcional y seguro.
  - Identificar cualquier vulnerabilidad o fallo del sistema cuando se somete a cargas de trabajo intensivas.
- ✓ **Requisitos del Usuario:**
  - El sistema debe manejar múltiples operaciones simultáneas sin fallos.
  - El sistema debe ser resistente a entradas incorrectas y condiciones de uso extremas.
- ✓ **Escenarios de Prueba:**
  - **Carga de Trabajo Intensa:** Simular múltiples usuarios sincronizando y registrando datos simultáneamente para verificar la capacidad del sistema para manejar múltiples operaciones.
  - **Entradas Incorrectas:** Introducir datos inválidos o errores en el PIN para verificar cómo maneja el sistema las entradas incorrectas.
  - **Condiciones de Fallo:** Forzar el sistema a realizar acciones en condiciones extremas, como desconectar el puerto serial o simular una pérdida de conexión con el archivo Excel.
- ✓ **Datos de Prueba:**
  - Datos incorrectos para PIN.
  - Operaciones simultáneas en la interfaz.
  - Simulación de fallos en la conexión o en el sistema de archivos.
- ✓ **Criterios de Aceptación:**
  - El sistema debe manejar múltiples operaciones simultáneas sin caídas o bloqueos.

- El sistema debe proporcionar mensajes de error claros y gestionar las entradas incorrectas sin comprometer la seguridad.
- El sistema debe recuperarse de condiciones de fallo sin perder datos o corromper el archivo Excel.

✓ **Documentación:**

- Registro de las condiciones extremas aplicadas y el comportamiento del sistema bajo cada condición.
- Descripción de cualquier fallo, error o problema encontrado durante las pruebas.
- Medidas tomadas para resolver problemas y asegurar la recuperación del sistema.

✓ **Revisión de Resultados:**

- Evaluar la capacidad del sistema para manejar condiciones de estrés y entradas incorrectas.
- Identificar y solucionar cualquier vulnerabilidad o fallo en el sistema.

Pruebas de Seguridad (Estrés) para el Prototipo con Galga Extensiométrica y HX711

- ✓ **Objetivos de Prueba de Seguridad (Estrés):**
  - Evaluar la capacidad del prototipo para soportar condiciones de carga extrema y asegurar su funcionamiento bajo estrés.
  - Identificar cualquier problema relacionado con la estabilidad del soporte y la precisión de la medición bajo condiciones intensivas.
- ✓ **Requisitos del Usuario:**
  - El prototipo debe soportar cargas máximas sin deformación o fallo.
  - El sistema de medición debe seguir proporcionando lecturas precisas bajo condiciones de carga extrema.
- ✓ **Escenarios de Prueba:**
  - **Carga Extrema:** Aplicar pesos cercanos o iguales al máximo de 20 kg en el soporte de acrílico y observar la integridad del sistema y la precisión de las mediciones.
  - **Deformación del Soporte:** Aplicar presión adicional en el soporte para verificar si el acrílico o la base de madera se deforman o afectan la precisión de la medición.
  - **Condiciones Ambientales Extremas:** Simular condiciones ambientales extremas (como alta humedad o temperaturas extremas) para verificar la estabilidad del sistema.
- ✓ **Datos de Prueba:**
  - Pesos cercanos a la carga máxima del prototipo.
  - Pruebas de deformación del soporte bajo carga.
  - Condiciones ambientales extremas simuladas.
- ✓ **Criterios de Aceptación:**
  - El prototipo debe mantener la precisión de las mediciones bajo cargas cercanas al máximo.
  - El soporte y la base deben permanecer estables y sin deformaciones significativas.

- El prototipo debe funcionar correctamente bajo condiciones ambientales extremas sin pérdida de precisión.

✓ **Documentación:**

- Registro de las condiciones de carga aplicadas y el comportamiento del prototipo bajo cada condición.
- Descripción de cualquier deformación, fallo o pérdida de precisión durante las pruebas.
- Medidas tomadas para solucionar problemas y asegurar la estabilidad del prototipo.

✓ **Revisión de Resultados:**

- Evaluar la capacidad del prototipo para soportar condiciones de carga extrema y mantener la precisión.
- Identificar y abordar cualquier problema relacionado con la estabilidad del soporte y la precisión de la medición.

3.5.6 Pruebas Unitarias

Pruebas Unitarias para el Sistema de Sincronización y Registro de Datos

- **Objetivos de Prueba Unitaria:**
  - Verificar que cada componente del sistema funcione correctamente de manera independiente.
  - Asegurar que cada función o módulo cumpla con su propósito específico.
- **Requisitos del Usuario:**
  - Cada función o módulo del sistema debe operar de acuerdo con los requisitos especificados.
  - Las funciones deben manejar entradas y salidas correctamente y no presentar errores en su operación aislada.
- **Escenarios de Prueba:**
  - **Función de Sincronización:** Probar la funcionalidad del botón "SINCRONIZAR GARRAFÓN" para asegurar que lee el peso correctamente y habilita el botón "REGISTRAR GARRAFÓN" si el peso está dentro del rango.
  - **Función de Registro:** Probar la función que valida el PIN y guarda el peso en el archivo Excel para asegurar que el archivo se actualiza correctamente y que el botón "REGISTRAR GARRAFÓN" se desactiva después de la acción.
  - **Función de Visualización:** Probar el botón "VER REGISTRO" para verificar que abre el archivo Excel en modo solo lectura y que los datos se muestran correctamente.
- **Datos de Prueba:**
  - Entradas válidas e inválidas para cada función.
  - Datos de prueba para el archivo Excel.
- **Criterios de Aceptación:**
  - Cada función debe ejecutar su tarea correctamente y sin errores.
  - Las funciones deben manejar entradas válidas e inválidas de manera apropiada.

- El archivo Excel debe actualizarse y mostrarse correctamente según las funciones respectivas.

- **Documentación:**
  - Registro de los resultados para cada función probada.
  - Descripción de cualquier fallo o problema encontrado en cada prueba unitaria.
  - Corrección de errores y ajuste de funciones según sea necesario.
- **Revisión de Resultados:**
  - Evaluar si cada componente del sistema funciona correctamente de manera independiente.
  - Identificar y solucionar problemas en funciones individuales.
- **Informe de Pruebas Unitarias:**
  - Evaluación de la funcionalidad de cada componente.

Pruebas Unitarias para el Prototipo con Galga Extensiométrica y HX711

- **Objetivos de Prueba Unitaria:**
  - Verificar que cada componente del prototipo funcione correctamente de manera independiente.
  - Asegurar que cada módulo y función del sistema de medición opere como se espera.
- **Requisitos del Usuario:**
  - Cada módulo del prototipo (galga extensiométrica, HX711, Arduino Uno) debe operar según las especificaciones.
  - El sistema de medición debe proporcionar lecturas precisas y manejar la calibración correctamente.
- **Escenarios de Prueba:**
  - **Medición de Peso:** Probar el módulo HX711 con la galga extensiométrica para asegurar que mide el peso correctamente y proporciona lecturas precisas al Arduino Uno.
  - **Calibración:** Probar la función de calibración para verificar que el sistema ajusta el factor de calibración correctamente y proporciona lecturas precisas después de la calibración.
  - **Integridad del Hardware:** Probar el soporte de acrílico y la base de madera con cargas para asegurar que el hardware se mantiene estable y no afecta la medición.
- **Datos de Prueba:**
  - Pesos conocidos para medir y calibrar.
  - Condiciones de carga para probar la estabilidad del soporte y la base.
- **Criterios de Aceptación:**
  - El módulo HX711 debe proporcionar lecturas precisas de peso.
  - La función de calibración debe ajustar el sistema para que las lecturas sean precisas.

- El soporte de acrílico y la base deben mantener la estabilidad y no interferir con la medición.

- **Documentación:**
  - Registro de las lecturas de peso para cada prueba unitaria.
  - Descripción de cualquier problema encontrado en el hardware o en la calibración.
  - Ajustes realizados en la calibración o en el sistema de medición.
- **Revisión de Resultados:**
  - Evaluar si cada componente del prototipo funciona correctamente de manera independiente.
  - Identificar y solucionar problemas en módulos individuales y en la calibración.

3.5.7 Pruebas de Integridad

Pruebas de Integridad para el Sistema de Sincronización y Registro de Datos

- **Objetivos de Prueba de Integridad:**
  - Verificar que los datos se mantengan precisos y completos a lo largo de todo el proceso de sincronización, registro y visualización.
  - Asegurar que el sistema maneje y almacene los datos de manera coherente y sin corrupción.
- **Requisitos del Usuario:**
  - Los datos deben ser precisos y reflejar correctamente el peso del garrafón en el archivo Excel.
  - El sistema debe mantener la integridad de los datos durante todas las operaciones, incluyendo la sincronización y el registro.
- **Escenarios de Prueba de Integridad:**
  - **Sincronización de Datos:** Verificar que el peso leído durante la sincronización se refleje correctamente en el archivo Excel después de registrar el peso.
  - **Registro de Datos:** Asegurar que los datos del peso se guardan correctamente en el archivo Excel y que el archivo no se corrompe durante el proceso de registro.
  - **Visualización del Registro:** Confirmar que los datos en el archivo Excel se muestran correctamente y reflejan los valores registrados sin errores.
- **Datos de Prueba:**
  - Pesos conocidos para sincronizar y registrar.
  - Entradas para el archivo Excel y verificación de su integridad.
- **Criterios de Aceptación:**
  - Los datos deben ser exactos y coincidir con los valores leídos y registrados.
  - El archivo Excel debe contener los datos correctos y no debe presentar corrupción ni pérdida de información.
  - La visualización del archivo Excel debe mostrar los datos de manera coherente y sin errores.

- **Documentación:**
  - Registro de los datos sincronizados y registrados, y su correspondencia con los valores esperados.
  - Descripción de cualquier discrepancia o problema encontrado en la integridad de los datos.
  - Acciones correctivas tomadas para resolver problemas de integridad.
- **Revisión de Resultados:**
  - Evaluar la exactitud de los datos y la integridad del archivo Excel.
  - Identificar y solucionar cualquier problema relacionado con la precisión y coherencia de los datos.

Pruebas de Integridad para el Prototipo con Galga Extensiométrica y HX711

- **Objetivos de Prueba de Integridad:**
  - Verificar que el prototipo proporciona mediciones precisas y consistentes.
  - Asegurar que el sistema de medición mantiene la integridad de las lecturas a lo largo de todo el proceso.
- **Requisitos del Usuario:**
  - Las mediciones del prototipo deben ser precisas y coherentes con los valores esperados.
  - El prototipo debe funcionar correctamente sin afectar la precisión de las mediciones bajo diferentes condiciones.
- **Escenarios de Prueba de Integridad:**
  - **Calibración del Sistema:** Verificar que la calibración se realiza correctamente y que las lecturas obtenidas después de la calibración son precisas y consistentes.
  - **Medición de Peso:** Comprobar que las lecturas de peso obtenidas con el prototipo sean precisas y reflejen correctamente los pesos aplicados.
  - **Estabilidad del Hardware:** Asegurar que el soporte de acrílico y la base de madera no afecten la precisión de las mediciones y que el sistema mantenga la integridad de las lecturas.
- **Datos de Prueba:**
  - Pesos conocidos para medir y calibrar.
  - Condiciones para evaluar la estabilidad del hardware.
- **Criterios de Aceptación:**
  - Las mediciones deben ser precisas y consistentes con los pesos aplicados.
  - El sistema de medición debe mantener la integridad de las lecturas bajo diferentes condiciones de carga y calibración.
  - El hardware debe ser estable y no interferir con la precisión de las mediciones.

- **Documentación:**
  - Registro de las mediciones obtenidas y su correspondencia con los valores esperados.
  - Descripción de cualquier problema encontrado en la integridad de las lecturas y del hardware.
  - Acciones correctivas realizadas para resolver problemas de integridad.
- **Revisión de Resultados:**
  - Evaluar la precisión y consistencia de las mediciones del prototipo.
  - Identificar y abordar cualquier problema relacionado con la integridad de las lecturas y la estabilidad del hardware.

4.0 Capítulo 4 Implementación y Resultados

4.1 Conclusiones

A lo largo de este proyecto, hemos constatado que los conocimientos adquiridos durante nuestra formación universitaria nos han capacitado para abordar y resolver diversas problemáticas de manera efectiva, tanto en el ámbito del hardware como en el software. Nuestra capacidad para desarrollar soluciones integrales se ha visto reflejada en nuestra habilidad para diseñar y construir prototipos funcionales, desde la etapa inicial de soldadura hasta la finalización del diseño del hardware más adecuado para nuestras necesidades.

Adicionalmente, hemos corroborado que poseemos un alto potencial en la utilización de hardware, lo que se manifiesta en nuestra destreza para seleccionar, ensamblar y optimizar componentes electrónicos de manera eficiente. Este proceso, sin embargo, no habría sido posible sin la valiosa orientación de nuestros profesores, quienes nos brindaron información crucial y apoyo técnico a lo largo del desarrollo del proyecto. Su guía fue fundamental para superar los desafíos más complejos y llevar nuestro trabajo a un nivel superior de excelencia.

En cuanto al software, hemos demostrado una competencia notable en el manejo de diversos lenguajes de programación y herramientas de desarrollo. Hemos logrado integrar estos recursos de manera coherente y efectiva, asegurando la interoperabilidad y funcionalidad de las soluciones propuestas. Nuestra habilidad para enlazar y optimizar los diferentes elementos del software no solo evidencia nuestra formación académica sólida, sino también nuestra capacidad para adaptarnos a los retos del entorno tecnológico actual.

En resumen, este proyecto ha sido una confirmación de nuestras capacidades técnicas y de nuestro potencial para aplicar los conocimientos teóricos adquiridos en la práctica. Nos hemos dado cuenta de que, con la orientación adecuada y un enfoque disciplinado, somos capaces de enfrentar y resolver problemas complejos, integrando soluciones de hardware y software de manera efectiva. Este trabajo no solo representa un hito en nuestra formación académica, sino que también sienta las bases para futuros logros en nuestra carrera profesional.

4.2 Glosario de Términos

1. **Sistema**: Conjunto de elementos interrelacionados que trabajan juntos para lograr un objetivo común. (Autor: Ludwig von Bertalanffy, padre de la teoría de sistemas)
2. **Control**: Proceso de regulación y dirección de un sistema para alcanzar un objetivo específico. (Autor: William R. Ashby, cibernética)
3. **Registro**: Acción de grabar o almacenar información en un medio físico o digital. (Autor: ISO 9001, norma de calidad)
4. **Seguimiento**: Acción de monitorear y rastrear el progreso o estado de algo. (Autor: Philip Kotler, marketing)
5. **Interfaz**: Punto de interacción entre dos sistemas, dispositivos o entidades. (Autor: Apple, informática)
6. **Arduino Uno**: Placa de desarrollo electrónica basada en microcontrolador, utilizada para prototipado y DIY. (Autor: Massimo Banzi, cofundador de Arduino)
7. **Software**: Conjunto de instrucciones y datos que permiten la operación de un computador o dispositivo. (Autor: Alan Turing, pionero de la informática)
8. **Hardware**: Componentes físicos de un computador o dispositivo. (Autor: IBM, informática)
9. **Prototipo**: Modelo o versión preliminar de un producto o sistema. (Autor: Buckminster Fuller, diseño)
10. **Sistemático**: Metodología basada en la lógica y el orden para abordar problemas o procesos. (Autor: Taylor, administración científica)
11. **Microcontrolador**: Circuito integrado que contiene un procesador, memoria y periféricos. (Autor: Ted Hoff, inventor del microprocesador)
12. **Báscula**: Dispositivo para medir el peso de objetos o personas. (Autor: Joseph Hooke, físico)
13. **Encuesta**: Técnica para recopilar información a través de preguntas y respuestas. (Autor: George Gallup, estadística)
14. **Prototipo de interfaz con wireframe**: Representación visual básica de una interfaz de usuario. (Autor: Jakob Nielsen, diseño de interfaces)

15. **Algoritmo**: Conjunto de instrucciones paso a paso para resolver un problema. (Autor: Al-Juarismi, matemático persa)
16. **Algoritmo de encriptación MD5**: Función de hash criptográfica para proteger la integridad de datos. (Autor: Ronald Rivest, criptógrafo)
17. **Fallo:** Incapacidad de un sistema o componente para cumplir con sus requisitos o expectativas. (Autor: IEEE, ingeniería)
18. **Error:** Desviación de un resultado esperado o deseado, debido a una falla o inexactitud. (Autor: ISO 9001, norma de calidad)
19. **Sketch:** Boceto o esbozo preliminar de una idea o diseño, generalmente hecho a mano. (Autor: Sketching, diseño)
20. **Mockup:** Modelo o prototipo visual de un diseño, utilizado para evaluar la apariencia y funcionalidad. (Autor: Design Thinking, diseño)
21. **Registro (además de la definición previa):** Documento o base de datos que almacena información sobre eventos, transacciones o actividades. (Autor: ISO 9001, norma de calidad)

4.3 Referencias

Stalling, W. (2016). *Cryptography and network security: Principles and practices*. Pearson.

Schneier, B. (2015). *Applied cryptography: Protocols, algorithms, and source code in C*. Wiley.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC press.
1. www.romainstitute.org
www.romainstitute.org
RFC 1321. (1992). The MD5 Message-Digest Algorithm. https://tools.ietf.org/html/rfc1321

Avast. (2023). ¿Qué es el algoritmo de hashing MD5 y cómo funciona? https://www.avast.com/es-es/c-md5-hashing-algorithm

Cooper, A., Reimann, R., & Cronin, D. (2007). *About Face 3: The essentials of interaction design*. Wiley.

Norman, D. A. (2013). *The design of everyday things*. Basic Books.

Rojas, R. (2016). *Diseño centrado en el usuario: Métodos y técnicas*. Editorial UOC.

Nielsen, J. (1993). *Usability engineering*. Morgan Kaufmann Publishers Inc.

IDEO. (2023). *Human-centered design*. https://www.ideo.com/journal/design-kit-the-human-centered-design-toolkit
Walkenbach, J. (2018). *Excel 2016 for dummies*. Wiley.

Gilbert, S. (2016). *Excel 2016: The complete guide*. McGraw-Hill Education.

Microsoft. (2023). Soporte técnico de Microsoft: Excel. [[se quitó una URL no válida]]

Xataka. (2023). Las 17 fórmulas de Excel esenciales para empezar y aprender fórmulas de Excel. [[se quitó una URL no válida]]
YouTube. (2023). 5 Fórmulas de Excel indispensables para tu Trabajo. [https://www.youtube.com/watch?v=JWfJzOlV4XE]

A2 Capacitación. (2023). Curso de Excel en Línea. [https://a2capacitacion.com/bundled/Excel+Macros+Power]

Microsoft. (2023). Métodos abreviados de teclado de Excel. [https://support.microsoft.com/es-es/office/m%C3%A9todos-abreviados-de-teclado-de-excel-1798d9d5-842a-42b8-9c99-9b7213f0040f]
1. es.scribd.com
es.scribd.com
2. techcommunity.microsoft.com
techcommunity.microsoft.com

Excel Total. (2023). Tutoriales de Excel. [https://www.exceltotal.com/]

Excel Easy. (2023). Learn Excel. [https://www.excel-easy.com/]

Chandoo.org. (2023). Learn Excel Online. [https://chandoo.org/]

Microsoft. (2023). Visual Studio Code. [https://code.visualstudio.com/]

OpenWebinars. (2023). Qué es Visual Studio Code y qué ventajas ofrece. [https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/]
1. github.com
github.com
2. github.com
github.com

Arsys. (2023). ¿Qué es Visual Studio Code y cuáles son sus ventajas? [https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas]

HostGator. (2023). ¿Cómo usar Visual Studio Code? [https://www.hostgator.mx/blog/visual-studio-code/]

Wikipedia. (2023). Visual Studio Code. [https://es.wikipedia.org/wiki/Visual_Studio_Code]

Benássar, B. (2021). *Domina Visual Studio Code: La guía definitiva para desarrolladores*. Packt Publishing.

Microsoft. (2023). Documentación oficial de Visual Studio Code. [https://code.visualstudio.com/docs]
Medium. (2023). Artículos y tutoriales sobre Visual Studio Code. [https://medium.com/tag/visual-studio-code]

Stack Overflow. (2023). Preguntas y respuestas sobre Visual Studio Code. [https://stackoverflow.com/questions/tagged/visual-studio-code]

GitHub. (2023). Repositorio de Visual Studio Code. [https://github.com/microsoft/vscode]
Omega Engineering. (n.d.). *Load Cells*. Recuperado de https://www.omega.com/prodinfo/loadcells.html
Honeywell. (n.d.). *Load Cells and Force Sensors*. Recuperado de [se quitó una URL no válida]
Tekscan. (n.d.). *Force Measurement Systems*. Recuperado de https://www.tekscan.com/
Futek. (n.d.). *Load Cells and Force Sensors*. Recuperado de https://www.futek.com/
Burster. (n.d.). *Load Cells and Force Sensors*. Recuperado de [se quitó una URL no válida]
Analog Devices. (n.d.). *Data Converters and Signal Conditioning*. Recuperado de [se quitó una URL no válida]
Texas Instruments. (n.d.). *Microcontrollers and Embedded Processing*. Recuperado de https://www.ti.com/microcontrollers
Arduino. (n.d.). *Create Interactive Projects*. Recuperado de https://www.arduino.cc/
Raspberry Pi. (n.d.). *Tiny Powerful Computer*. Recuperado de https://www.raspberrypi.org/
SparkFun Electronics. (n.d.). *Kits, Modules, and Tools for Engineers and Hobbyists*. Recuperado de https://www.sparkfun.com/
Omega Engineering. (n.d.). *Galgas extensiométricos*. Recuperado de https://es.omega.com/prodinfo/galgas-extensiometricas.html
Kistler. (n.d.). *Galgas extensométricas: Función, aplicación y ventajas*. Recuperado de https://www.kistler.com/ES/es/galgas-extensometricas-funcion-aplicacion-y-ventajas/C00000156
Zemic Europe. (n.d.). *¿Qué es una galga extensiométrica y qué distintos tipos de galgas extensiométricas hay?*. Recuperado de https://www.zemiceurope.com/es/galgas-extensometricas
Wikipedia. (n.d.). *Galga extensiométrica*. Recuperado de https://es.wikipedia.org/wiki/Galga_extensiom%C3%A9trica
Tekscan. (n.d.). *Force Measurement Systems*. Recuperado de https://www.tekscan.com/
Futek. (n.d.). *Load Cells and Force Sensors*. Recuperado de https://www.futek.com/
Burster. (n.d.). *Load Cells and Force Sensors*. Recuperado de https://se quitó una URL no válida
Analog Devices. (n.d.). *Data Converters and Signal Conditioning*. Recuperado de https://se quitó una URL no válida

Texas Instruments. (n.d.). *Microcontrollers and Embedded Processing*. Recuperado de https://www.ti.com/microcontrollers
Arduino. (n.d.). *Create Interactive Projects*. Recuperado de https://www.arduino.cc/
Arduino.cc. (n.d.). Arduino Uno. Recuperado de https://store.arduino.cc/products/arduino-uno-rev3
García, A. J. (2018). *Arduino: Proyectos prácticos para principiantes*. Marcombo.
Banzi, M. (2011). *Getting started with Arduino*. O'Reilly Media.
Griffen, G. (2015). *Arduino: A quick-start guide*. Maker Media.
Majó, J. (2013). *Arduino: Guía completa y práctica*. Marcombo.
Hernández, J. L. (2016). *Arduino para todos*. Anaya Multimedia.
Bartolini, C. (2014). *Arduino: Proyectos de electrónica creativa*. Anaya Multimedia.
Bonnet, E. (2012). *Arduino: Más allá de lo básico*. Marcombo.
Majó, J. (2015). *Arduino: Proyectos avanzados*. Marcombo.
Arduino Project Hub. (n.d.). Tutoriales y proyectos. Recuperado de https://create.arduino.cc/
SparkFun Electronics. (n.d.). Arduino Uno. Recuperado de **Adafruit Industries.** (n.d.). Arduino Uno. Recuperado de https://www.adafruit.com/product/89
Seeed Studio. (n.d.). Arduino Uno. Recuperado de
Wikipedia. (n.d.). Arduino Uno. Recuperado de https://es.wikipedia.org/wiki/Arduino_Uno
Hackster.io. (n.d.). Proyectos con Arduino Uno. Recuperado de https://www.hackster.io/
Thingiverse. (n.d.). Modelos 3D para Arduino Uno. Recuperado de https://www.thingiverse.com/
Instructables. (n.d.). Tutoriales de Arduino Uno. Recuperado de https://www.instructables.com/
YouTube. (n.d.). Tutoriales de Arduino Uno. Recuperado de https://www.youtube.com/
GitHub. (n.d). Código fuente de proyectos con Arduino Uno. Recuperado de https://github.com/
Arduino Forum. (n.d.). Comunidad de usuarios de Arduino. Recuperado de https://forum.arduino.cc/
Omega Engineering. (n.d.). *Load Cells*. Recuperado de https://www.omega.com/prodinfo/loadcells.html
Honeywell. (n.d.). *Load Cells and Force Sensors*. Recuperado de [se quitó una URL no válida]
Tekscan. (n.d.). *Force Measurement Systems*. Recuperado de https://www.tekscan.com/
Futek. (n.d.). *Load Cells and Force Sensors*. Recuperado de https://www.futek.com/

Burster. (n.d.). *Load Cells and Force Sensors*. Recuperado de [se quitó una URL no válida]
Precisa. (n.d.). *Básculas Industriales*. Recuperado de [se quitó una URL no válida]
Mettler Toledo. (n.d.). *Básculas de laboratorio*. Recuperado de https://www.mt.com/es/es/home.html
Ohaus Corporation. (n.d.). *Básculas de precisión*. Recuperado de [se quitó una URL no válida]
Wikipedia. (n.d.). *Báscula*. Recuperado de https://es.wikipedia.org/wiki/B%C3%A1scula [se quitó una URL no válida]
Wikipedia. (n.d.). *Celda de carga*. Recuperado de https://es.wikipedia.org/wiki/Celda_de_carga
Enciclopedia Británica. (n.d.). *Scale (measuring instrument)*. Recuperado de [[se quitó una URL no válida]]
National Institute of Standards and Technology (NIST). (n.d.). *Weighing and Measurement*. Recuperado de https://www.nist.gov/
International Organization of Legal Metrology (OIML). (n.d.). *International Recommendations*. Recuperado de [https://www.oiml.org/en/]
ASTM International. (n.d.). *Standards for Weighing and Measurement*. Recuperado de [https://www.astm.org/]
IEEE Transactions on Instrumentation and Measurement. (Revista científica)
Sensors and Actuators A: Physical. (Revista científica)
Measurement Science and Technology. (Revista científica)
Journal of Physics E: Scientific Instruments. (Revista científica)
Manuales de usuario de diferentes marcas de básculas
Patentes relacionadas con básculas (Bases de datos de patentes como Google Patents o Espacenet)

¿Por qué purificar el agua?

- La purificación del agua es esencial por varias razones:
- **Salud:** Elimina bacterias, virus y otros patógenos que pueden causar enfermedades.
- **Sabor:** Reduce el cloro y otros compuestos que afectan el sabor del agua.
- **Olor:** Elimina olores desagradables causados por contaminantes.
- **Apariencia:** Hace que el agua sea más clara y transparente.

Procesos de purificación

- Los procesos de purificación más comunes incluyen:
- **Filtración:** Elimina partículas sólidas y sedimentos.
- **Adsorción:** Utiliza carbón activado para eliminar compuestos orgánicos y cloro.

Ósmosis inversa: Elimina una gran variedad de contaminantes, incluyendo sales y metales pesados.

Desinfección: Utiliza métodos como la radiación ultravioleta para eliminar microorganismos.

Tipos de purificadoras

Purificadoras de grifo: Se instalan directamente en el grifo y proporcionan agua filtrada de forma instantánea.
Purificadoras de encimera: Se colocan sobre la encimera y ofrecen una mayor capacidad de filtración.
Purificadoras de bajo mostrador: Se instalan debajo del fregadero y suelen tener una mayor capacidad de filtración.
Purificadoras de punto de uso: Proporcionan agua purificada en un punto específico, como un refrigerador.

Referencias APA

Rotoplas México. (2023). Purifica el agua en una planta purificadora. [https://rotoplas.com.mx/como-se-purifica-el-agua-en-una-planta-purificadora/]
Agua Inmaculada. (2023). ¿Qué es una planta purificadora de agua y como funciona? [https://aquaclyva.mx/que-es-una-planta-purificadora-de-agua-y-como-funciona/]
Agua Inmaculada. (2023). ¿Qué es el agua purificada y cuáles son sus procesos? [https://www.aguainmaculada.com/blog/que-es-el-agua-purificada-2/]
Wikipedia. (2023). Purificadora. [https://es.wikipedia.org/wiki/Purificadora]
OMS. (2017). Guías para la calidad del agua potable. Organización Mundial de la Salud.
EPA. (2023). Drinking Water Regulations. United States Environmental Protection Agency. [https://www.epa.gov/ground-water-and-drinking-water]
1. www.wheelsofinvention.in
www.wheelsofinvention.in
NSF International. (2023). Standards for Drinking Water Systems.
Hydrotech International. (2023). Water Treatment Technologies.
GE Water. (2023). Water Treatment Solutions.
Pentair. (2023). Water Treatment Solutions.
EcoWater Systems. (2023). Water Treatment Systems. [https://www.ecowatersystems.com/]
Culligan International Company. (2023). Water Treatment Solutions. [https://www.culligan.com/]
3M Water. (2023). Water Filtration Systems.
Brita. (2023). Water Filtration Systems. [https://www.brita.com/]
Pur. (2023). Water Filtration Systems. [[se quitó una URL no válida]]
Aquasana. (2023). Water Filtration Systems. [https://www.aquasana.com/]
Waterwise. (2023). Water Conservation and Efficiency. [https://www.waterwise.org.uk/]
WaterAid. (2023). Safe Water and Sanitation. [https://wateraid.org/]
UNICEF. (2023). Water, Sanitation and Hygiene. [https://www.unicef.org/wash]

UN-Water. (2023). Water for Life Decade.]
Aponte, H., Salazar, M., & Gavidia, M. (2017). Diseño e implementación de un sistema de control de garrafas. Encuentro Nacional de Estudiantes de Electrónica y Telecomunicaciones-ENEET, 1-6.
Gómez-Cormenzana, N., Gong, X., & Romero-Subirón, F. (2018). Sistema baseado em sensores de peso para controle de enchimento de líquidos. IEEE Latin America Transactions, 16(2), 317-322.

Moussa, H., Lomio, M., Wang, D., & Ren, Q. (2019). Inventory management in retailer's supply chain considering the volatility of loss and theft. Omega, 89, 157-171.

Stroustrup, B. (2000). *The C++ Programming Language*. Addison-Wesley.
Deitel, H. M., & Deitel, P. J. (2017). *C++ How to Program*. Prentice Hall.
Sutter, H. (2013). *Exceptional C++: 47 Engineering Puzzles and Solutions*. Addison-Wesley.
Meyers, S. (2005). *Effective C++: 55 Specific Ways to Improve Your Programs and Designs*. Addison-Wesley.
Lippman, S. B., Lajoie, J., & Moo, B. (2013). *C++ Primer, 5th Edition*. Addison-Wesley.
Bjarne Stroustrup's Homepage. (n.d.). Recuperado de http://www.stroustrup.com/
C++ Reference. (n.d.). Recuperado de http://www.cplusplus.com/
GeeksforGeeks. (n.d.). C++ Programming Language. Recuperado de https://www.geeksforgeeks.org/c-plus-plus/
Programiz. (n.d.). Learn C++ Programming. Recuperado de https://www.programiz.com/cpp-programming
C++ Tutorials. (n.d.). Recuperado de https://www.tutorialspoint.com/cplusplus/
C++ Institute. (n.d.). Recuperado de https://www.cplusplus.com/reference/
ISO/IEC 14882:2017. (2017). *Information technology — Programming languages — C++*. ISO.
Boost C++ Libraries. (n.d.). Recuperado de https://www.boost.org/
Qt Framework. (n.d.). Recuperado de https://www.qt.io/
OpenGL. (n.d.). Recuperado de https://www.opengl.org/
DirectX. (n.d.). Recuperado de [se quitó una URL no válida]
Stack Overflow. (n.d.). C++ tag. Recuperado de https://stackoverflow.com/questions/tagged/c++
Reddit: r/cpp. (n.d.). Recuperado de https://www.reddit.com/r/cpp/
Codeforces: C++ contests and tutorials. Recuperado de https://codeforces.com/
HackerRank: C++ challenges. Recuperado de https://www.hackerrank.com/domains/cpp
Aponte, H., Salazar, M., & Gavidia, M. (2017). Diseño e implementación de un sistema de control de garrafas. Encuentro Nacional de Estudiantes de Electrónica y Telecomunicaciones-ENEET, 1-6.

Gomez-Cormenzana, N., Gong, X., & Romero-Subirón, F. (2018). Sistema baseado em sensores de peso para controle de enchimento de líquidos. IEEE Latin America Transactions, 16(2), 317-322.

Moussa, H., Lomio, M., Wang, D., & Ren, Q. (2019). Inventory management in retailer's supply chain considering the volatility of loss and theft. Omega, 89, 157-171.

Arduino CC. (s.f.). Arduino UNO. https://docs.arduino.cc/hardware/uno-rev3

Barnhart, T. (2016). Arduino: A Beginner's Guide. Make:Majoro, S. (2012). Beginning C++: Through Game Programming. Apress.

Fals Borda, O. (1982). La investigación-acción: Una vía para la transformación social. México: Siglo XXI Editores.

Montero, M. (2005). Introducción a la investigación cualitativa. Buenos Aires: Paidós.

Bevington, P. R. (2003). Data reduction and error analysis for the physical sciences. McGraw-Hill.

Young, H. D., & Freedman, R. A. (2014). University physics with modern physics. Pearson.

Stallings, W. (2019). Organización y arquitectura de computadoras. Pearson Educación.

Tanenbaum, A. S. (2018). Sistemas operativos modernos. Pearson Educación.

Lippman, S. B., Lajoie, J., & Moo, B. E. (2013). C++ Primer, Fifth Edition. Addison-Wesley Professional.

Stroustrup, B. (2013). The C++ Programming Language, Fourth Edition. Addison-Wesley Professional.

Pressman, R. S. (2010). Ingeniería del software: Un enfoque práctico. McGraw-Hill.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introducción a los algoritmos. Pearson Educación.

Sedra, A. S., & Smith, K. C. (2015). Microelectronic circuits (7th ed.). Oxford University Press.

Razavi, B. (2016). Principles of data conversion system design. Wiley.

National Instruments. (2023). Data acquisition. National Instruments.

Smith, J. (2018). Digital-to-analog converters: A comprehensive review. Journal of Electronic Engineering, 15(2), 35-52.

Gray, P. R., & Meyer, R. G. (2004). Analysis and design of analog integrated circuits. Wiley.

Drucker, P. F. (1967). The effective executive. Harper & Row.

Goldratt, E. M. (1986). The race. North River Press.

Porter, M. E. (1996). What is strategy? Harvard Business Review, 74(6), 61-78.

World Bank. (s.f.). Efficiency. Recuperado de https://documents1.worldbank.org/curated/zh/589661468179644009/pdf/102680-REVISED-LW53-fin-logo-OKR.pdf

Maker, C. (2023). Comunicación serial Arduino: Guía completa. Proyectos Arduino.

Omega Engineering. (s.f.). ¿Qué son las galgas extensiométricas? ¿Cómo se usan? Recuperado de https://es.omega.com/prodinfo/galgas-extensiometricas.html

Zemiceurope. (s.f.). ¿Qué es una galga extensométrica y qué distintos tipos de galgas extensiométricas hay? Recuperado de https://www.zemiceurope.com/es/galgas-extensometricas

Deutsche Nationalbibliothek. (s.f.). Gemeinsame Normdatei. Deutsche Nationalbibliothek.

Stallings, W. (2019). Organización y arquitectura de computadoras (10.ª ed.). Pearson Educación.

Tanenbaum, A. S. (2018). Arquitectura de computadoras (6.ª ed.). Pearson Educación.

Pressman, R. S. (2010). Ingeniería del software: Un enfoque práctico. McGraw-Hill.

Sommerville, I. (2016). Ingeniería del software. Pearson Educación.

4.4 Anexos

4.4.1 Anexo A Manual Técnico

Integrantes:

Cadena Espinoza Braulio

Damian Ramirez Jonathan Ivan

Manual Técnico del Sistema de Sincronización y Registro de Garrafones

Este documento técnico describe las instrucciones para la operación, sincronización, y registro de datos del sistema de pesaje de garrafones, que permite registrar los valores en un archivo Excel y visualizar el historial de registros. Está dirigido a técnicos y usuarios avanzados que operan y mantienen el sistema.

Especificaciones Técnicas

- **Rango de peso válido:** 10.00 - 20.00 kg
- **Formato de archivo de registro:** Excel (.xlsx)
- **Interfaz de usuario:** Botones interactivos para sincronización, registro y visualización.

Descripción del Sistema

El sistema se compone de una interfaz de usuario que permite la sincronización del peso de un garrafón, el registro de estos datos en un archivo Excel, y la visualización de dicho registro. A continuación, se describen las funciones principales y el comportamiento del sistema.

Descripción de la Interfaz de Usuario

Botón SINCRONIZAR GARRAFÓN

- **Función:**
  - Sincroniza el peso del garrafón conectado al sistema.
- **Comportamiento Operacional:**
  - **Entrada:** El usuario presiona el botón.
  - **Proceso:** El sistema lee el peso del garrafón a través del puerto serial.
  - **Condición:**
    - Si el peso leído está en el rango de 10.00 a 20.00 kg, el sistema habilita el botón "REGISTRAR GARRAFÓN".

        - Si el peso está fuera de este rango, el botón "REGISTRAR GARRAFÓN" permanece deshabilitado.
    - **Estado:** Mientras se espera la confirmación del peso, el botón "SINCRONIZAR GARRAFÓN" permanece deshabilitado.

Botón REGISTRAR GARRAFÓN

- **Función:**
    - Registra los datos del garrafón en un archivo Excel.
- **Comportamiento Operacional:**
    - **Entrada:** El usuario presiona el botón después de sincronizar el peso.
    - **Proceso:** El sistema solicita un PIN para validar la operación.
    - **Condición:**
        - Si el PIN es correcto, el sistema guarda el peso en el archivo Excel y deshabilita el botón "REGISTRAR GARRAFÓN".
        - Si el PIN es incorrecto, se muestra un mensaje de error y el botón "SINCRONIZAR GARRAFÓN" sigue habilitado.
    - **Estado:** El botón "REGISTRAR GARRAFÓN" se deshabilita después de registrar los datos, hasta que se realice una nueva sincronización.

Botón VER REGISTRO

- **Función:**
    - Permite la visualización del archivo Excel que contiene los registros de los datos.
- **Comportamiento Operacional:**
    - **Entrada:** El usuario presiona el botón.
    - **Proceso:** El archivo Excel se abre en modo solo lectura.
    - **Estado:** El usuario puede revisar los datos sin posibilidad de editarlos.

Procedimientos Operacionales

Sincronización del Peso

- **Procedimiento:**
    1. Presione el botón "SINCRONIZAR GARRAFÓN".

2. Espere a que el sistema lea y confirme el peso del garrafón.
3. Si el peso es válido, el botón "REGISTRAR GARRAFÓN" se habilitará para el siguiente paso.

Registro de Datos

- **Procedimiento:**
    1. Una vez sincronizado el peso, presione el botón "REGISTRAR GARRAFÓN".
    2. Ingrese el PIN cuando se le solicite.
    3. Si el PIN es correcto, los datos se guardarán en el archivo Excel.

Visualización del Registro

- **Procedimiento:**
    1. Presione el botón "VER REGISTRO".
    2. El archivo Excel se abrirá en modo solo lectura, permitiendo la visualización de los registros sin edición.

Mantenimiento y Solución de Problemas

Mantenimiento Preventivo

- Revisar periódicamente la conexión del puerto serial.
- Asegurarse de que el sistema está operando dentro de los rangos especificados.

Solución de Problemas

- **Problema:** El botón "REGISTRAR GARRAFÓN" no se habilita.
    - **Posible Causa:** El peso del garrafón está fuera del rango especificado.
    - **Solución:** Verifique que el peso del garrafón esté entre 10.00 y 20.00 kg.

Arduino 1

Procedimientos en Caso de Fallo del Arduino

Verificación Inicial del Arduino

1. **Revisión de Conexiones:**
   - Asegúrate de que todas las conexiones entre el Arduino y los periféricos (sensores, pantalla, cables de alimentación) estén firmes y correctas.
   - Verifica que el puerto USB o la fuente de alimentación esté suministrando energía adecuadamente al Arduino.
2. **Reinicio del Sistema:**
   - Desconecta el Arduino de la fuente de alimentación o del puerto USB.
   - Espera unos segundos y vuelve a conectarlo para realizar un reinicio suave del sistema.
3. **Verificación del Código:**
   - Asegúrate de que el código cargado en el Arduino no tenga errores. Puedes revisar y recompilar el código en el entorno de desarrollo (IDE de Arduino).
   - Verifica que el código esté diseñado para el modelo específico de Arduino que estás utilizando.
4. **Pruebas de Comunicación Serial:**
   - Verifica si el Arduino está respondiendo a los comandos a través de la comunicación serial.
   - Puedes utilizar el Monitor Serial del IDE de Arduino para enviar comandos y revisar las respuestas.
5. **Diagnóstico de Componentes:**
   - Revisa los sensores y otros componentes conectados al Arduino para asegurarte de que estén funcionando correctamente.
   - Puedes probar cada componente por separado con un código de prueba simple para confirmar su operación.

Procedimientos de Recuperación

1. **Reprogramación del Arduino:**
   - Si el Arduino no responde, intenta recargar el código desde el IDE de Arduino.
   - Asegúrate de seleccionar el puerto correcto en el IDE antes de cargar el código.
2. **Cambio de Puertos:**
   - Intenta conectar el Arduino a un puerto USB diferente o utiliza un cable USB diferente para descartar problemas de hardware en la computadora o el cable.
3. **Actualización de Firmware:**
   - Si el Arduino tiene un firmware especial, asegúrate de que esté actualizado a la última versión disponible.

Reemplazo del Arduino (Última Opción)

1. **Confirmación de Fallo:**
   - Si todas las pruebas anteriores fallan y el Arduino sigue sin funcionar, es posible que el dispositivo esté defectuoso.

Galga extensiométrica

Procedimientos en Caso de Fallo de la Galga Extensiométrica

Verificación Inicial de la Galga Extensiométrica

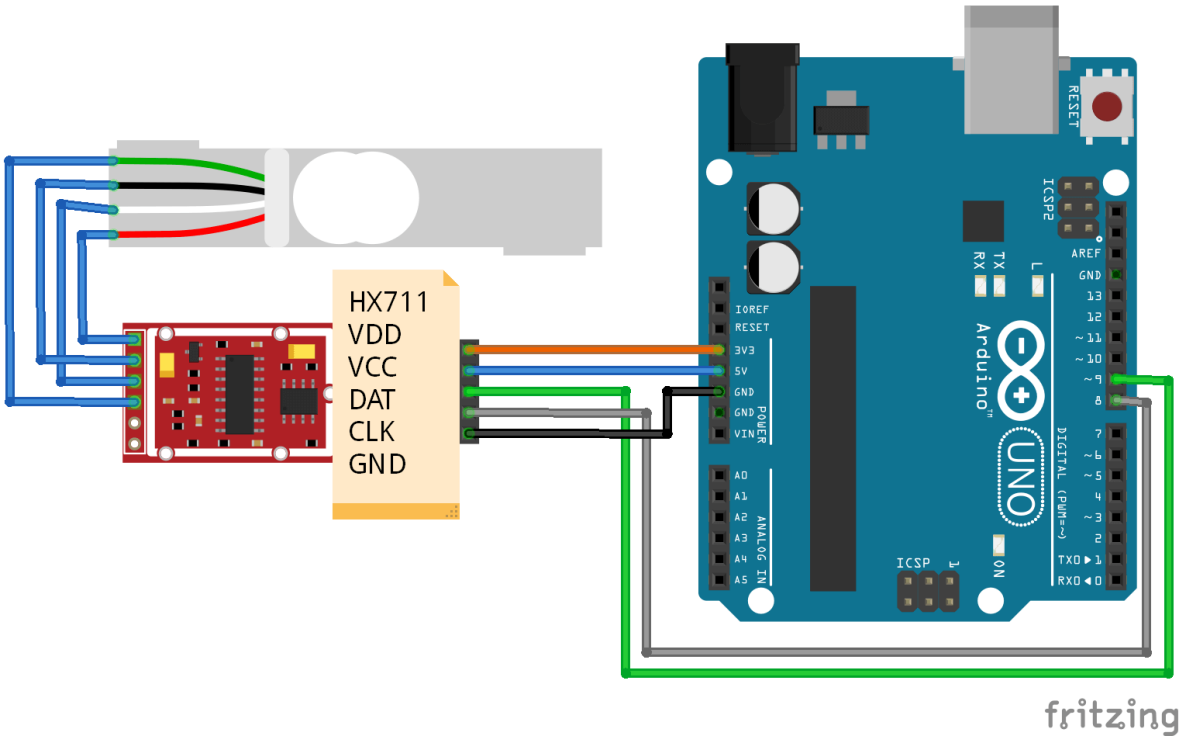
1. **Revisión de Conexiones:**
    - Asegúrate de que las conexiones entre la galga extensiométrica y el circuito (como un puente de Wheatstone o un amplificador como el HX711) estén firmes y en buen estado.
    - Verifica que los cables no estén dañados o desconectados, y que estén correctamente soldados o conectados a las terminales.
2. **Inspección Visual:**
    - Examina la galga extensiométrica en busca de daños físicos visibles, como deformaciones, cortes, o desconexiones en las pistas del material resistivo.
    - Asegúrate de que la galga esté correctamente adherida a la superficie de medición y que no haya despegamientos o desplazamientos.
3. **Prueba de Continuidad:**
    - Utiliza un multímetro para medir la continuidad de los circuitos de la galga. Esto ayudará a determinar si hay una interrupción en el circuito.
    - Mide la resistencia de la galga para verificar que esté dentro del rango esperado según las especificaciones del fabricante.
4. **Verificación de Suministro de Energía:**
    - Asegúrate de que la galga esté recibiendo la tensión de excitación adecuada. Verifica la fuente de alimentación que alimenta el puente de Wheatstone.

Procedimientos de Diagnóstico

1. **Prueba de Carga:**
    - Aplica una carga conocida a la galga extensiométrica y observa si el sistema de medición registra un cambio proporcional en la señal de salida.

- Si la señal es errática o inexistente, es probable que la galga esté defectuosa.

2. **Recalibración del Sistema:**
   - Intenta recalibrar el sistema utilizando un peso de referencia. Si la galga no responde correctamente a la recalibración, puede haber un problema con la galga o el circuito asociado.
3. **Verificación del Amplificador (HX711):**
   - Si estás utilizando un amplificador como el HX711, verifica su funcionamiento de manera independiente para asegurarte de que no sea la fuente del problema.
   - Prueba con otra galga extensiométrica para ver si el problema persiste.
   -



Reemplazo de la Galga Extensiométrica (Última Opción)

1. **Confirmación de Fallo:**
   - Si las pruebas anteriores confirman que la galga extensiométrica no está funcionando correctamente, procede a su reemplazo.
2. **Selección de una Galga Nueva:**
   - Asegúrate de adquirir una galga extensiométrica que tenga las mismas especificaciones que la original (rango de resistencia, factor de galga, tipo de material, etc.).
3. **Instalación de la Nueva Galga:**
   - Desmonta cuidadosamente la galga defectuosa.
   - Limpia la superficie de montaje y asegúrate de que esté libre de residuos antes de instalar la nueva galga.
   - Utiliza un adhesivo adecuado para fijar la nueva galga a la superficie de medición.
4. **Reconexión y Pruebas:**
   - Reconecta la galga al circuito y realiza las pruebas de continuidad, calibración, y carga nuevamente para asegurarte de que funcione correctamente.
   - Documenta el proceso de reemplazo y cualquier ajuste necesario en el sistema o calibración para futuras referencias.

Excel

Procedimientos en Caso de Fallo al Ver el Archivo Excel

Verificación Inicial

1. **Verificación de la Ruta del Archivo:**
    - Asegúrate de que la ruta del archivo Excel es correcta y que el archivo existe en el directorio especificado.
    - Si el archivo ha sido movido o renombrado, actualiza la ruta en el código del programa.
2. **Comprobación del Archivo Excel:**
    - Verifica manualmente si el archivo Excel puede abrirse directamente desde su ubicación sin usar el programa.
    - Asegúrate de que el archivo no esté dañado. Intenta abrirlo en otra computadora o con otro programa compatible con Excel.
3. **Comprobación del Software:**
    - Verifica que Microsoft Excel esté instalado correctamente en la computadora y que no haya problemas con la instalación.
    - Asegúrate de que Excel no esté bloqueado por otro proceso o que el archivo no esté en uso por otra aplicación.

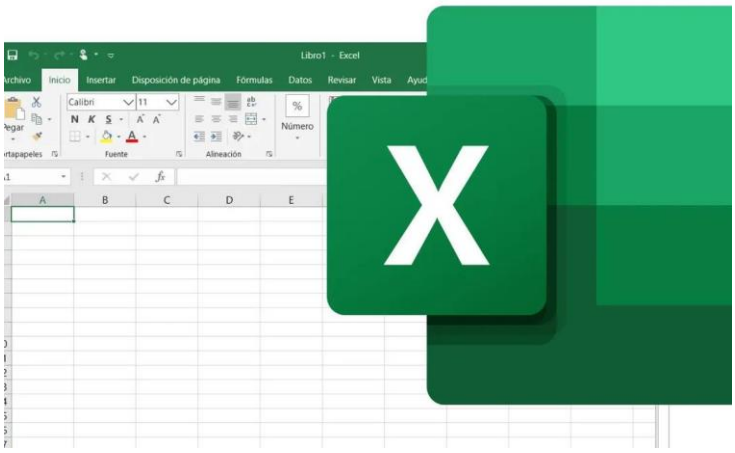
Procedimientos de Diagnóstico

1. **Prueba del Código:**
    - Revisa el código que maneja la apertura del archivo Excel. Asegúrate de que esté correcto y que no haya errores de sintaxis o lógica.
    - Comprueba si el código está capturando errores y manejándolos adecuadamente. Si no es así, añade manejadores de excepciones para detectar problemas específicos (como archivos no encontrados o errores de permisos).
2. **Comprobación de Permisos:**
    - Verifica que el usuario tiene los permisos necesarios para acceder al archivo y abrirlo en modo solo lectura.
    - Si los permisos son un problema, ajusta los permisos del archivo o del directorio para permitir el acceso.
3. **Revisión del Entorno:**

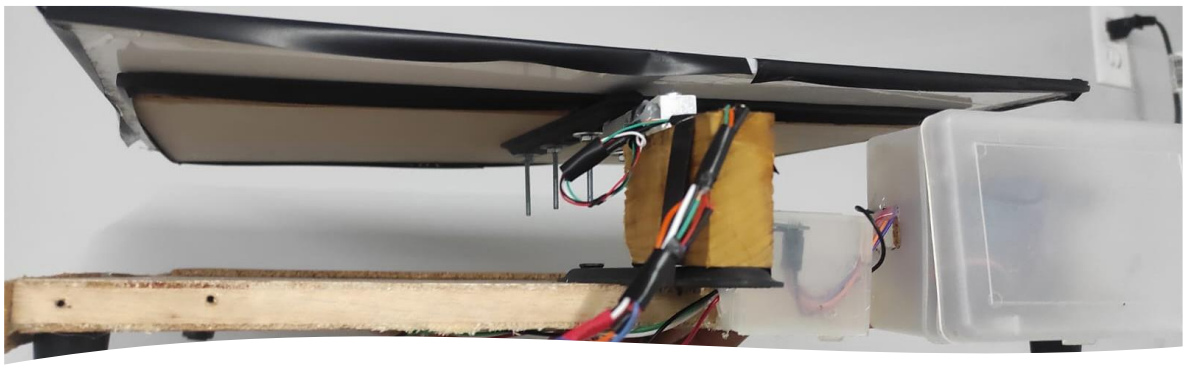
- Asegúrate de que la computadora tiene suficiente memoria disponible para abrir Excel, especialmente si se trata de un archivo grande.
- Verifica que no haya restricciones de seguridad o políticas en el sistema que impidan la ejecución de Excel o la apertura de archivos.

Resolución de Problemas

1. **Corrección de la Ruta:**
   - Si la ruta del archivo ha cambiado, actualiza el programa con la nueva ruta y vuelve a intentarlo.
2. **Reparación del Archivo Excel:**
   - Si el archivo está dañado, intenta repararlo utilizando la función de reparación de Excel o restaurando una copia de seguridad.
3. **Reinstalación o Reparación de Excel:**
   - Si Excel tiene problemas para abrir el archivo, considera reparar la instalación de Microsoft Office o reinstalar Excel.
4. **Manejo de Errores en el Código:**
   - Mejora el manejo de errores en el código para que el sistema notifique al usuario en caso de que ocurra un problema al intentar abrir el archivo.



Procedimiento en caso de fallo en prototipo



Verificación Inicial

1. **Verificación de los cables**
   - Asegúrate de que todos los cables estén correctamente soldados y en su respectivo lugar
   - Si alguno de los cables se encuentra dañado, reemplázalo
2. **Comprobación de material estable:**
   - Verifica manualmente si todos los tornillos estén bien ajustados y que todo este correctamente armado sin que quede algún tornillo flojo
   - Asegúrate de que la madera no este dañada ni un tornillo de serlo reemplázalo y aprieta fijamente el tornillo
3. **Comprobación de las cajitas:**
   - Verifica que las cajitas estén fijamente colocadas sin perjudicar algún cable.
   - Verifica que ningún cable este dañado dentro de las cajitas

4.4.2 Anexo B Manual de Usuario

Integrantes:

Cadena Espinoza Braulio

Damian Ramirez Jonathan Ivan

Introducción

Bienvenido al manual de usuario del prototipo de control de salidas de garrafones de agua. Este sistema innovador combina software y hardware para gestionar de manera eficiente y precisa la distribución de garrafones de agua en una purificadora.

Nuestro objetivo es proporcionar una solución integral que facilite el control y monitoreo de las salidas de garrafones, asegurando que cada proceso se realice de manera óptima y sin errores. Este manual está diseñado para guiarte a través de la instalación, configuración y uso del sistema, ofreciendo instrucciones claras y detalladas para maximizar su rendimiento.

Componentes del Sistema

1. **Hardware**

   :

   - **Sensor de Peso**: Mide el peso de los garrafones para asegurar que estén llenos correctamente.

   - **Módulo de Control**: Gestiona las señales del sensor y controla las salidas.

2. **Software**:

   - **Interfaz de Usuario**: Permite la interacción con el sistema, mostrando datos en tiempo real y permitiendo ajustes según sea necesario.

Ingreso al sistema

Para comenzar a utilizar el sistema se deberá realizar el ingreso del PIN que a sido elegido por el cliente.

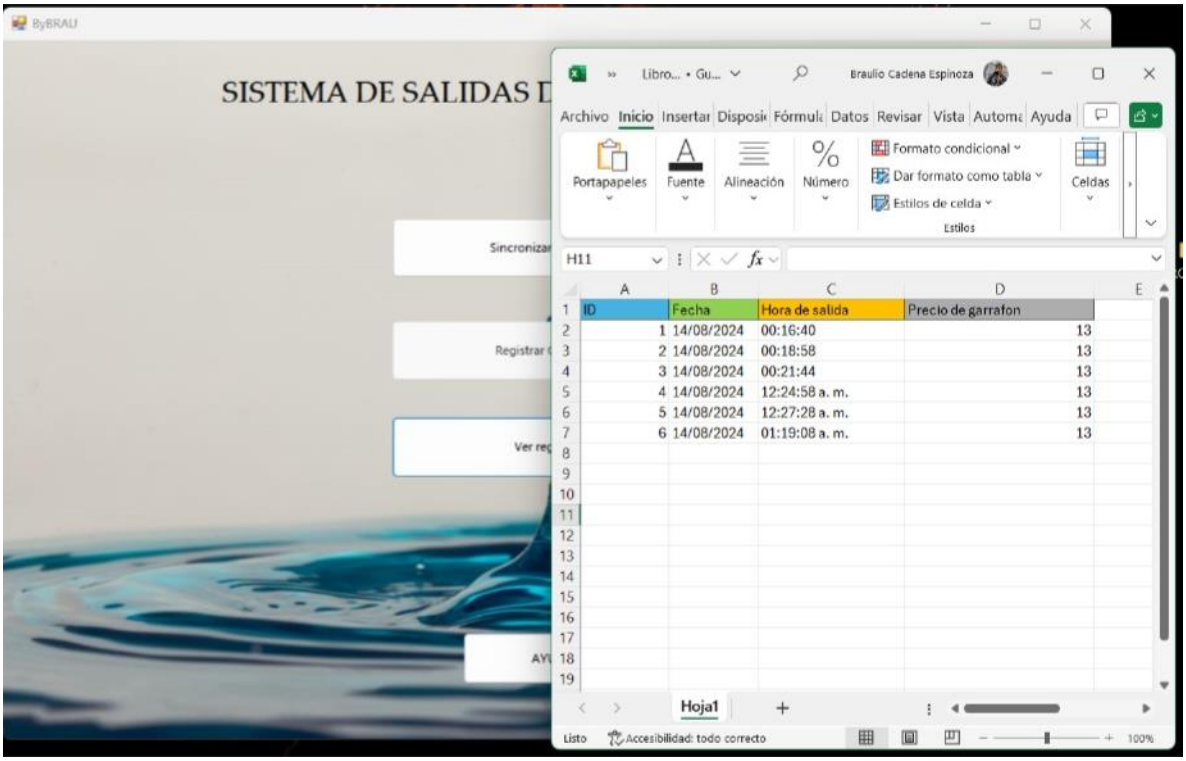
Una vez se ingreso el PIN se mostraran los botones con los que cuenta la interfaz.

En el momento en que el sensor reciba un peso lo mostrara en la interfaz y se deberá aceptar con un “Si” o denegar con un “No”.

Para visualizar que fue correcto el registro se deberá presionar el botón de “ver registro”

Visualización del Registro:

- Presione el botón **"VER REGISTRO"** para abrir el archivo Excel en modo solo lectura y revisar los datos registrados.



4.4.3 Anexo C Codificación del programa y BD

sketch_jul31a Arduino 1.8.19

Código Fuente Completo ARDUINO(CALIBRACIÓN):

```
#include <EEPROM.h>   // Incluye la biblioteca EEPROM
#include<Q2HX711.h>  // Incluye la biblioteca HX711
// Pines
const byte hx711_data_pin = 3;    // Data pin desde HX711
const byte hx711_clock_pin = 2;   // Clock pin desde HX711
Q2HX711 hx711(hx711_data_pin, hx711_clock_pin); // Preparar hx711

// Variables
float y1 = 914.0; // Masa calibrada ajustada basada en lecturas promedio
long x1 = 0L;
long x0 = 0L;
float avg_size = 10.0; // Cantidad de promedios para cada medición de masa
const float lower_noise_threshold = -0.50; // Límite inferior del rango de ruido
aceptable
const float upper_noise_threshold = 2.50;  // Límite superior del rango de ruido
aceptable

void setup() {
  Serial.begin(9600);                // Configura la velocidad de transmisión

  delay(1000);                       // Permite que la celda de carga y el hx711 se
estabilicen

  // Procedimiento de tara
  for (int ii = 0; ii < int(avg_size); ii++) {
```

```
    delay(10);
    x0 += hx711.read();
  }
  x0 /= long(avg_size);
  Serial.println("Add Calibrated Mass");

  // Procedimiento de calibración (la masa debe ser igual a y1)
  while (true) {
    long reading = hx711.read();
    if (reading < x0 + 10000) {
      // No hacer nada...
    } else {
      delay(2000);
      for (int jj = 0; jj < int(avg_size); jj++) {
        x1 += hx711.read();
      }
      x1 /= long(avg_size);
      break;
    }
  }
  Serial.println("Calibration Complete");

  // Guardar los valores calibrados en la EEPROM
  EEPROM.put(0, x0);
  EEPROM.put(sizeof(long), x1);
}

void loop() {
```

```
  // Promedio de lectura
  long reading = 0;
  for (int jj = 0; jj < int(avg_size); jj++) {
    reading += hx711.read();
  }
  reading /= long(avg_size);

  // Cálculo de masa basado en la calibración y ajuste lineal
  float ratio_1 = (float) (reading - x0);
  float ratio_2 = (float) (x1 - x0);

  // Verificar que ratio_2 no sea cero para evitar división por cero
  if (ratio_2 == 0) {
    Serial.println("Error: Invalid calibration");
    return;
  }
  float ratio = ratio_1 / ratio_2;
  float mass = y1 * ratio;
  // Ajustar el valor inicial a cero si está dentro del rango de ruido aceptable
  if (mass <= upper_noise_threshold && mass >= lower_noise_threshold) {
    mass = 0;
  }
  // Ajustar el valor de masa para que después de 999 salte a 1.0, 1.2, etc.
  if (mass > 999) {
    mass = 1.0 + (mass - 1000) / 1000.0;
  }  // Mostrar el peso en el Monitor Serial
  Serial.println(mass);
}
```

Código Fuente Completo visual:

```
Imports System.IO.Ports

Imports OfficeOpenXml

Imports System.IO


Public Class Form1

    Private WithEvents serialPort As SerialPort


    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

        Try

            serialPort = New SerialPort("COM6", 9600)

            serialPort.Open()

        Catch ex As UnauthorizedAccessException

            MessageBox.Show("No se puede acceder al puerto COM6. Asegúrese de
que no está en uso por otra aplicación y que tiene los permisos necesarios.",
"Error de Puerto Serial", MessageBoxButtons.OK, MessageBoxIcon.Error)

        Catch ex As Exception

            MessageBox.Show("Se produjo un error al abrir el puerto COM6: " &
ex.Message, "Error de Puerto Serial", MessageBoxButtons.OK,
MessageBoxIcon.Error)

        End Try

    End Sub


    Private Async Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

        Button1.Enabled = False

        Button2.Enabled = False


        Dim weight As String
```

```
        Dim weightValue As Double = 0
        Dim confirmDialogResult As DialogResult

        Do
            serialPort.DiscardInBuffer()

            weight = Await ReadWeightFromArduinoAsync()
            If Double.TryParse(weight, weightValue) Then
                ' Interpretar los valores según su longitud
                If weight.Length = 5 Then
                    MessageBox.Show($"Se ha detectado {weight} gramos. ¿Desea
continuar?", "Confirmar", MessageBoxButtons.YesNo)
                Else
                    confirmDialogResult = MessageBox.Show($"Se ha detectado
{weightValue} kg. ¿Desea continuar?", "Confirmar", MessageBoxButtons.YesNo)
                End If
            Else
                MessageBox.Show("Error al leer el peso. Inténtelo de nuevo.", "Error de
Lectura", MessageBoxButtons.OK, MessageBoxIcon.Error)
                Continue Do
            End If

            If confirmDialogResult = DialogResult.No Then
                Exit Do
            End If

            If confirmDialogResult = DialogResult.Yes AndAlso (weightValue = 2.1
OrElse weightValue = 3.0 OrElse weightValue >= 5.0) Then
                Button2.Enabled = True
                Exit Do
```

```
            End If
        Loop While weightValue < 2.1 OrElse (weightValue > 3.0 AndAlso
weightValue < 5.0)

        ' Si el peso es válido, habilitar el botón de registrar
        Button1.Enabled = True
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Dim pin As String = InputBox("Ingrese el PIN:", "PIN Requerido")

        If EncryptPIN(pin) = "e99a18c428cb38d5f260853678922e03" Then
            SaveToExcel()
            ' Desactivar el botón de registrar después de guardar en Excel
            Button2.Enabled = False
            ' Volver a habilitar el botón de sincronización de peso
            Button1.Enabled = True
        Else
            MessageBox.Show("PIN incorrecto.", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            ' Rehabilitar el botón de registrar si el PIN es incorrecto
            Button1.Enabled = True
        End If
    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
        ShowExcel()
    End Sub
```

```
    Private Async Function ReadWeightFromArduinoAsync() As Task(Of String)
        Return Await Task.Run(Function()
                                Try
                                    While serialPort.BytesToRead = 0
                                        Threading.Thread.Sleep(100)
                                    End While
                                    Dim data As String = serialPort.ReadLine().Trim()
                                    ' Convertir valores negativos a cero
                                    If data.StartsWith("-") Then
                                        data = "0.0"
                                    End If
                                    Return data
                                Catch ex As IOException
                                    MessageBox.Show("Error de lectura desde el puerto serial:
" & ex.Message, "Error de Puerto Serial", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                                    Return String.Empty
                                End Try
                            End Function)
    End Function
    Private Function EncryptPIN(pin As String) As String
        Dim md5 As System.Security.Cryptography.MD5 =
System.Security.Cryptography.MD5.Create()
        Dim hash As Byte() =
md5.ComputeHash(System.Text.Encoding.UTF8.GetBytes(pin))
        Dim sb As New System.Text.StringBuilder()
        For Each b As Byte In hash
            sb.Append(b.ToString("x2"))
        Next
        Return sb.ToString()
```

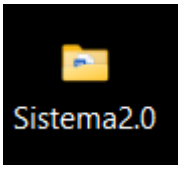
```
    End Function
    Private Async Sub SaveToExcel()
        Dim filePath As String =
"C:\Users\braup\OneDrive\Desktop\Sistema2.0\Libro1.xlsx"
        Dim fileInfo As New FileInfo(filePath)

        Try
            Using excelPackage As New ExcelPackage(fileInfo)
                Dim worksheet As ExcelWorksheet

                If excelPackage.Workbook.Worksheets.Count > 0 Then
                    worksheet = excelPackage.Workbook.Worksheets("Hoja1")
                Else
                    worksheet = excelPackage.Workbook.Worksheets.Add("Hoja1")
                    worksheet.Cells("A1").Value = "ID"
                    worksheet.Cells("B1").Value = "Fecha"
                    worksheet.Cells("C1").Value = "Hora"
                    worksheet.Cells("D1").Value = "Precio"
                End If

                Dim lastRow As Integer = worksheet.Dimension.End.Row + 1

                Dim weight As String = Await ReadWeightFromArduinoAsync()
                If String.IsNullOrEmpty(weight) Then
                    MessageBox.Show("Error al leer el peso. No se puede guardar en
Excel.", "Error de Lectura", MessageBoxButtons.OK, MessageBoxIcon.Error)
                    Return
                End If
```

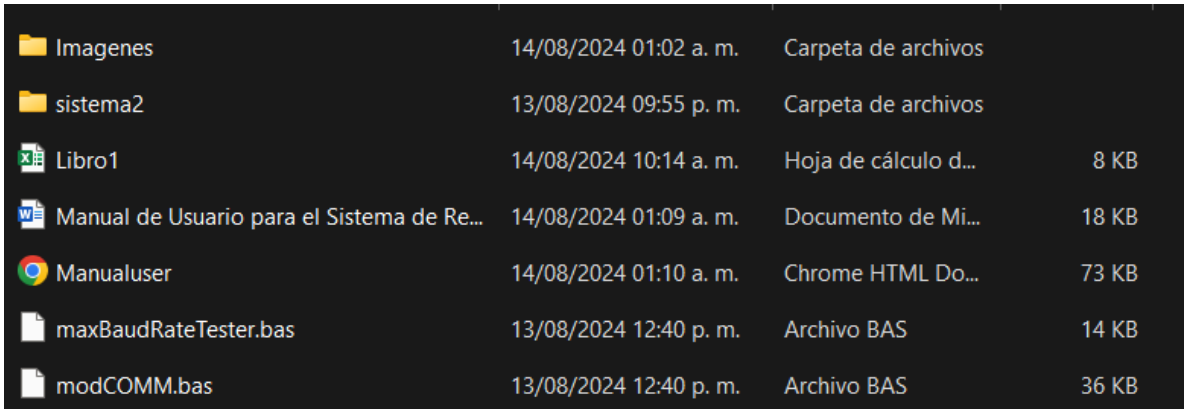
```
            Dim weightValue As Double = Double.Parse(weight)
            ' Determinar el precio en función del peso
            Dim price As Double = If(weightValue = 5, 6.5, 13)
            Dim currentDateTime As DateTime = DateTime.Now


            worksheet.Cells(lastRow, 1).Value = lastRow - 1
            worksheet.Cells(lastRow, 2).Value =
currentDateTime.ToString("dd/MM/yyyy")
            worksheet.Cells(lastRow, 3).Value =
currentDateTime.ToString("hh:mm:ss tt") ' Cambiado a formato 12 horas
            worksheet.Cells(lastRow, 4).Value = price
            excelPackage.Save()
        End Using
    Catch ex As IOException
        MessageBox.Show("Error al guardar el archivo Excel: " & ex.Message,
"Error de Archivo", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Sub
  Private Sub ShowExcel()
    Dim excelPath As String =
"C:\Users\braup\OneDrive\Desktop\Sistema2.0\Libro1.xlsx"
    Process.Start("EXCEL.EXE", $"/r ""{excelPath}""")
  End Sub
  Private Sub Form1_FormClosing(sender As Object, e As
FormClosingEventArgs) Handles MyBase.FormClosing
    If serialPort IsNot Nothing AndAlso serialPort.IsOpen Then
        serialPort.Close()
    End If
  End Sub
```

```
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
button4.Click

        Dim pdfPath As String =
"C:\Users\braup\OneDrive\Desktop\Sistema2.0\manualuser.pdf" ' Reemplaza con
la ruta de tu archivo PDF

        Try

            Process.Start(pdfPath)

        Catch ex As Exception

            MessageBox.Show("No se pudo abrir el archivo PDF. Asegúrese de que la
ruta es correcta y que tiene un lector de PDF instalado.", "Error al Abrir PDF",
MessageBoxButtons.OK, MessageBoxIcon.Error)

        End Try

    End Sub

End Class
```

Estructura del Proyecto:

Carpeta del proyecto





Imagenes	14/08/2024 01:02 a. m.	Carpeta de archivos	
sistema2	13/08/2024 09:55 p. m.	Carpeta de archivos	
Libro1	14/08/2024 10:14 a. m.	Hoja de cálculo d...	8 KB
Manual de Usuario para el Sistema de Re...	14/08/2024 01:09 a. m.	Documento de Mi...	18 KB
Manualuser	14/08/2024 01:10 a. m.	Chrome HTML Do...	73 KB
maxBaudRateTester.bas	13/08/2024 12:40 p. m.	Archivo BAS	14 KB
modCOMM.bas	13/08/2024 12:40 p. m.	Archivo BAS	36 KB

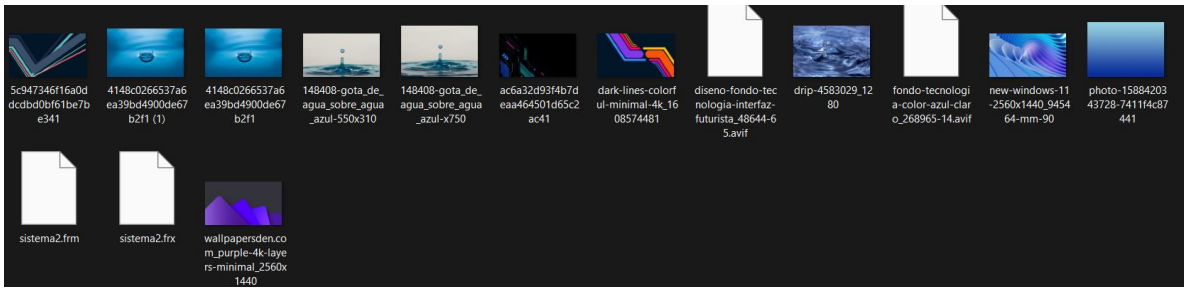
Manual de usuario

Modulo Comm para la conexión con excel

El módulo moddcommon se utiliza en programación para realizar operaciones específicas que pueden variar según el contexto en el que se aplique. En general, los módulos son componentes de software que contienen funciones, clases y variables que pueden ser reutilizadas en diferentes partes de un programa.

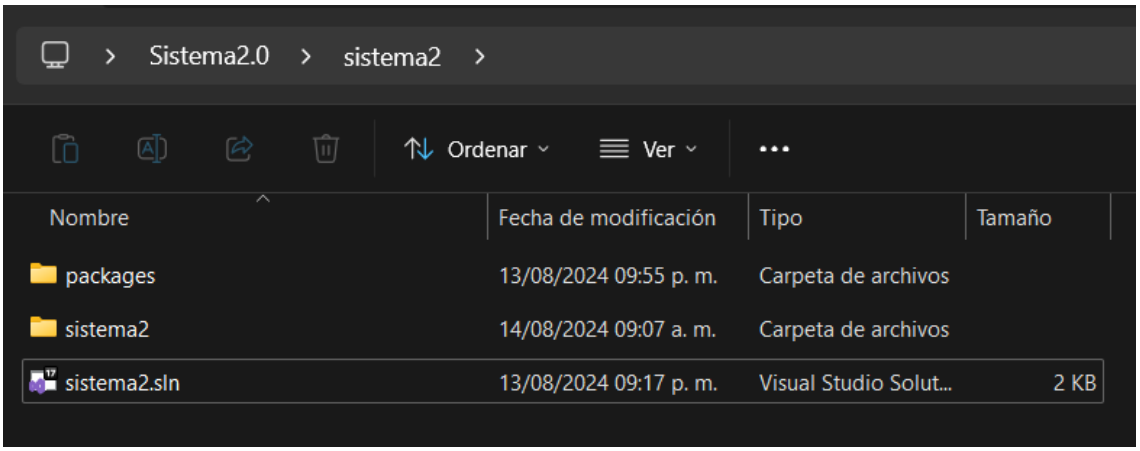
Carpeta imágenes

Personalización: Permite a los usuarios personalizar la apariencia de la interfaz según sus preferencias, lo que puede mejorar su experiencia y satisfacción.

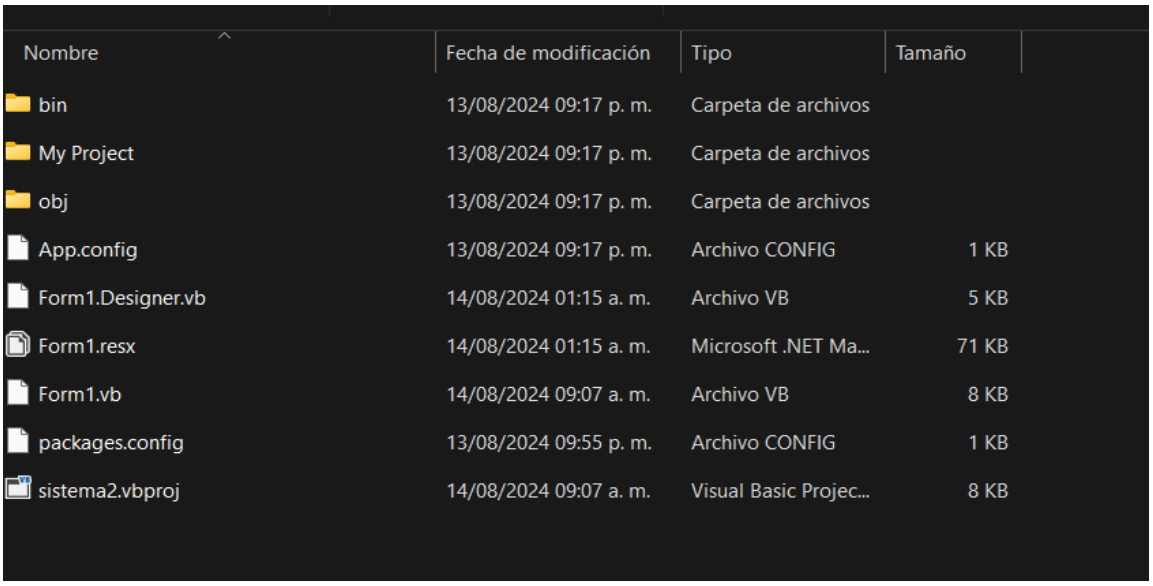


Segunda carpeta sistema

Los archivos con extensión .sln son archivos de solución de Visual Studio. Estos archivos contienen información sobre la organización de uno o más proyectos dentro de una solución. Aquí tienes algunos puntos clave sobre los archivos .sln:



Carpeta sistema 2 de suma importancia



Nombre	Fecha de modificación	Tipo	Tamaño
bin	13/08/2024 09:17 p. m.	Carpeta de archivos	
My Project	13/08/2024 09:17 p. m.	Carpeta de archivos	
obj	13/08/2024 09:17 p. m.	Carpeta de archivos	
App.config	13/08/2024 09:17 p. m.	Archivo CONFIG	1 KB
Form1.Designer.vb	14/08/2024 01:15 a. m.	Archivo VB	5 KB
Form1.resx	14/08/2024 01:15 a. m.	Microsoft .NET Ma...	71 KB
Form1.vb	14/08/2024 09:07 a. m.	Archivo VB	8 KB
packages.config	13/08/2024 09:55 p. m.	Archivo CONFIG	1 KB
sistema2.vbproj	14/08/2024 09:07 a. m.	Visual Basic Projec...	8 KB

bin: Esta carpeta contiene los archivos binarios compilados del proyecto. Los archivos binarios son el resultado de la compilación del código fuente y son ejecutables por la máquina.

obj: Esta carpeta almacena archivos de objeto intermedios generados durante la compilación. Estos archivos son utilizados por el compilador para crear el archivo binario final

.

App.config: Es un archivo de configuración de la aplicación. Aquí se definen parámetros y ajustes que la aplicación necesita para funcionar correctamente, como cadenas de conexión a bases de datos, configuraciones de logging, etc.

Form1.Designer.vb: Este archivo contiene el código generado automáticamente para el diseño del formulario en Visual Basic. Define la disposición y las propiedades de los controles en el formulario.

Form1.resx: Es un archivo de recursos asociado al formulario. Puede incluir cadenas de texto, imágenes y otros recursos que el formulario utiliza.

packages.config: Este archivo lista las dependencias del proyecto, especificando los paquetes NuGet necesarios. NuGet es un gestor de paquetes para .NET que facilita la instalación y actualización de bibliotecas.

site2sms.vbproj: Es el archivo de proyecto de Visual Basic. Contiene configuraciones y metadatos del proyecto, como las referencias a otros archivos y bibliotecas, configuraciones de compilación, etc.

Carpeta Packages

Nombre	Fecha de modificación	Tipo	Tamaño
EPPlus.7.2.2	13/08/2024 09:55 p. m.	Carpeta de archivos	
EPPlus.Interfaces.6.1.1	13/08/2024 09:55 p. m.	Carpeta de archivos	
EPPlus.System.Drawing.6.1.1	13/08/2024 09:55 p. m.	Carpeta de archivos	
Microsoft.IO.RecyclableMemoryStream.3....	13/08/2024 09:55 p. m.	Carpeta de archivos	
System.Buffers.4.5.1	13/08/2024 09:55 p. m.	Carpeta de archivos	
System.ComponentModel.Annotations.5....	13/08/2024 09:55 p. m.	Carpeta de archivos	
System.Memory.4.5.5	13/08/2024 09:55 p. m.	Carpeta de archivos	
System.Numerics.Vectors.4.5.0	13/08/2024 09:55 p. m.	Carpeta de archivos	
System.Runtime.CompilerServices.Unsafe....	13/08/2024 09:55 p. m.	Carpeta de archivos	

EPPlus.7.2.2: Carpeta que contiene la biblioteca EPPlus, utilizada para trabajar con archivos Excel en .NET.

EPPlus.Interfaces.6.1.1: Carpeta con las interfaces de la biblioteca EPPlus, que definen los contratos que las clases deben implementar.

EPPlus.System. Drawing.6.1.1: Carpeta que incluye funcionalidades de dibujo y gráficos para EPPlus.

Microsoft.IO.RecyclableMemoryStream.3.0.0: Carpeta con la biblioteca de Microsoft para gestionar flujos de memoria reciclables, mejorando el rendimiento y reduciendo la fragmentación de memoria.

System.Buffers.4.5.1: Carpeta que contiene la biblioteca para trabajar con buffers de datos en .NET.

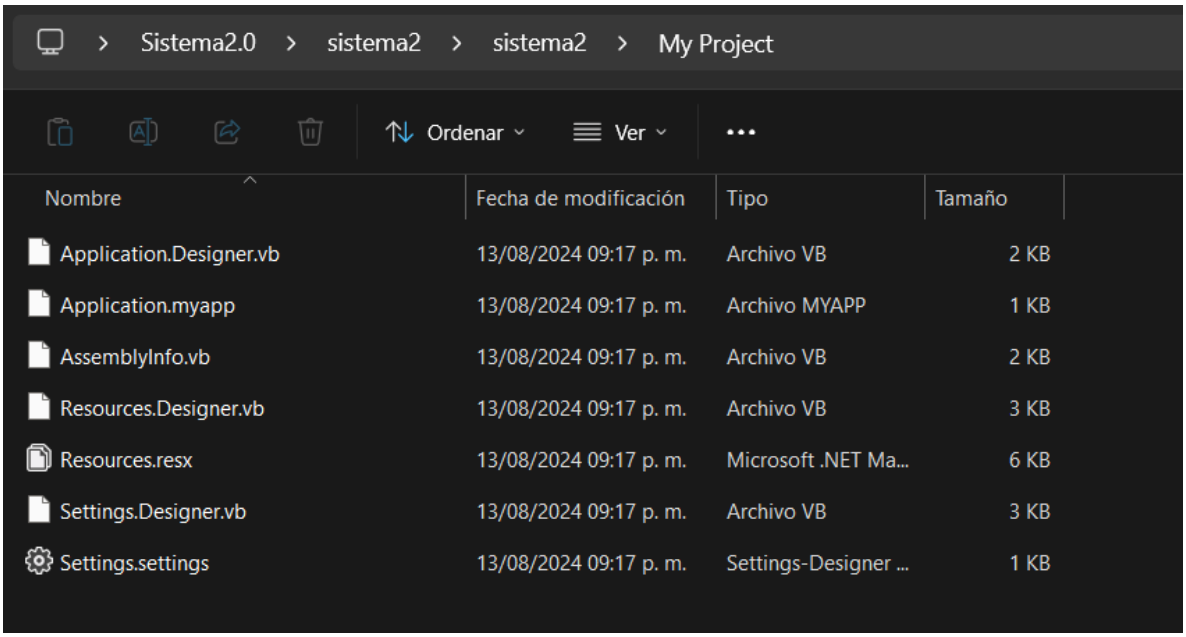
System.ComponentModel. Annotations.5.0.0: Carpeta con anotaciones para la validación de datos y otras funcionalidades en .NET.

System.Memory.4.5.5: Carpeta que incluye la biblioteca para trabajar con memoria y span en .NET.

System.Numerics.Vectors.4.5.0: Carpeta con la biblioteca para trabajar con vectores y operaciones numéricas en .NET.

System.Runtime. CompilerServices.Unsafe.4.5.2: Carpeta que contiene métodos para operaciones de bajo nivel y sin comprobaciones de seguridad en tiempo de ejecución.

Carpeta My Project



Application.Designer.vb: Archivo generado automáticamente que contiene el código para el diseño de la aplicación en Visual Basic.

Application.myapp: Archivo de configuración específico de Visual Basic que define las propiedades y configuraciones de la aplicación.

AssemblyInfo.vb: Archivo que contiene información sobre el ensamblado, como el nombre, la versión y otros atributos.

Resources.Designer.vb: Archivo generado automáticamente que contiene el código para gestionar los recursos de la aplicación, como imágenes y cadenas de texto.

Resources.resx: Archivo de recursos que almacena datos como cadenas de texto, imágenes y otros recursos utilizados por la aplicación.

Settings.settings: Archivo que define las configuraciones de la aplicación, permitiendo almacenar y recuperar configuraciones de usuario y de aplicación.

Detalles de la Base de Datos:

Excel no es una base de datos relacional como SQL Server o MySQL. Sin embargo, se puede usar Excel para gestionar y organizar datos de manera estructurada, especialmente para proyectos más pequeños o cuando no se necesita la complejidad de una base de datos completa.

	A	B	C	D
1	ID	Fecha	Hora de salida	Precio de garrafon
2	1	14/08/2024	00:16:40	13
3	2	14/08/2024	00:18:58	13
4	3	14/08/2024	00:21:44	13
5	4	14/08/2024	12:24:58 a. m.	13
6	5	14/08/2024	12:27:28 a. m.	13
7	6	14/08/2024	01:19:08 a. m.	13
8	7	14/08/2024	08:29:02 a. m.	13
9	8	14/08/2024	09:08:00 a. m.	13
10	9	14/08/2024	09:08:26 a. m.	13
11	10	14/08/2024	09:30:32 a. m.	13
12	11	14/08/2024	10:14:51 a. m.	13

Excel permite crear tablas, usar fórmulas y funciones para relacionar datos, y generar informes y gráficos. Aunque no tiene todas las capacidades de una base de datos relacional, puede ser una herramienta útil para ciertos tipos de proyectos.

- ✓ **Entidades**:
  - **Garrafón**: Representa cada garrafón de agua.
  - **Registro**: Representa cada registro de llenado.
- ✓ **Atributos**:
  - **Garrafón**:
    - ID_Garrafón (Clave primaria)
    - Litraje
    - Precio
  - **Registro**:
    - ID_Registro (Clave primaria)
    - Fecha
    - Hora_Salida
    - ID_Garrafón (Clave foránea)
- ✓ **Relaciones**:
  - Un **Garrafón** puede tener muchos **Registros** (Relación uno a muchos).