



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Apizaco

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

**DISEÑO DE UNA ARQUITECTURA PARA SISTEMA DE
CONTROL INTEGRAL, PARA EL DESPLIEGUE
DE KPI'S DE PORCELANITE – LAMOSA, PLANTA GRES**

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TITULO PROFESIONAL DE:
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES.**

**PRESENTA:
ALEJANDRO RIVERA MUÑOZ**

**NOMBRE DEL ASESOR:
M. JOSÉ ANTONIO CRUZ ZAMORA**



APIZACO, TLAX., FEBRERO 2024



Instituto Tecnológico de Apizaco
Sistemas y Computación

Apizaco, Tlaxcala, **13/junio/2023**

OFICIO No: SC/OPV/149/23

ASUNTO: Liberación de proyecto
para titulación integral

MARTÍN ROJAS RAMÍREZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	ALEJANDRO RIVERA MUÑOZ
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	18371006
NOMBRE DEL PROYECTO:	DISEÑO DE UNA ARQUITECTURA PARA SISTEMA DE CONTROL INTEGRAL PARA EL DESPLIEGUE DE KPI's DE PORCELANITE- LAMOSA, PLANTA GRES.
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar*


ALAN AUGUSTO GALLEGOS CUELLAR
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
AAGC/mrol*




JOSE ANTONIO CRUZ ZAMORA

ASESOR INTERNO



Carretera Apizaco-Tzompantepec, esquina con Av. Instituto Tecnológico S/N,
Conurbado Apizaco-Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010 Ext. 115,
e-mail: sistemas@apizaco.tecnm.mx | apizaco.tecnm.mx



2022 Flores
Año de
Magón
PRELUDIO DE LA REVOLUCIÓN MEXICANA



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Apizaco, Tlax., 12/febrero/2024

No. OFICIO: DEP-CT/130/2024

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

ALEJANDRO RIVERA MUÑOZ
EGRESADO DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL: 18371006
P R E S E N T E

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** cuyo tema es: **"DISEÑO DE UNA ARQUITECTURA PARA SISTEMA DE CONTROL INTEGRAL, PARA EL DESPLIEGUE DE KPI'S DE PORCELANITE-LAMOSA, PLANTA GRES"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga:

AUTORIZACIÓN DE IMPRESION

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envió un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®


MARTÍN ROJAS RAMÍREZ

JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

c.p. archivo



Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec, Municipio de Tzompantepec,
Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010 Ext. 304, e-mail:
dep_titulacion@apizaco.tecnim.mx



2024
Felipe Carrillo
PUERTO

GOBIERNO DEL ESTADO DE OAXACA
SECRETARÍA DE EDUCACIÓN
Y CULTURA

Agradecimientos.

Es con gran emoción y gratitud que presento este informe de residencias profesionales. Durante este período, tuve la oportunidad de sumergirme en el mundo laboral y aplicar los conocimientos adquiridos durante mi formación académica. Me complace expresar mi profundo agradecimiento a todas las personas e instituciones que hicieron posible esta experiencia enriquecedora.

Quiero expresar mi gratitud a mis familiares y seres queridos, especialmente a mis padres: Diana y Juan Manuel. Su apoyo incondicional, palabras de aliento y comprensión han sido fundamentales para mi éxito y bienestar durante este período. Agradezco su paciencia y comprensión mientras me dedicaba a mis residencias profesionales, los quiero. *Sin duda, los mejores padres...*

A ti, por la continua motivación y las gratas experiencias durante este proceso de formación profesional, superando grandes retos en este camino, para poder llegar hasta este día, a unos cuantos pasos de finalizar, espero con ansias aquello que nos espera en el futuro.

Agradezco sinceramente a mi asesor interno de residencias profesionales, por su invaluable orientación y apoyo durante mi estadía en la empresa.

Me gustaría agradecer especialmente a mi asesor externo por su apoyo y orientación constante durante mis residencias profesionales. Su mentoría y amistad han sido invaluable, y estoy agradecido por su disposición para compartir conocimientos y experiencias, fundamentales para mi crecimiento y desarrollo profesional. Agradecido por la oportunidad de aprender de usted y por brindarme una visión más amplia de mi campo de estudio.

En resumen, me siento enormemente agradecido por todas las personas que han contribuido a mi desarrollo profesional y personal durante mis residencias profesionales. Esta experiencia ha sido una oportunidad invaluable para crecer y aprender, y estoy emocionado por lo que el futuro me depara.

¡Muchas gracias!

Atentamente, Rivera Muñoz Alejandro </>

Resumen.

En este resumen se busca presentar de manera concisa los aspectos más destacados del proyecto, “Diseño de una arquitectura para el despliegue de KPI’s de Porcelanite – Lamosa, planta Gres”, en el cual, desde el inicio del proyecto nos encontramos con algunos obstáculos con el proceso del levantamiento de requerimientos del sistema, por lo que desde esa etapa ya se podía intuir que no sería fácil este proyecto, poco a poco las cosas fueron surgiendo, permitiéndonos recabar los requisitos correspondientes para el análisis, viabilidad y el desarrollo del proyecto.

Al obtener los requerimientos comenzamos con el análisis de la tecnología que se usarían en el proceso de desarrollo, al tener un presupuesto nulo, se llegó a la conclusión hacer uso de las tecnologías que teníamos a la mano, por lo que la opción más factible fue desarrollar con herramientas low code, que a su vez también se convirtieron en Low cost, ya que la empresa contaba con cuentas para el uso de las herramientas de Microsoft.

La suite de Microsoft, específicamente la Power Platform, nos permitió realizar el desarrollo del sistema planteado haciendo uso de herramientas como Power apps, Power BI y Power Automate para hacer una integración de estas tecnologías y realizar dicho sistema sin inconvenientes.

A mitad del desarrollo nos encontramos con un problema que podía comprometer la mitad del funcionamiento del sistema, ya que en algunas áreas no cuentan con conexión a internet, no podían hacer el ingreso de información, por lo que después de un análisis e investigaciones se pudo implementar una funcionalidad offline para almacenar los datos en el dispositivo cuando las áreas no cuenten con internet para el envío de datos.

Una vez teníamos el método de captura de información, necesitábamos consumir dicha información y mostrarla para realizar un análisis estadístico de la planta en tiempo real, por lo que es aquí donde entra la plataforma de Power BI, la cual nos permite consumir la base de datos y mostrarlo en un dashboard construido para ser visto por operadores, supervisores, jefes y gerencia.

Finalmente, ya que la empresa usa Excel para la creación de reportes, se usó Power Automate para ingresar los datos seleccionados en los flujos a un documento en Excel específico cada que se detectara un nuevo registro en la base de datos y generara el reporte de manera automática.

De esta forma, pudimos optimizar varios procesos de la planta los cuales son repetitivos y tardados además que también se contribuyó para la obtención de una certificación de la empresa, ya que con ayuda de este sistema pueden estar más cerca de la obtención.

Índice.

AGRADECIMIENTOS	2
RESUMEN.....	5
GENERALIDADES DEL PROYECTO.....	9
INTRODUCCIÓN.....	9
DESCRIPCIÓN DE LA EMPRESA.....	9
PROBLEMAS PARA RESOLVER.....	10
OBJETIVOS.....	10
○ <i>Objetivo general</i>	10
○ <i>Objetivos específicos</i>	10
JUSTIFICACIÓN	11
MARCO TEÓRICO	11
CONCEPTOS Y TECNOLOGÍAS USADAS.....	11
<i>Power apps</i>	11
<i>Power BI</i>	11
<i>Power Automate</i>	12
<i>DBdiagram.io</i>	12
<i>Diagramas de flujo</i>	12
<i>Casos de uso</i>	13
<i>SQL</i>	13
<i>MySQL</i>	14
<i>Arquitectura 4+1</i>	14
DESARROLLO	16
DETERMINACIÓN DE REQUERIMIENTOS	16
<i>Recopilación de la información</i>	16
<i>Análisis de la información</i>	16
<i>Análisis de las tecnologías de desarrollo que se usaran</i>	17
DISEÑO DEL SISTEMA	18
<i>Diseño de la base de datos</i>	18
<i>Diseño de los diagramas de funcionamiento</i>	18
Diagrama de flujo (Funcionamiento general del sistema, SCI)	18
Diagrama de flujo (Logística operacional del sistema)	20
Diagrama de casos de uso (SCI)	21
Distribución de indicadores KPI's.....	22
Diagrama (Vista Física del sistema)	22
<i>Diseño de interfaces (SCI)</i>	23
.....	26
Interfaces del dashboard (Power Bi)	27
DESARROLLO DEL SISTEMA	29
CREACIÓN DE LA BASE DE DATOS.....	29
DESARROLLO DEL FRONTEND.....	31
DOCUMENTACIÓN SCI POWER APPS.....	32
<i>Edición de la aplicación (SCI)</i>	33
<i>Agregar nuevas pantallas</i>	34
<i>Navegación entre pantallas</i>	36

<i>Agregar componentes (inputs, etiquetas, galerías, etc.)</i>	37
<i>Listas desplegables</i>	38
<i>Agregar datos manualmente a listas desplegables</i>	39
<i>Agregar datos mediante orígenes de datos</i>	39
<i>Envío de información a bases de datos</i>	41
<i>Colecciones</i>	43
<i>Datos sin conexión a internet</i>	45
<i>Agregar colecciones o datos a una galería</i>	47
DESARROLLO DEL BACKEND	48
PRUEBAS	51
PRUEBAS DE EFICIENCIA	53
PRUEBAS DE RENDIMIENTO	53
PRUEBAS DE USABILIDAD	53
PRUEBAS DE ACCESIBILIDAD	53
RESULTADOS	54
CONCLUSIONES	54
COMPETENCIAS DESARROLLADAS	55
FUENTES DE INFORMACIÓN	55

Tabla de ilustraciones.

ILUSTRACIÓN 1DIAGRAMA DE LA BASE DE DATOS	18
ILUSTRACIÓN 2DIAGRAMA DE FUNCIONAMIENTO DEL SISTEMA	19
ILUSTRACIÓN 3LOGÍSTICA OPERACIONAL DEL SISTEMA	20
ILUSTRACIÓN 4CASOS DE USO - SCI	21
ILUSTRACIÓN 5DISTRIBUCIÓN DE INDICADORES KPI'S	22
ILUSTRACIÓN 6 VISTA FÍSICA DEL SISTEMA	23
ILUSTRACIÓN 7 PANTALLA DE INICIO - APLICACIÓN SCI	23
ILUSTRACIÓN 8 HOMESCREEN SCI.....	24
ILUSTRACIÓN 9 PANTALLAS DE LOGIN Y SELECCIÓN DE INDICADORES KPI'S	25
ILUSTRACIÓN 10 PANTALLA DE NAVEGACIÓN EN MÓDULO DE CP	25
ILUSTRACIÓN 11 PANTALLA DE VISUALIZACIÓN DE DATOS OFFLINE	26
ILUSTRACIÓN 12 INTERFACES POWER BI - KPI'S.....	27
ILUSTRACIÓN 13INTERFACES POWER BI - CONTROL DE PROCESO	28
ILUSTRACIÓN 14MODELADO DE LA BASE DE DATOS	29
ILUSTRACIÓN 15 ARCHIVO SQL DE LA BASE DE DATOS	30
ILUSTRACIÓN 16 PROCESO DE DESARROLLO DE LA APLICACIÓN	31
ILUSTRACIÓN 17 DESARROLLO DE FORMULARIOS EN POWER APPS.....	31
ILUSTRACIÓN 18 DEMOSTRACIÓN DE INTERFACES EN POWER APPS	32
ILUSTRACIÓN 20SELECCIÓN KPI's O SCI	32
ILUSTRACIÓN 20PANTALLA DE INICIO	32
ILUSTRACIÓN 21EDICIÓN DE APLICACIÓN EN POWER APPS	33
ILUSTRACIÓN 22 SELECCIÓN DE LA APLICACIÓN A EDITAR	33
ILUSTRACIÓN 23 INTERFAZ DE EDICIÓN EN POWER APPS	34
ILUSTRACIÓN 24 CREACIÓN DE NUEVAS PANTALLAS	35

ILUSTRACIÓN 25 RENOMBRAMIENTO Y DUPLICIDAD DE PANTALLAS	36
ILUSTRACIÓN 26 FORMULA PARA NAVEGACIÓN ENTRE PANTALLAS	36
ILUSTRACIÓN 27 COMPONENTES EN POWER APPS	37
ILUSTRACIÓN 28 COMPONENTES Y COMO IDENTIFICARLOS PARA USO EN FORMULAS.....	38
ILUSTRACIÓN 29 USO DE LISTAS DESPLEGABLES.....	38
ILUSTRACIÓN 30 FORMA MANUAL PARA INGRESO DE DATOS A LISTAS	39
ILUSTRACIÓN 31 AGREGAR DATOS A LISTAS MEDIANTE ORÍGENES DE DATOS O COLECCIONES\	39
ILUSTRACIÓN 32 DEMOSTRACIÓN DE COMPONENTE COMBO BOX CON COLECCIONES	40
ILUSTRACIÓN 33 IDENTIFICADOR DE COMPONENTE PARA USO EN EL CÓDIGO DE ENVÍO DE DATOS	41
ILUSTRACIÓN 34 PROPIEDAD ONSELECT DE COMPONENTES	41
ILUSTRACIÓN 35 CÓDIGO PARA ENVÍO DE INFORMACIÓN A BASE DE DATOS	42
ILUSTRACIÓN 36 CREACIÓN DE COLECCIÓN PARA DATOS OFFLINE	43
ILUSTRACIÓN 37 RESET DE COMPONENTES	43
ILUSTRACIÓN 38 PROPIEDAD ONSTART DEL ARCHIVO DE INICIO DE LA APLICACIÓN PARA CARGA DE INFORMACIÓN.....	44
ILUSTRACIÓN 39 INSTRUCCIONES PARA CARGA DE INFORMACIÓN GUARDADA EN EL DISPOSITIVO	44
ILUSTRACIÓN 40 ICONO DE ENVÍO A VISTA DE DATOS OFFLINE.....	45
ILUSTRACIÓN 41 PANTALLA DE VISUALIZACIÓN DE DATOS OFFLINE	45
ILUSTRACIÓN 42 MÉTODO DE ENVÍO DE DATOS ALMACENADOS EN EL DISPOSITIVO CON THISITEM.....	46
ILUSTRACIÓN 43 USO DE FUNCIÓN THISRECORD PARA ENVÍO DE DATOS DE MANERA MASIVA.....	47
ILUSTRACIÓN 44 CONEXIÓN DE GALERÍAS A ORÍGENES/COLECCIONES DE DATOS.....	47
ILUSTRACIÓN 45 CODIGO PARA ENVÍO DE INFORMACIÓN A LA BASE DE DATOS.....	48
ILUSTRACIÓN 46 CÓDIGO PARA ENVÍO DE INFORMACIÓN OFFLINE - INDIVIDUAL	49
ILUSTRACIÓN 47 CÓDIGO PARA ENVÍO DE INFORMACIÓN OFFLINE - CARGA MASIVA DE DATOS	49
ILUSTRACIÓN 48 CÓDIGO PARA LA CARGA DE DATOS ALMACENADOS AL INICIO DE APLICACIÓN.....	50
ILUSTRACIÓN 49 DISTRIBUCIÓN DEL SISTEMA EN LÍNEA PILOTO PARA PRUEBAS.....	52

Generalidades del proyecto.

Introducción.

Porcelanite – Lamosa, comenzó la producción en el año de 1933, siendo la primera empresa mexicana en fabricación de azulejos, pisos y barro blocks, posteriormente, años después, Grupo Lamosa S.A de C.V adquiere la empresa de Porcelanite para continuar con la tendencia de crecimiento hasta la actualidad, poco a poco ha ido creciendo esta empresa, y actualmente es considerada una empresa internacional, por lo que se puede imaginar la magnitud de la empresa. Debido al gran tamaño y específicamente en planta Gres, se genera una gran cantidad de información, por lo que con el tiempo se puede llegar a dificultar la captura, análisis y visualización de dicha información, cabe mencionar que no se cuenta con una forma estandarizada de información, por lo que eso lo vuelve un poco más complejo y menos accesible para la empresa.

Dicho lo anterior, se propone el desarrollo de un proyecto que permita la captura, visualización y análisis de datos más eficiente para la planta, mediante el desarrollo de una arquitectura de software para un sistema de control integral que permita el despliegue de KPI's de la planta, este sistema se compone de una aplicación que permita la captura de información de manera más eficiente y estandarizada con interfaces amigables para los usuarios, solicitando solo la información requerida para el análisis y la visualización de los datos, la información será almacenada en una base de datos, la cual será consumida por PowerBi, tecnología de Microsoft que permite la visualización grafica de los datos según sea requerida permitiendo a la planta una mejor toma de decisiones para el continuo crecimiento de la planta y la mejora continua.

Descripción de la empresa.

La historia como Grupo Lamosa comienza en 1890 con la fundación de la compañía manufacturera de ladrillos Monterrey, mismo año en el que fue adquirida por los ingenieros Bernardo Elosúa y Viviano Valdés, convirtiéndose en Ladrillera Monterrey S.A. La empresa Porcelanite, inicia su producción en el año 1933, convirtiéndola en la primera empresa mexicana en la fabricación de azulejos, pisos y barro blocks, fue hasta el año de 1995 que Ladrillera Monterrey, modifica su razón social por la de Grupo Lamosa S.A de C.V, mismo año en el que adquiere a la empresa Porcelanite para continuar con la tendencia de crecimiento vista hasta hoy en día, convirtiéndola en Porcelanite – Lamosa.

Porcelanite fue galardonado con el premio “Best in show” durante su participación en la expo “Coverings” con sede en Las Vegas, Nevada, “seguiremos construyendo una historia que merezca la pena contarse”. Porcelanite. (2019, 7 octubre).

Problemas a resolver.

1. Captura de datos poco eficiente.
 - 1.1. Actualmente la captura de información de la empresa se ha vuelto poco eficiente, ya que la captura actual, conlleva una gran cantidad de tiempo, por lo que se plantea reducir los tiempos de captura de la información.
2. Información no estandarizada.
 - 2.1. La información que se captura no suele estar en una forma estándar, por lo que muchas veces la información no sigue un formato específico.
3. Duplicación de documentos e información.
 - 3.1. Al manejar distintos documentos y enviarlos a más personal, se crea un duplicado de los archivos y de la información, por lo que se ocupa gran cantidad de almacenamiento en los dispositivos y en el servidor local de la empresa.
4. Información dispersa.
 - 4.1. La información de la empresa se encuentra dispersa, por lo que para poder acceder a cierta información se vuelve en ocasiones un problema, por lo que se plantea centralizar dicha información mediante el proyecto, de esta forma la información será accesible para el personal autorizado que desee consultarla.

Objetivos.

- **Objetivo general.**
 - Diseñar una arquitectura para sistema de control integral de porcelanite – Lamosa, planta Gres, permitiendo desplegar los KPI's de la empresa, agilizando los procesos de visualización, captura, y análisis de los datos.
- **Objetivos específicos.**
 1. Desarrollar la arquitectura para un sistema de despliegue de KPI's de la empresa.
 2. Desarrollo de interfaces amigables e intuitivas para los usuarios.
 3. Uso de tecnologías low code y low cost.
 4. Optimizar y agilizar los procesos de visualización, captura y análisis de datos de la empresa.
 5. Despliegue en producción para ser usado por los usuarios en sus actividades laborales.

Justificación.

El presente proyecto, se enfocará en el diseño de una arquitectura, para el desarrollo de un sistema de control integral, el cual permitirá el despliegue de los KPI's de la empresa, ya que actualmente se lleva a cabo este proceso bajo un sistema poco eficiente para el negocio, por lo que este proyecto permitirá agilizar y mejorar el control de este proceso, reduciendo tiempos en la captura de información, lo cual permite a los trabajadores enfocarse en otras actividades.

Marco Teórico.

Conceptos y tecnologías usadas.

Power apps.

“Power Apps es un conjunto de aplicaciones, servicios y conectores, así como una plataforma de datos que proporciona un entorno de desarrollo de aplicaciones ágil para crear aplicaciones personalizadas para las necesidades de su empresa. Al usar Power Apps, puede crear aplicaciones empresariales de forma rápida que se conectan a los datos de su negocio almacenados en la plataforma de datos subyacentes (Microsoft Dataverse) o en varios orígenes de datos locales y en línea (como SharePoint, Microsoft 365, Dynamics 365, SQL Server, etc.).

Las aplicaciones creadas usando Power Apps ofrecen una completa lógica de negocios y capacidades de flujo de trabajo con el fin de transformar las operaciones empresariales manuales para procesos digitales y automatizados. Además, las aplicaciones creadas con Power Apps presentan un diseño dinámico y pueden ejecutarse sin problemas en un explorador y en dispositivos móviles (teléfono o tableta). Power Apps "democratiza" la experiencia de creación de aplicaciones empresariales personalizadas ya que permite a los usuarios crear aplicaciones empresariales personalizadas con múltiples características sin escribir código.

Power Apps también proporciona una plataforma extensible que permite a los desarrolladores profesionales interactuar mediante programación con datos y metadatos, aplicar lógica empresarial, crear conectores personalizados e integrarse con datos externos.”

(KumarVivek, 2023)

Power BI

Power BI es una colección de servicios de software, aplicaciones y conectores que funcionan conjuntamente para convertir orígenes de datos sin relación entre sí en información coherente,

interactiva y atractiva visualmente. Sus datos podrían ser una hoja de cálculo de Excel o una colección de almacenes de datos híbridos locales y basados en la nube. Power BI permite conectarse con facilidad a los orígenes de datos, visualizar y descubrir qué es importante y compartirlo con cualquiera o con todos los usuarios que desee.

(Mihart, 2023)

Power Automate.

Power Automate es una herramienta capaz de conectar dos o más aplicaciones (tanto de Office 365 como de terceros) y realizar acciones con la finalidad de automatizar procesos empresariales rutinarios. Esta aplicación puede enviar recordatorios sobre tareas vencidas, puede mover datos de negocios entre sistemas en una programación, puede hablar con más de 275 orígenes de datos o cualquier API disponible de acceso público, e incluso puede automatizar tareas en su ordenador local como el cálculo de datos en Excel. Todo esto lo pueden hacer todo tipo de usuarios, gracias a su plataforma sin código/de poco código.

Power Automate es el nuevo nombre del antiguo Microsoft Flow. Microsoft ha decidido renombrar la aplicación para darle cabida en Power Platform. El nuevo sistema empresarial de Microsoft nos permitirá integrar y sincronizar todo nuestro análisis de datos con flujos automatizados y aplicaciones. Power Automate está pensado para aumentar nuestra productividad y eficiencia empresarial.

(Bartumeu, 2022)

DBdiagram.io

dbdiagram.io es una herramienta sencilla y gratuita para dibujar diagramas de base de datos (ERD) escribiendo código DSL. dbdiagram utiliza el popular DBML (lenguaje de marcado de base de datos).

(dbdocs.io - 2023)

Diagramas de flujo.

Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender. Los diagramas de flujo emplean rectángulos, óvalos, diamantes y otras numerosas figuras para definir el tipo de paso, junto con flechas conectoras que establecen el flujo y la secuencia. Pueden variar desde diagramas simples

y dibujados a mano hasta diagramas exhaustivos creados por computadora que describen múltiples pasos y rutas. Si tomamos en cuenta todas las diversas figuras de los diagramas de flujo, son uno de los diagramas más comunes del mundo, usados por personas con y sin conocimiento técnico en una variedad de campos. Los diagramas de flujo a veces se denominan con nombres más especializados, como "diagrama de flujo de procesos", "mapa de procesos", "diagrama de flujo funcional", "mapa de procesos de negocios", "notación y modelado de procesos de negocio (BPMN)" o "diagrama de flujo de procesos (PFD)". Están relacionados con otros diagramas populares, como los diagramas de flujo de datos (DFD) y los diagramas de actividad de lenguaje unificado de modelado (UML).

(Lucidchart, 2023)

Casos de uso.

Un *caso de uso* es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable. Los casos de uso proporcionan una estructura para expresar requisitos funcionales en el contexto de procesos empresariales y de sistema. Los casos de uso pueden representarse como un elemento gráfico en un diagrama y como una especificación de caso de uso en un documento textual.

Un caso de uso de sistema es una secuencia de acciones que un sistema lleva a cabo que da lugar a un resultado de valor observable para un actor particular (alguien o algo fuera del sistema que interactúa con el sistema). Un diagrama de caso de uso puede incluir varios casos de uso y las relaciones entre casos de uso y las personas, los grupos o los sistemas que interactúan para llevar a cabo el caso de uso.

(IBM, 2021)

SQL

Se inventó en la década de 1970 con base en el modelo de datos relacional. AL inicio se conocía como el lenguaje de consultas estructuradas en inglés (SEQUEL). Mas tarde, el término se abrevió a SQL. Oracle, antes conocido como Relational Software, se convirtió en el primer proveedor en ofrecer un sistema comercial de administración de bases de datos relacionales SQL.

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar,

eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos.

(Amazon Web Services, 2023)

MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

(Robledano, 2023)

Arquitectura 4+1

La Arquitectura de Software constituye un diseño de alto nivel del sistema. Una forma de representarlo es mediante el Modelo de Vistas de Arquitectura “4+1”. El estándar se ha perfilado como referente en el ámbito del diseño. El modelo fue ideado en 1995 por Philippe Kruchten, un Ingeniero de Software Canadiense, para “describir la arquitectura de sistemas software, basados en el uso de múltiples puntos de vista concurrentes”. Se utiliza para organizar el software en un entorno de desarrollo.

En el modelo se proponen cuatro vistas (lógica, desarrollo, procesos y física) y una vista adicional (escenario) utilizada para vincular a las demás.

- ❑ Lógica View (vista lógica): Representa la funcionalidad que el sistema proporciona al o los usuarios finales, lo que el sistema debe hacer, las funciones y los servicios que debe ofrecer, etc.
- ❑ Development View (vista de desarrollo): Ilustra el sistema desde la perspectiva del programador, se ocupa de la gestión del software; muestra como está dividido el sistema, sus componentes y las dependencias entre estos. Esta vista también se conoce como vista de implementación o despliegue.

- ❑ **Process View (vista de proceso):** Trata los aspectos dinámicos del sistema, explica los procesos de sistema y cómo se comunican. Se enfoca en el comportamiento del sistema en ejecución. La vista considera aspectos de concurrencia, distribución, rendimiento, escalabilidad, etc.
- ❑ **Physical View (vista física):** Describe el sistema desde el punto de vista de un ingeniero de sistemas. Incluye todos los componentes físicos del sistema, así como las conexiones entre estos componentes que conforman la solución (incluyendo servicios).
- ❑ **Scenarios (escenarios):** La descripción de la arquitectura se ilustra utilizando un conjunto de casos de uso, o escenarios lo que genera una quinta vista. Los escenarios describen secuencias de interacciones entre objetos, y entre procesos. Se utilizan para identificar y validar el diseño de arquitectura. También sirven como punto de partida para pruebas de un prototipo de arquitectura. Esta vista es también conocida como vista de casos de uso.

(Platzi, 2020)

Desarrollo.

Determinación de requerimientos.

Recopilación de la información.

Se lleva a cabo la recopilación de la información y toma de requerimientos para el desarrollo del sistema de control integral para despliegue de KPI's, para esta actividad se hizo una visita a los jefes de área involucrados en el proceso, para hablar con ellos y preguntar los datos específicos que requerían visualizar y capturar en el sistema, de igual forma se realizó una investigación de campo, hablando con operadores para conocer cómo es su método de captura de información actual, conocer los procesos productivos y como se relacionan entre sí, todo esto para conocer de mejor forma la lógica de negocio que seguiríamos para el desarrollo del proyecto.

Análisis de la información.

En esta actividad se lleva a cabo el análisis correspondiente de la información recabada para definir el funcionamiento más óptimo para el sistema y pueda satisfacer las necesidades de los usuarios, durante el análisis de la información, se observó que los operadores de algunas áreas deben desplazarse a la computadora más cercana para poder capturarla información requerida lo cual hacia perder una gran cantidad de tiempo en esa actividad para los operadores, algunos operadores escribían la información en papel para posteriormente capturar la información a final de turno, por lo que a simple son actividades que son poco funcionales y mal optimizadas para los operadores, existen excepciones que permiten a los operadores capturar la información de una forma más rápida, ya que cuenta con equipo de cómputo a su disposición en su área de trabajo, por lo que la solución que se propuso para el desarrollo del sistema, fue realizar una aplicación que funcionara de igual forma en dispositivos móviles como computadoras, de esta forma los que tuvieran computadoras y dispositivos móviles podrían usar el sistema de igual manera.

Esta propuesta mejoraría considerablemente estas actividades, ya que de esta forma se optimizaría la captura de información y la información sería capturada en tiempo real y con el control de fecha y hora en la que los datos son ingresados, de esta forma se asegura que los datos son ingresados en el tiempo y fecha debido.

Otro problema que se observo es la conexión a internet, ya que existen áreas donde no es posible tener esta conexión y dificulta el proceso de captura, por lo que se propuso que el sistema cuente con una funcionalidad que permita a los usuarios capturar información de manera local, hasta que se puedan

trasladar a un área con conexión a internet y enviar la información capturada en ese turno, esto permite que se obtenga la información de todas las áreas sin alguna restricción.

Análisis de las tecnologías de desarrollo que se usaran.

Una vez analizada la información y tomando en consideración las propuestas de desarrollo para el sistema, se lleva a cabo el análisis de las tecnologías que se usaran para llevar a cabo el desarrollo del sistema. Uno de los objetivos para este proyecto es desarrollar el sistema haciendo uso de tecnologías low code y low cost, por lo que las opciones se veían más reducidas, por lo que se decidió usar las herramientas de la *Power Platform* de *Microsoft*, dichas herramientas son: Power Apps, Power Automate y Power BI.

Power Apps, nos permite desarrollar aplicaciones que pueden ser usadas en dispositivos móviles y computadoras, además que permite que el desarrollo sea más ágil y rápido, de esta forma se puede desarrollar el sistema para que sea compatible y soportado por móviles y ordenadores.

Power BI, nos permite consumir la información almacenada en la base de datos para mostrarla posteriormente a los jefes de área, gerencia y operadores diariamente y conocer el estado de la planta mediante el desarrollo de un dashboard que muestre la información gráficamente y legible para los usuarios.

Power Automate, permite la realización de flujos automatizado para soluciones en la nube o de escritorio, útil para realizar acciones en el sistema según ciertas condiciones que se definan en el flujo, ya que, al cumplirse ciertas condiciones establecidas, desencadenara la acción que le hayamos ordenado, por ejemplo; ingresar datos en documentos Excel, envió automático de emails, etc.

Por lo que usar Power Platform es la mejor opción a utilizar para el desarrollo, ya que no solo es una herramienta low code, también es low cost ya que la empresa hace uso de cuentas empresariales de Microsoft Office, por lo que ya no se genera un gasto extra de desarrollo.

MySQL,

Diseño del sistema.

Diseño de la base de datos.

Para el diseño de la base de datos que se usara en el sistema de control integral (SCI), se tomaron en cuenta todas las variables obtenidas durante la fase de recopilación de la información que se menciona anteriormente, conociendo dichas variables podremos construir la base de datos, ya que conocemos específicamente los datos que guardaremos para cada área de la planta junto con sus respectivas subáreas, en este caso se realizó una base de datos que almacena los datos de KPI's y Control de proceso, la cual debido a la gran cantidad de información requerida se convirtió en una base de datos robusta y extensa, siendo que el propósito es ingresar toda la información de la planta en la base de datos, la cual será alimentada por el sistema que se desarrolla.

para lograr la relación entre los datos, tomamos las variables principales y de mayor repetición en las áreas, las cuales son: Maquinas, Diseños, Supervisores, Fechas y Turnos, esta misma relación funciona para el modelado de los datos en Power BI

Diseño de los diagramas de funcionamiento.

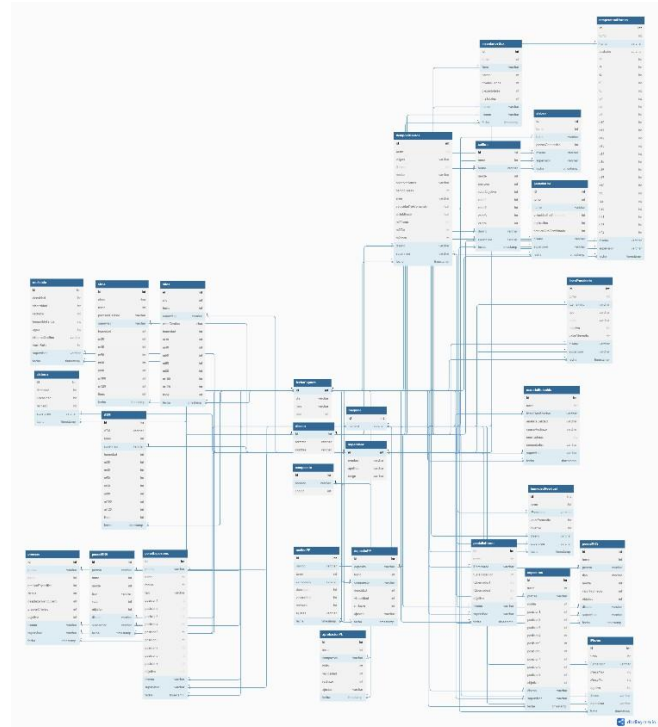


Ilustración 1 Diagrama de la base de datos

Diagrama de flujo (Funcionamiento general del sistema, SCI)

Se realiza diagrama de flujo para representar el funcionamiento del sistema SCI con la finalidad de poder plasmar correctamente las funcionalidades al momento de realizar el desarrollo del sistema, por

lo que en el diagrama se muestran los diferentes procesos que realiza el sistema, desde que se accede a la aplicación de captura, selecciona el cargo correspondiente, en caso de ingresar a la sección de KPI's, se le solicitara una contraseña al usuario para mantener la seguridad de la información que se ingresa por parte de las jefaturas, en caso de ser un supervisor u operador, deberán ingresar directamente a la sección de control de proceso, una vez que los usuarios hayan ingresado a sus respectivos módulos capturaran la información correspondiente a sus puestos o áreas, una vez hayan ingresado la información deberán enviarla a la base de datos, en caso de que el usuario no cuente con una conexión a internet, los datos ingresados serán almacenados de manera local en el dispositivo y podrá enviarlos a la base de datos hasta que la conexión a internet se restaure en la aplicación, una vez que la información ha sido ingresada a las bases de datos, podremos consumir esta información mediante la herramienta de Power BI, desarrollando interfaces para poder mostrar la información mediante un dashboard para que la información pueda ser leída e interpretada por gerencia y superintendentes, con dicha información podrán tomar mejores decisiones que permitan la mejora continua en la empresa.

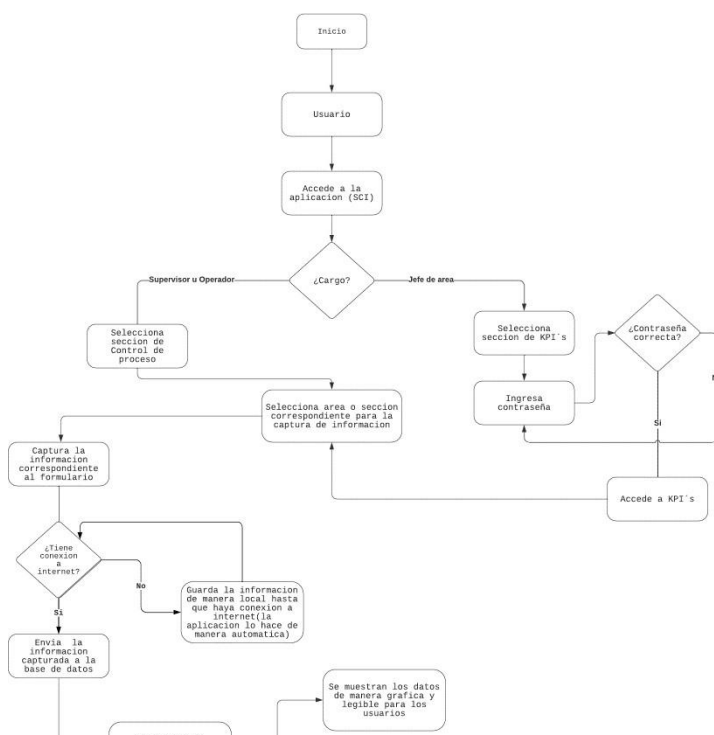


Ilustración 2 Diagrama de funcionamiento del sistema

Diagrama de flujo (Logística operacional del sistema)

En el siguiente diagrama se explica el cómo los usuarios del sistema deberán hacer uso de la aplicación para poder ingresar la información a la base de datos, aquí se hace mención desde que el usuario recibe la información que va ingresar a la aplicación, después si el usuario es un operador o supervisor, deberá situarse en un lugar seguro donde pueda sacar el dispositivo móvil y acceder a la aplicación y detenerse por completo ya que al estar en las líneas de producción caminar con el dispositivo móvil mientras camina e ingresa la información podría causar un accidente, por lo que deberá hacer caso a la medidas de seguridad que se mencionan en el diagrama, ahora, en caso de que el usuario sea parte de una jefatura o su método de ingreso de información sea a través de una computadora, no será necesarios seguir los pasos de seguridad anteriormente mencionados ya que las computadoras se encuentran ya en un sitio seguro para el usuario, posteriormente ingresaran al sistema de como se explica en el diagrama anterior , capturan la información correspondiente y la envían a la base de datos, en caso de no tener conexión a internet, como se comentó anteriormente, la información se almacenara de manera local en el celular hasta que la conexión se restaure y el usuario pueda enviar la información a las bases de datos.

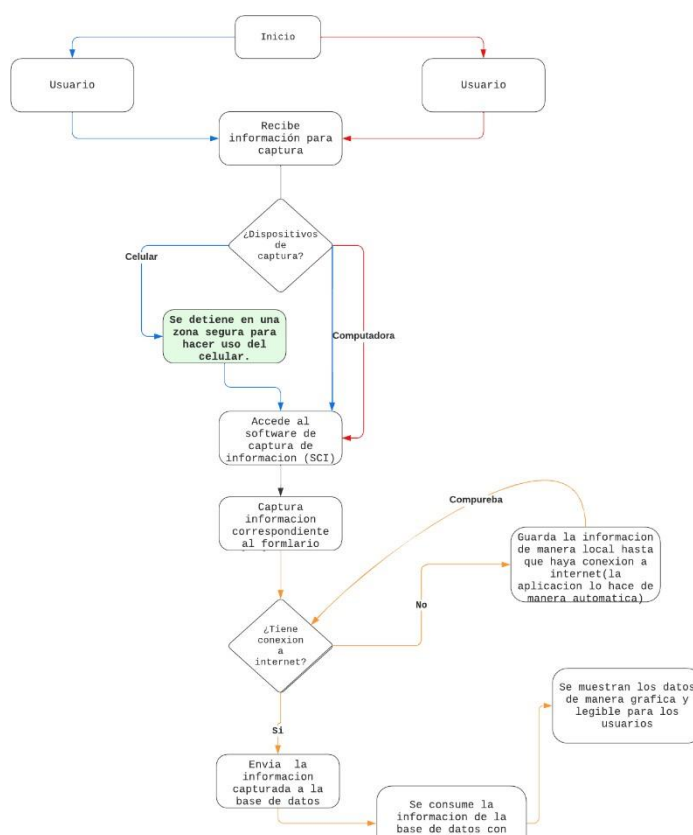


Ilustración 3 Logística operacional del sistema

Diagrama de casos de uso (SCI)

Se muestra el diagrama de casos de uso, para determinar a los usuarios y sus respectivos roles en el sistema de control integral (SCI), en este caso, identificamos a 4 actores en el sistema; Supervisores, jefes de áreas, Operadores y Gerencia, estos son los principales usuarios en el sistema donde supervisores y operadores acceden a control de proceso y los jefes de área acceden a los KPI's, finalmente gerencia es quien visualiza la información de una manera gráfica y legible para el usuarios, permitiéndole conocer el estado día con día de la planta mediante las variables de proceso que se recaban en el sistema.

De igual forma, este diagrama nos permite al momento de realizar la capacitación referente al uso de la aplicación a los usuarios, puedan conocer sus roles dentro del sistema, facilitando la comprensión del uso e ingreso a la aplicación

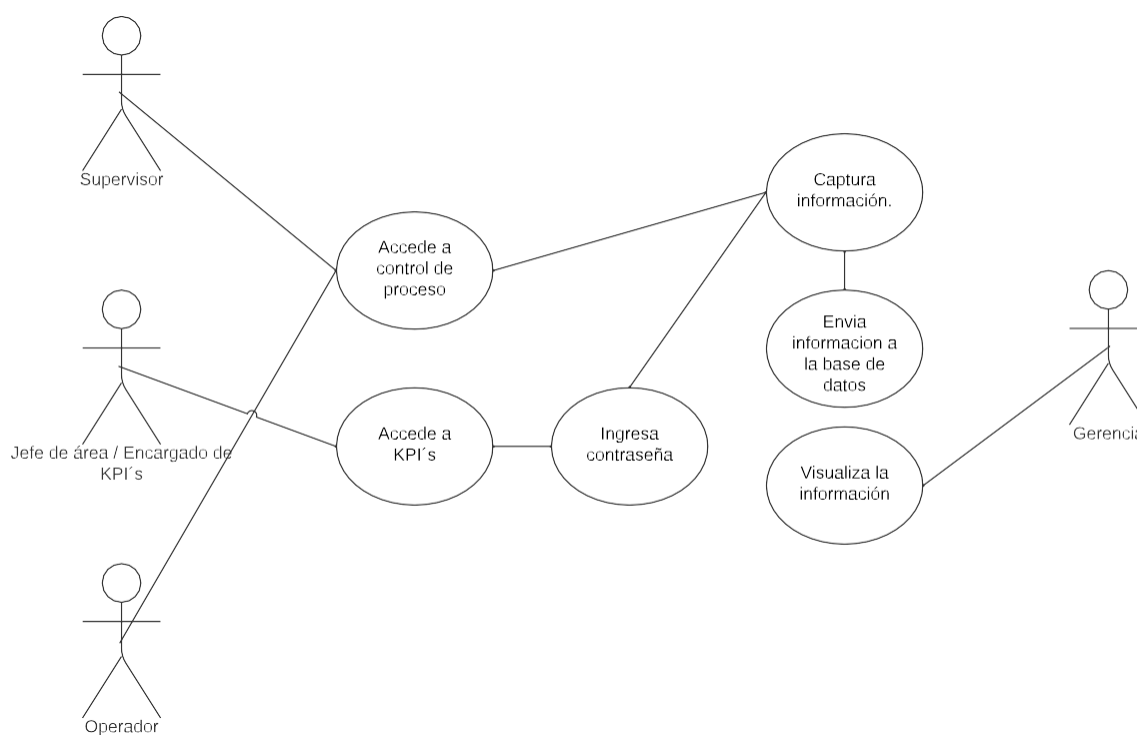


Ilustración 4Casos de uso - SCI

Distribución de indicadores KPI's

En la siguiente imagen, se muestran los indicadores respectivos a KPI's de la empresa junto con sus respectivos responsables, la realización de este diagrama, nos permite establecer los roles y responsabilidades dentro del sistema, estableciendo módulos personalizados en la aplicación para cada responsable de KPI's, de esta forma, cada usuarios puede ingresar a su apartado personalizado las necesidades del usuario para el cumplimiento de la información de KPI's y puedan ingresar información a la base de datos de tal forma que permita lograr una mejor trazabilidad de las variables de proceso del negocio, lo cual permite a gerencia y a los superintendentes verificar que el negocio vaya bien de acuerdo a los objetivos establecidos de cada mes.

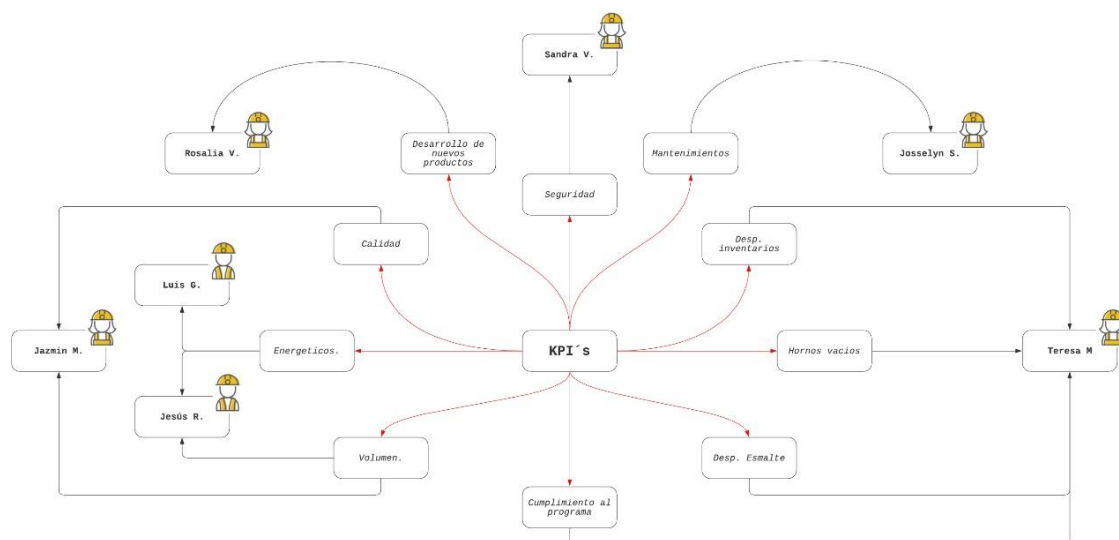


Ilustración 5 Distribución de indicadores KPI's

Diagrama (Vista Física del sistema)

En el siguiente diagrama se muestra el cómo está estructurado el sistema (SCI), donde se muestran los clientes, los cuales hace referencia a los dispositivos que permiten al usuario ingresar a la aplicación, ya sea mediante una computadora o un dispositivo móvil, después se representa el servidor de la aplicación, en dicho servidor es donde se encuentra la aplicación, mostrando la interfaz al usuario, ya sea la interfaz de KPI's o de control de proceso, finalmente se establece la conexión con la base de datos

del sistema, este diagrama nos permite identificar y conocer la manera en que funcionara la conexión entre los clientes, servidor y la base de datos.

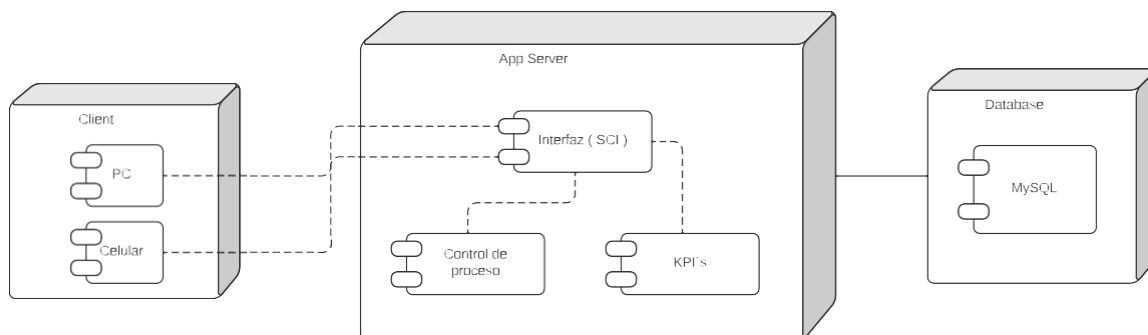


Ilustración 6 Vista física del sistema

Diseño de interfaces (SCI)

En este punto, se inicia la realización y diseños de las interfaces, tanto de la aplicación móvil como las interfaces del dashboard en Power BI, el objetivo es diseñar interfaces que sean amigables e intuitivas para los usuarios de tal forma que les facilite la comprensión y la usabilidad del sistema.

Para la aplicación, se contemplan 5 interfaces principales, las cuales son para el Inicio, Selección de modulo (KPI's o Control de proceso), Selección del área correspondiente, Selección de la subárea y la pantalla de visualización y envío de datos Offline.

La interfaz de inicio permite al usuario acceder a las funcionalidades de la aplicación además de ser la presentación al usuario, por lo que debe ser llamativa y atractiva a la vista, como se muestra en la siguiente imagen, además cuenta con pequeñas animaciones y transiciones al navegar por la aplicación.

La interfaz es intuitiva y ayuda a mejorar la experiencia de usuario en la aplicación desde el inicio.



Ilustración 7 Pantalla de inicio - Aplicación SCI

La interfaz de selección de modulo, de igual forma es minimalista e intuitiva, ya que lleva de la mano al usuario para poder hacer uso de la aplicación sin volverla una herramienta complicada, permitiendo ingresar a dos funcionalidades completamente distintas, además que cuenta con un indicador de conexión, el cual permite al usuario saber si la aplicación tiene una conexión a internet o no.



Ilustración 8 HomeScreen SCI

La siguiente interfaz es la que permite el acceso al módulo de KPI's, protegiéndolo mediante un login, en el cual solo los usuarios con permisos pueden acceder a este módulo para el ingreso de la información correspondiente, garantizando la integridad de la información ante posibles modificaciones no deseadas por terceros, una vez que se ingresa la contraseña permite al usuario ingresar una pantalla de selección, donde mediante el nombre de los responsables, podrán ingresar a sus respectivos indicadores para el ingreso de información como se muestra en la imagen consecuente a la pantalla de inicio de sesión.

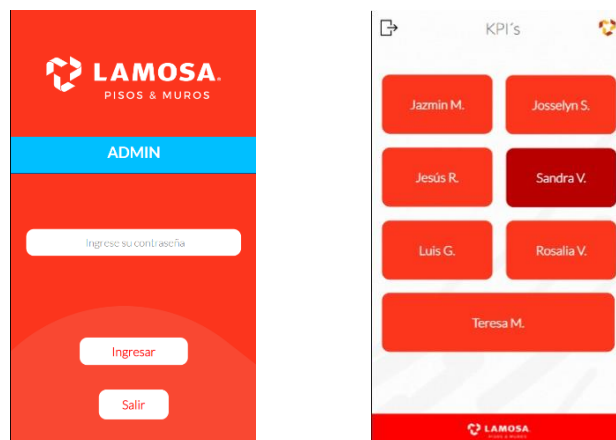


Ilustración 9 Pantallas de login y selección de indicadores KPI's

Para el módulo de control de proceso, encontraremos interfaces más extensas, ya que se incorporaron todas las áreas de la planta, para poder llevar el control de las variables de proceso de la planta.

En la primer imagen, se muestra la interfaz que permite al usuario de control de proceso seleccionar el área correspondiente a su cargo donde vaya a ingresar los datos, en algunos casos, las áreas se encuentran divididas en sub áreas cuando se accede a una, por ejemplo en el área de pastas se divide como se muestra en la imagen consecuente a la primera, y de igual forma en casos especiales, algunas sub áreas también se dividen como en el caso de arcillas, que se divide en inspección de Materia prima y carga de basculas, finalmente se ingresa al formulario donde se podrá capturar la información como se muestra



Ilustración 10 Pantalla de navegación en módulo de CP

En las demás áreas las interfaces son similares a las anterior mostradas, el único cambio son los campos de los formularios y el nombre de las áreas y subáreas, por lo que para el usuario resulta más fácil el acoplarse a la aplicación.

Finalmente encontramos las interfaces que permiten al usuario visualizar y enviar los datos capturados sin una conexión a internet, por lo que es de gran utilidad para las áreas que no tienen acceso a una conexión por las condiciones del área, en esta interfaz se contiene las funciones de enviar datos de manera individual y de manera masiva, además que si el usuario quiere seguir con la captura de datos una vez haya enviado ya cierta cantidad puede hacer pulsando sobre el botón de la esquina superior derecha, el cual redirige al usuario nuevamente al formulario de captura.

El botón de carga masiva de datos le permite enviar todo el conjunto de datos capturados durante el turno y no haya tenido una conexión a internet, por lo que ahora tener una conexión no es indispensable para la captura de información en la aplicación



Ilustración 11 Pantalla de visualización de datos offline

Interfaces del dashboard (Power Bi)

Ahora se muestran las interfaces correspondientes a los dashboard desarrollados en la plataforma de Power BI, las interfaces se dividen en dos, en KPI's y en control de proceso, al igual que en la aplicación, esto con la finalidad de tener un mejor control y orden sobre la información, además que trabajan de manera separada, por lo que si en algún momento, alguno deja de funcionar el otro siempre estará disponible.

En la siguiente imagen se muestra la interfaz de inicio al dashboard para la sección de KPI's, sección en la cual encontraremos las interfaces de volumen, calidad, desperdicio y energéticos, y en cada uno se muestran los valores objetivos y reales, permitiendo observar las tendencias de cada indicador además de lograr una mejora trazabilidad de los datos, para poder filtrar información según la fecha que se requiera y así poder llevar un mejor análisis estadístico de toda la planta



Ilustración 12 Interfaces Power BI - KPI's

Para la sección de control de proceso, se realizaron interfaces por separado al de KPI's, esto para tener un mejor orden de la información, además de dividir la carga de trabajo para los archivos, ya que, al manejar gran cantidad de información, la plataforma de Power Bi tarda en ejecutarse, afectando el rendimiento de los dashboard y también afecta la experiencia de usuario.

En las siguientes imágenes se muestra de manera general las interfaces realizadas para la sección de control de proceso:

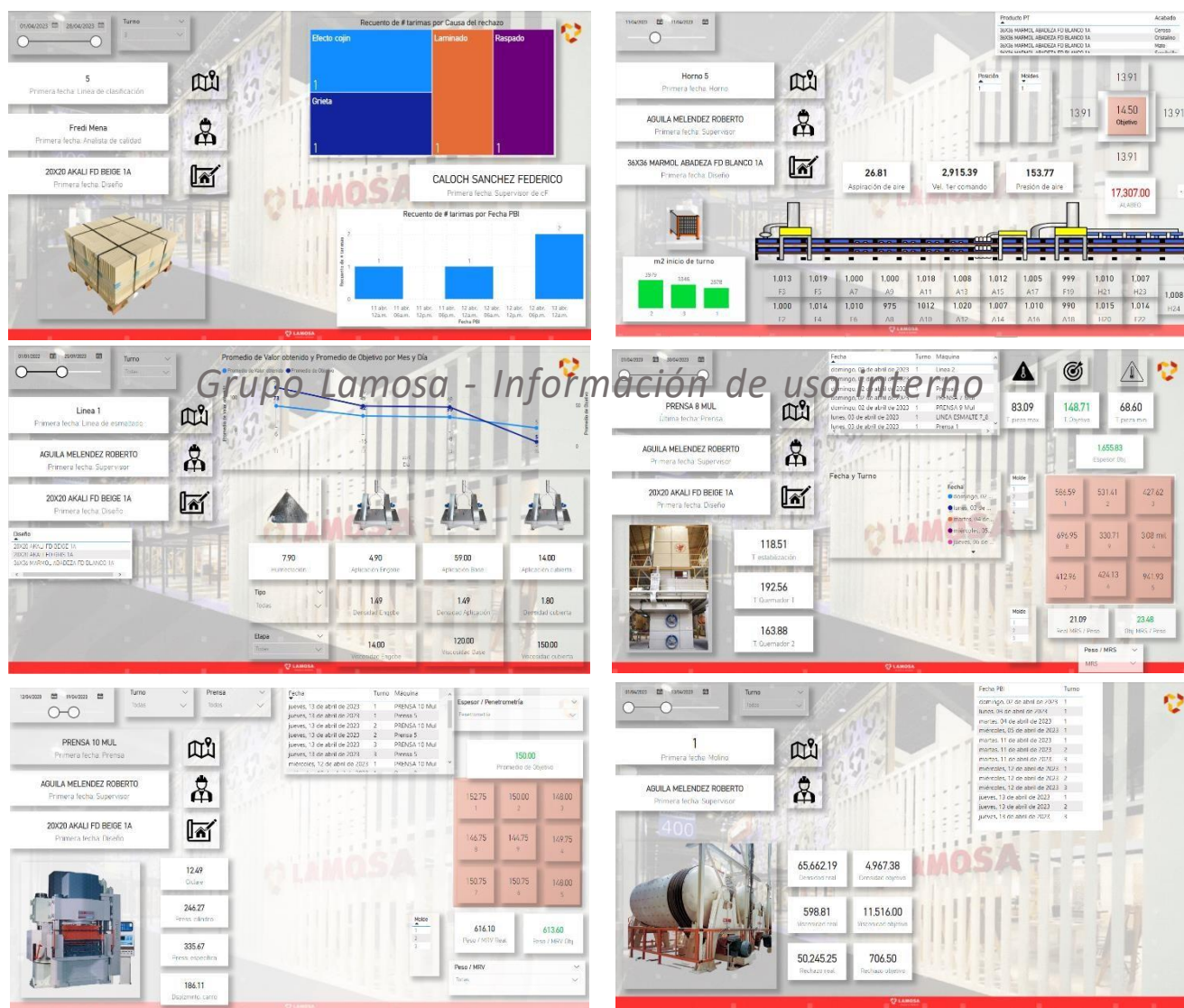


Ilustración 13 Interfaces Power BI - Control de proceso

Las interfaces anteriormente mostradas corresponden a las áreas de producción, midiendo en tiempo real los datos generados en cada línea de producción de manera continua, las 24 horas y los 7 días de la semana, por lo que la información se encuentra disponible para todo el personal, jefaturas gerencia.

Las interfaces fueron diseñadas con el objetivo de ser agradables a la vista y que los datos puedan ser legibles por cualquier persona en la planta, de esta manera nos aseguramos de que se preste total

atención a los resultados diarios y conozcan el estado de la planta, ya sea para saber si los resultados son buenos y mantenerlos o si los resultados son malos tomar acciones para mejorarlos.

Desarrollo del sistema.

Creación de la base de datos.

Para la creación de la base de datos, tomaremos como base el diagrama de base de datos que se mencionó anteriormente, en el apartado de diseño de los diagramas de funcionamiento, iniciamos con la instalación de nuestro gestor de base de datos, que para este caso se realizara con el gestor de MySQL, el gestor nos permitirá realizar la base de datos ya sea mediante líneas de código con lenguaje SQL o mediante ingeniería inversa la cual es una opción que nos brinda el gestor de MySQL.

En este caso aprovecharemos la funcionalidad de DBDiagrama.io, la cual es una herramienta que nos permite realizar el diagrama de la base de datos a través de DBML que usa como sintaxis para el modelado de diagramas.

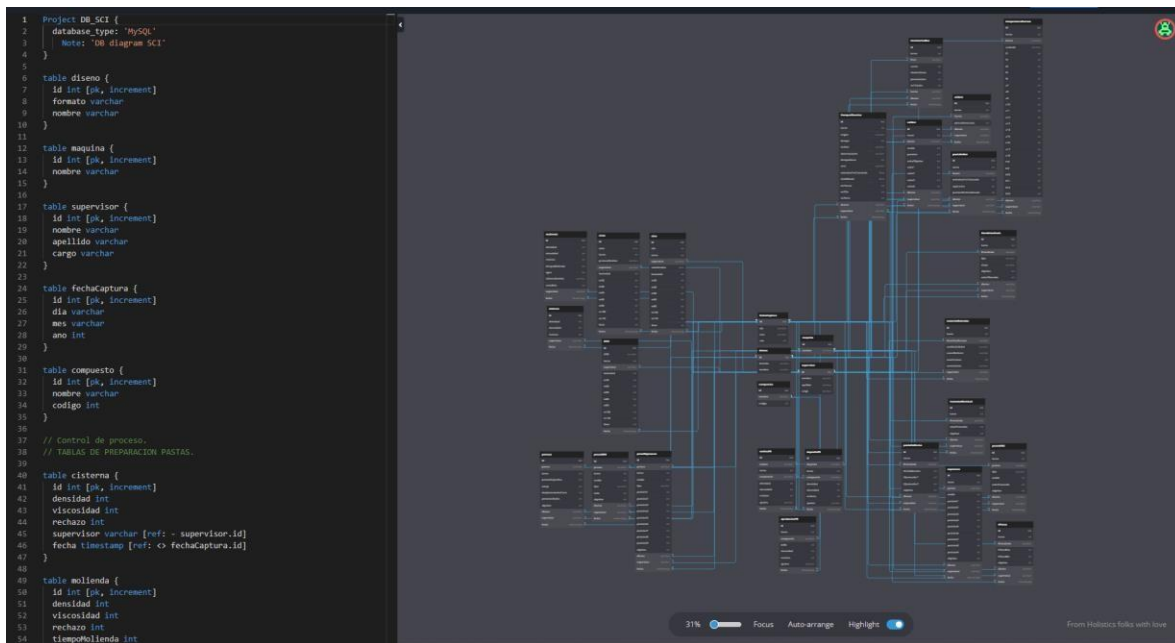


Ilustración 14 Modelado de la base de datos

De esta forma podemos modelar el diagrama de la base de datos a través de DBML, lo cual nos permite crear las relaciones de una forma más rápida y sencilla, además de que una vez que hayamos finalizado el modelado tenemos la opción de convertirlo en lenguaje de SQL, ya sea para MySQL, PostgreSQL, SQL server, lo cual agiliza en gran medida el desarrollo de la base de datos, dando como resultado el siguiente código.

```

CREATE TABLE 'tiemposMuertos' (
);

ALTER TABLE 'cisterna' ADD FOREIGN KEY ('supervisor') REFERENCES 'supervisor' ('id');

CREATE TABLE 'fechaCaptura_cisterna' (
    'fechaCaptura_id' int,
    'cisterna_fecha' timestamp,
    PRIMARY KEY ('fechaCaptura_id', 'cisterna_fecha')
);

ALTER TABLE 'fechaCaptura_cisterna' ADD FOREIGN KEY ('fechaCaptura_id') REFERENCES 'fechaCaptura' ('id');
ALTER TABLE 'fechaCaptura_cisterna' ADD FOREIGN KEY ('cisterna_fecha') REFERENCES 'cisterna' ('fecha');

ALTER TABLE 'molienda' ADD FOREIGN KEY ('supervisor') REFERENCES 'supervisor' ('id');
CREATE TABLE 'fechaCaptura_molienda' (
    'fechaCaptura_id' int,
    'molienda_fecha' timestamp,
    PRIMARY KEY ('fechaCaptura_id', 'molienda_fecha')
);

ALTER TABLE 'fechaCaptura_molienda' ADD FOREIGN KEY ('fechaCaptura_id') REFERENCES 'fechaCaptura' ('id');
ALTER TABLE 'fechaCaptura_molienda' ADD FOREIGN KEY ('molienda_fecha') REFERENCES 'molienda' ('fecha');

ALTER TABLE 'ATM' ADD FOREIGN KEY ('supervisor') REFERENCES 'supervisor' ('id');
CREATE TABLE 'fechaCaptura_ATM' (
    'fechaCaptura_id' int,
    'ATM_fecha' timestamp,
    PRIMARY KEY ('fechaCaptura_id', 'ATM_fecha')
);

ALTER TABLE 'fechaCaptura_ATM' ADD FOREIGN KEY ('fechaCaptura_id') REFERENCES 'fechaCaptura' ('id');
ALTER TABLE 'fechaCaptura_ATM' ADD FOREIGN KEY ('ATM_fecha') REFERENCES 'ATM' ('fecha');

ALTER TABLE 'silos' ADD FOREIGN KEY ('supervisor') REFERENCES 'supervisor' ('id');
CREATE TABLE 'fechaCaptura_silos' (
    'fechaCaptura_id' int,
    'silos_fecha' timestamp,
    PRIMARY KEY ('fechaCaptura_id', 'silos_fecha')
);

ALTER TABLE 'fechaCaptura_silos' ADD FOREIGN KEY ('fechaCaptura_id') REFERENCES 'fechaCaptura' ('id');

CREATE TABLE 'maquina' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'nombre' varchar(255)
);

CREATE TABLE 'supervisor' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'nombre' varchar(255),
    'apellido' varchar(255),
    'cargo' varchar(255)
);

CREATE TABLE 'fechaCaptura' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'dia' varchar(255),
    'mes' varchar(255),
    'ano' int
);

CREATE TABLE 'compuesto' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'nombre' varchar(255),
    'codigo' int
);

CREATE TABLE 'cisterna' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'densidad' int,
    'viscosidad' int,
    'rechazo' int,
    'supervisor' varchar(255),
    'fecha' timestamp
);

CREATE TABLE 'molienda' (
    'id' int PRIMARY KEY AUTO_INCREMENT,
    'densidad' int,
    'viscosidad' int,
    'rechazo' int,
    'tiempoMolienda' int,
    'agua' int,
    'cisternaDestino' varchar(255),
    'vacuola' int,
    'supervisor' varchar(255),
    'fecha' timestamp
);

```

Ilustración 15 Archivo SQL de la base de datos.

Las imágenes anteriores muestran el código de manera general, ya que el archivo de la base de datos es realmente extenso como para ponerlo completo, pero con el simple hecho de que, con la realización del modelado, podremos obtener el código es una gran optimización en el tiempo de desarrollo.

Desarrollo del frontend

Para el desarrollo del frontend se usó la plataforma de Power apps, la cual es una herramienta low code, permitiendo el desarrollo de aplicación de forma rápida y sin tantas líneas de código, aunque para algunas funcionalidades es necesarios personalizarlo mediante código.

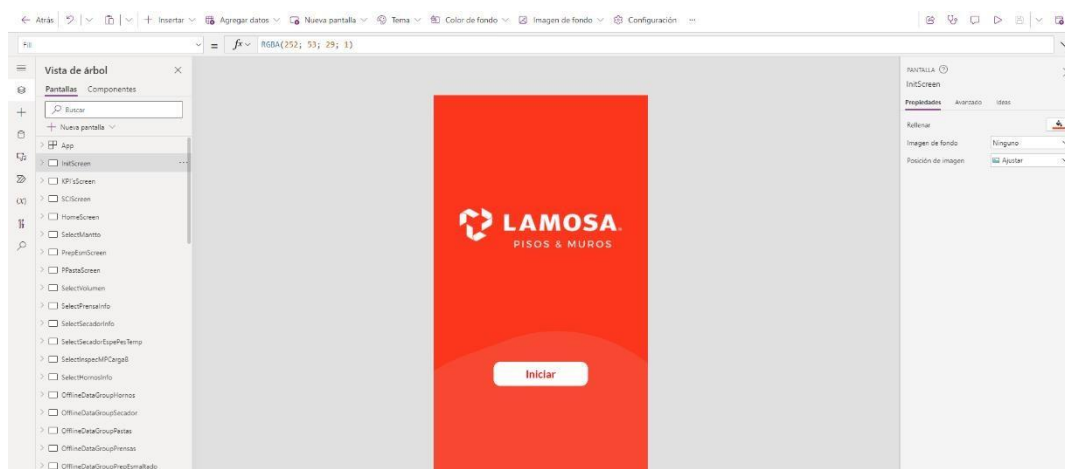


Ilustración 16 proceso de desarrollo de la aplicación

Como se muestra en la imagen anterior, así es como es el entorno de desarrollo de Power apps, esta aplicación fue desarrollada desde un lienzo en blanco, por lo que cada una de sus partes es totalmente personalizada, como ya se comentó en el apartado de diseño de interfaces, aquí se implementaron los diseños mostrados, de esta manera le damos interactividad a los diseños para obtener una aplicación completa y robusta para el uso en producción y obtener los resultados esperados.

El desarrollo de la aplicación cuenta con 86 pantallas en las que el usuario puede navegar según su rol correspondiente, cada pantalla cuenta con una personalización tanto en diseño como en funcionalidades, ya que se hizo a medida de los usuarios y áreas, debido a que así se solicitó en el levantamiento de requerimiento que se llevó al inicio del proyecto.

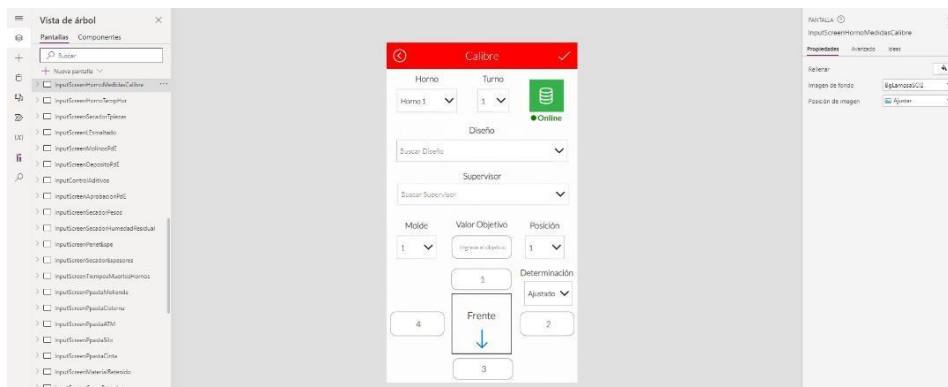


Ilustración 17 Desarrollo de formularios en Power apps

Documentación SCI | Power Apps

En este documento, se explica el proceso de desarrollo de la aplicación SCI en Power Apps, abarcando puntos como el desarrollo de interfaces, integración de nuevas variables, adaptación de funcionalidades (Online/Offline), creación de colecciones de datos, explicación del proceso lógico del código implementado para ciertas funcionalidades, etc. Este documento tiene como objetivo informar y dar contexto sobre el proceso de desarrollo a nuevos participantes del proyecto para realizar futuras actualizaciones o dar mantenimiento a la aplicación.

Power apps, ofrece dos formas principales para el desarrollo de aplicaciones, la primera es el desarrollo basado en modelos, es decir que, para la creación de una aplicación basada en modelos, deberemos indicar a Power apps el origen de datos que usaremos en la aplicación, posterior a eso Power apps nos creara automáticamente una aplicación básica con métodos de inserción y visualización de datos. Para un desarrollo más personalizado y que se ajuste a las necesidades del proyecto, tenemos la opción de desarrollar la aplicación desde un lienzo en blanco, esto nos da la posibilidad de diseñar las interfaces propias de la aplicación.

La aplicación de SCI (Sistema de Control Integral) se desarrolló desde un lienzo, donde se diseñaron e implementaron las interfaces.

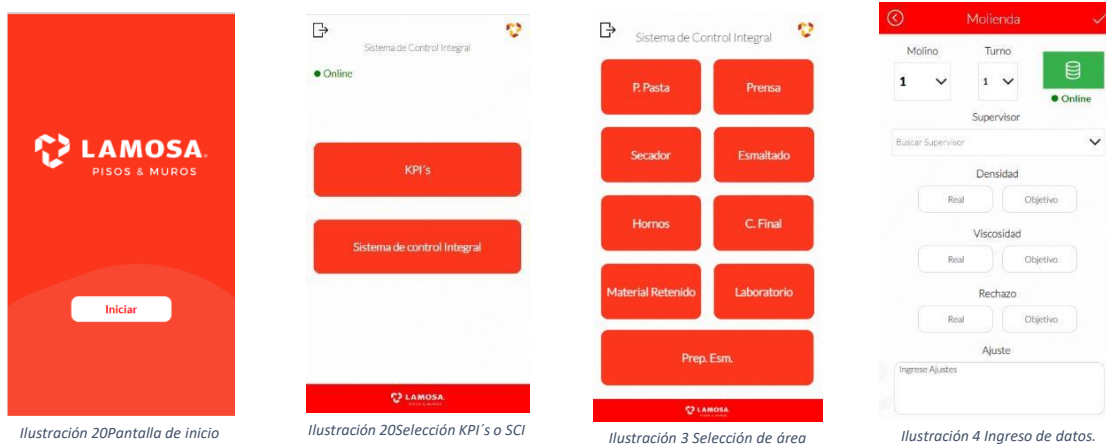


Ilustración 18 Demostración de interfaces en Power apps

Edición de la aplicación (SCI)

Para poder editar la aplicación en Power apps, ingresaremos a la herramienta de Power apps y daremos clic a la pestaña de aplicaciones.

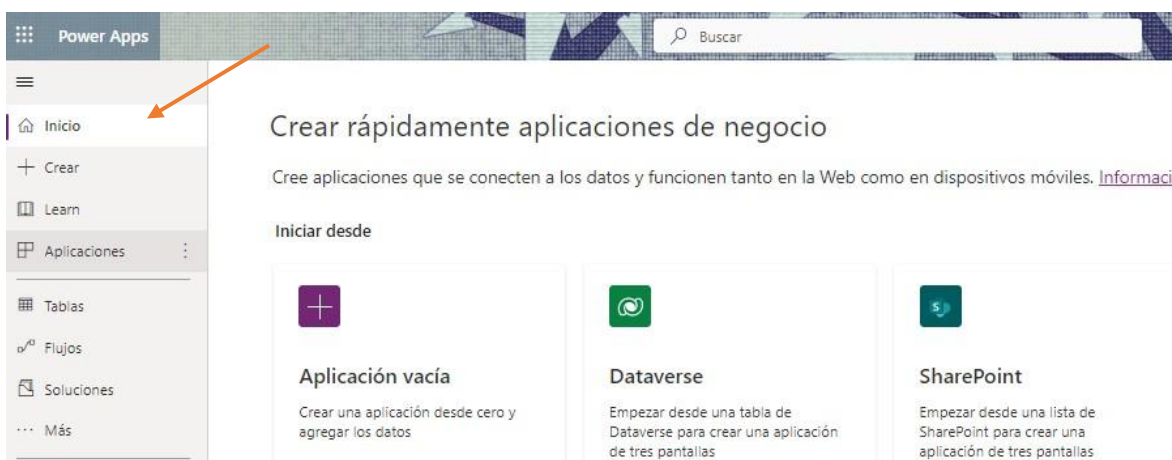


Ilustración 21 Edición de aplicación en Power apps

Posteriormente, en la sección de aplicaciones buscaremos la aplicación, en este caso la aplicación tiene el siguiente nombre: “SCI – V1”, cuando la hayamos identificado, deberemos dar clic en los tres puntos que aparecen delante del nombre, a lo cual nos desplegará un menú con distintas opciones, a lo cual deberemos dar clic a la opción de editar como se muestra en la siguiente imagen y comenzará a cargar la aplicación para su edición.

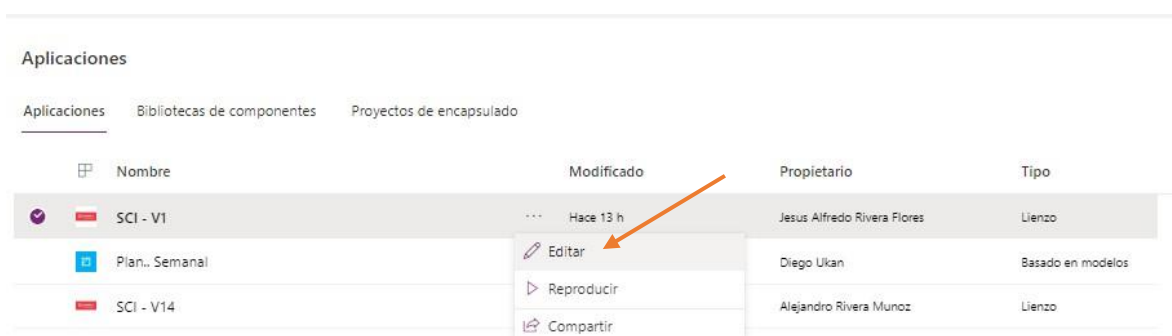


Ilustración 22 Selección de la aplicación a editar

Cuando cargue la aplicación para poder realizar la edición correspondiente, nos aparecerá la siguiente interfaz, donde podremos encontrar distintas herramientas que podremos usar para crear nuevas funciones en la aplicación, en la flecha de color azul encontraremos las secciones de herramientas, para insertar nuevas pantallas, agregar inputs, botones, etiquetas, galerías, etc., también encontraremos la sección para agregar orígenes de datos, estas opciones mencionadas, se explicaran detalladamente más adelante; en la flecha de color naranja, encontraremos el menú que despliega cada sección de herramientas de la flecha azul y en la flecha de color amarillo, encontraremos el menú de propiedades de cada componente que tengamos seleccionado, donde podremos modificar colores, tamaños, tipos de letra, ubicación, etc.

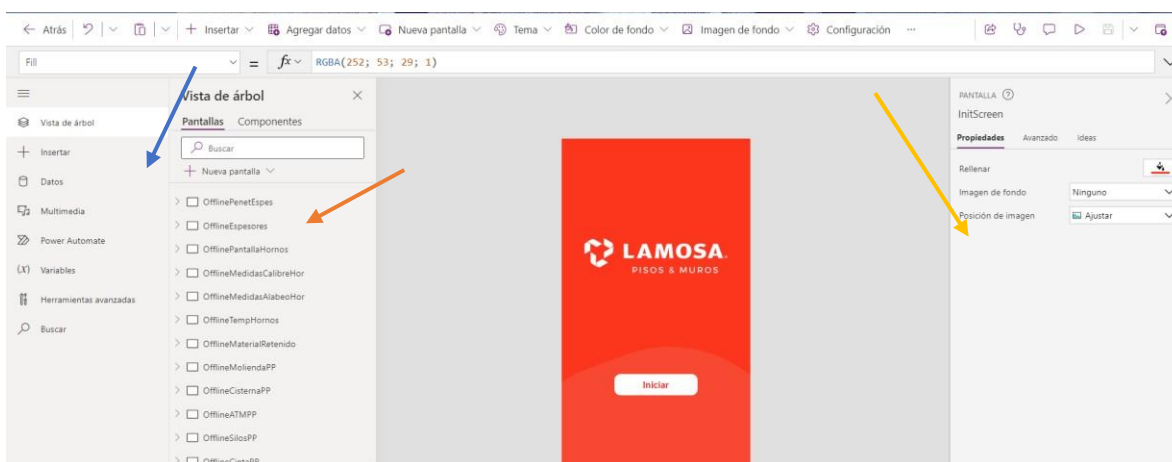


Ilustración 23 Interfaz de edición en Power apps

Agregar nuevas pantallas

Para agregar nuevas pantallas tenemos dos formas de hacerlo, desde una pantalla en blanco o desde una plantilla que proporcionar Power apps, una forma adicional para agregar una pantalla es que si ya tenemos pantallas creadas podemos duplicar una que ya haya sido creada, de esta forma podremos mantener el mismo estilo entre pantallas, además de que será más fácil la edición e integración de funcionalidades, ya que se duplican con la misma configuración, es decir que si en una pantalla ya creada con funcionalidades que queremos replicar en otra distinta, podremos duplicarla sin perder las funcionalidades que ya se tenían, esto nos permite agilizar el proceso de desarrollo.

En la siguiente imagen, podemos observar el cómo se añaden nuevas pantallas como se comentó al inicio, para esto tendremos que seleccionar la sección de “Vista de árbol” y seleccionar la opción de “Nueva pantalla” a lo que nos dará la opción de iniciar en blanco o desde una plantilla, como se comentó anteriormente

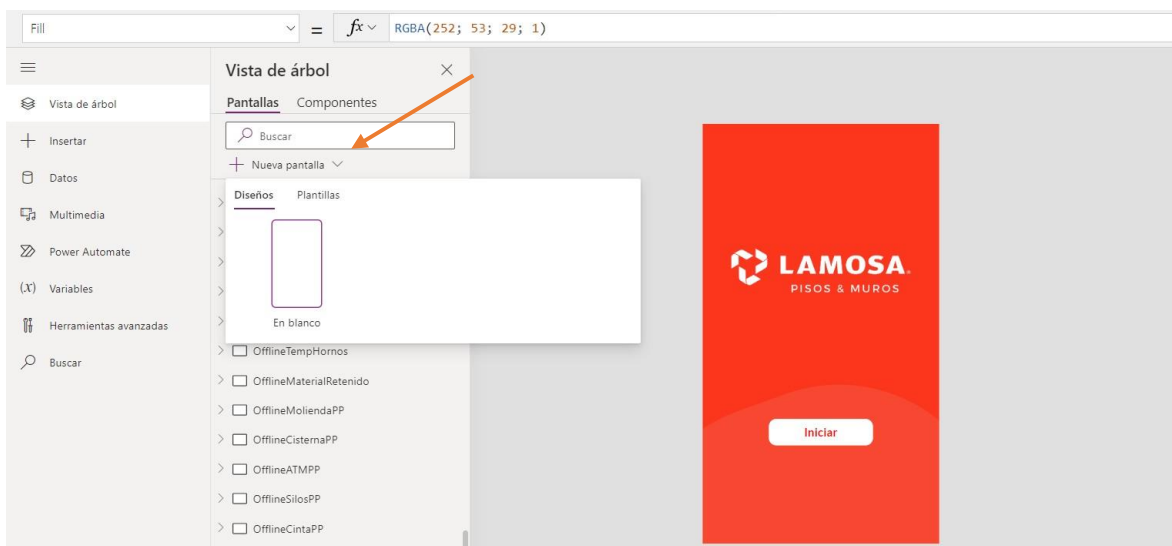


Ilustración 24 Creación de nuevas pantallas

Para duplicar una pantalla existente, deberemos seleccionar una pantalla con las características que deseemos duplicar para solo editar y adaptar los componentes a la nueva pantalla, para esto nos situaremos en la sección de “Vista de árbol”, en esta sección nos aparecerán todas las pantallas de la aplicación, deberemos situarnos e identificar la pantalla que queremos duplicar, la seleccionamos y nos aparecerá tres puntos delante del nombre de la aplicación, cuando demos clic en los tres puntos, nos desplegará un menú con las opciones que se muestran en la siguiente imagen, las opciones que más se usar son: Duplicar pantalla, Eliminar y Renombrar pantalla. Es importante nombrar o renombrar las pantallas cada que agreguemos una nueva con un nombre con el cual podamos identificarlas fácilmente, ya que al agregar más pantallas continuamente estas incrementan y se complica llevar un control de cada una, por lo que deberemos colocar nombres relacionados a la pantalla.

Una vez que tengamos duplicada la pantalla y con un nombre identificable, podemos continuar con la modificación la vista y funcionalidades.

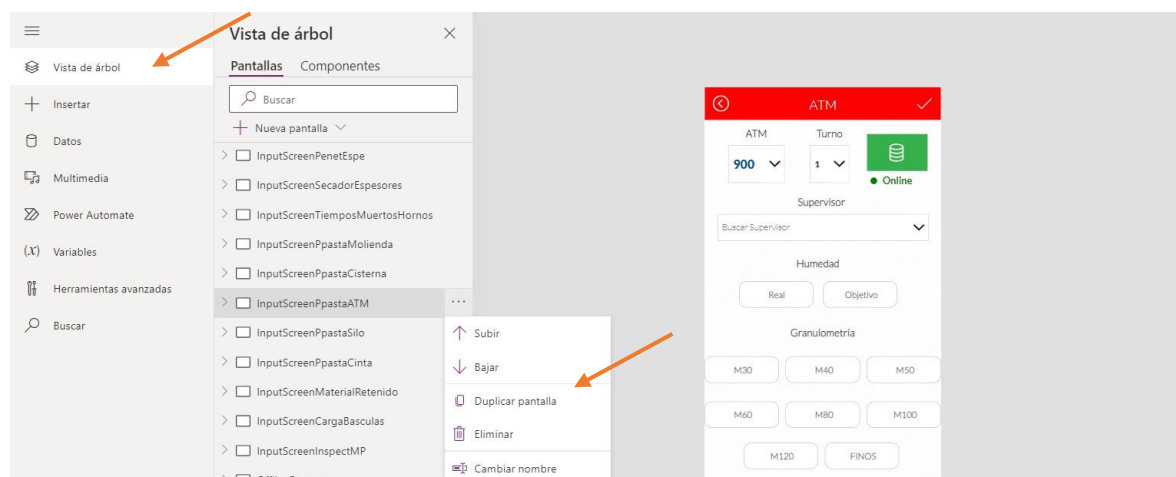


Ilustración 25 Renombramiento y duplicidad de pantallas

Navegación entre pantallas

Para realizar la navegación entre pantallas, deberemos insertar un botón, lo seleccionamos e ingresamos a su propiedad “OnSelect” y en el editor de código pondremos la instrucción que se muestra en la imagen, dicha instrucción lleva la siguiente estructura: `Navigate (<Nombre de la pantalla>; “ScreenTransition. <Nombre de la transición>”)`

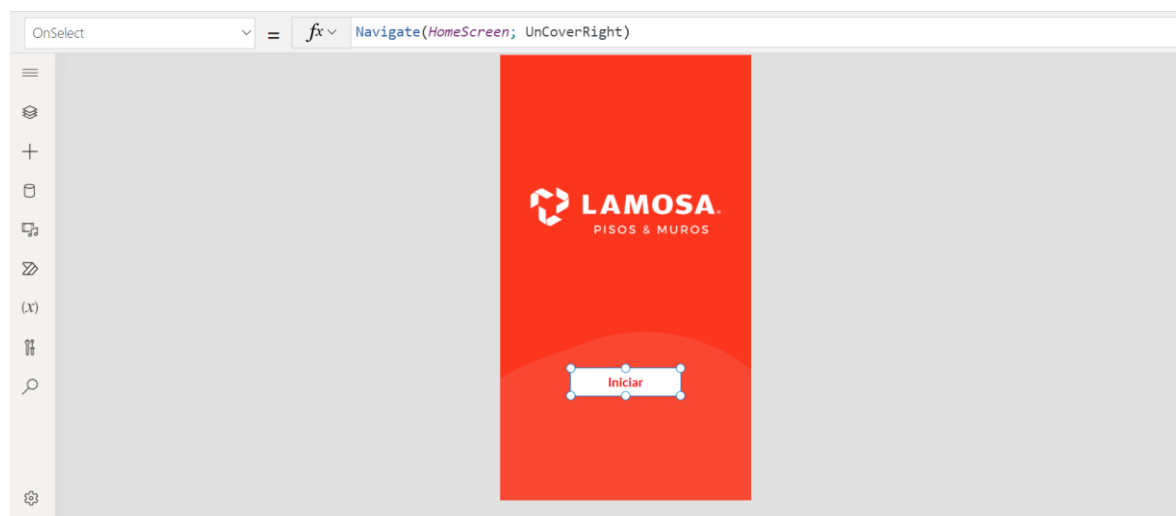


Ilustración 26 Formula para navegación entre pantallas.

Agregar componentes (inputs, etiquetas, galerías, etc.)

Para poder agregar componentes deberemos situarnos en la sección de “Insertar”, al dar clic nos mostrara todos los componentes que Power apps nos permite usar para el desarrollo de una aplicación, los componentes se colocan por clasificaciones, como componentes de entrada, visualización, diseño, multimedia, etc. Son todos los que están señalados con la flecha color azul, cada que demos clic en cada clasificación, se nos mostraran los componentes respectivos a la clasificación seleccionada como se muestra en la siguiente imagen.

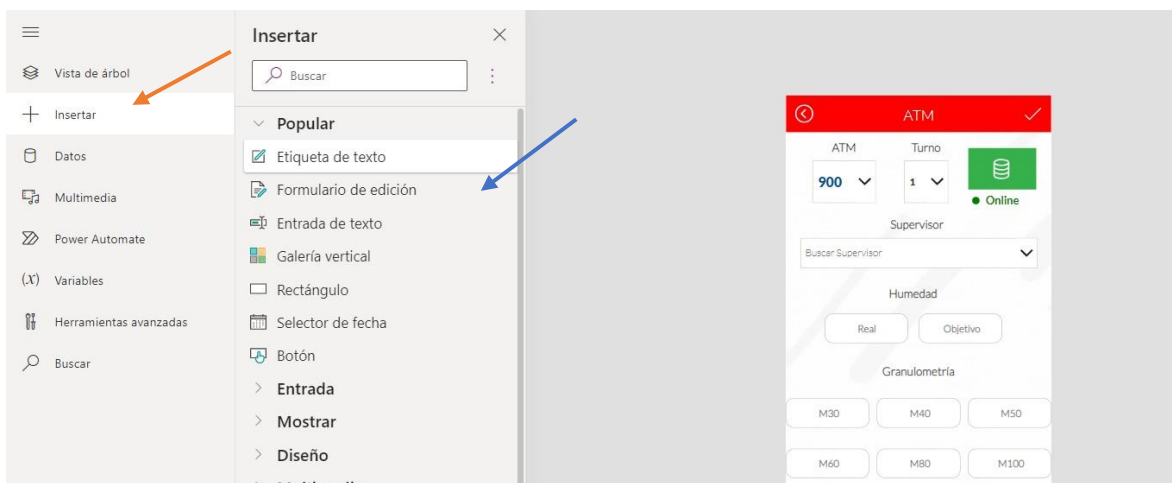


Ilustración 27 Componentes en Power apps

Para agregar componentes al lienzo de la aplicación es muy sencillo ya que solo deberemos seleccionar el componente que deseamos usar y lo arrastraremos al lienzo y se colocara en automático, es importante que al igual que la pantalla, los componentes como inputs, galerías, etc., sean nombrados con nombres que nos permitan identificarlos con facilidad y poder realizar interacciones entre componentes, para nombrar los componentes deberemos seleccionar el componente e ir al menú de propiedades del componente, ahí podremos cambiar el nombre, color, letra, tamaño, posición, etc.

Esto aplica para la mayoría de los componentes básicos usados en esta aplicación (SCI)

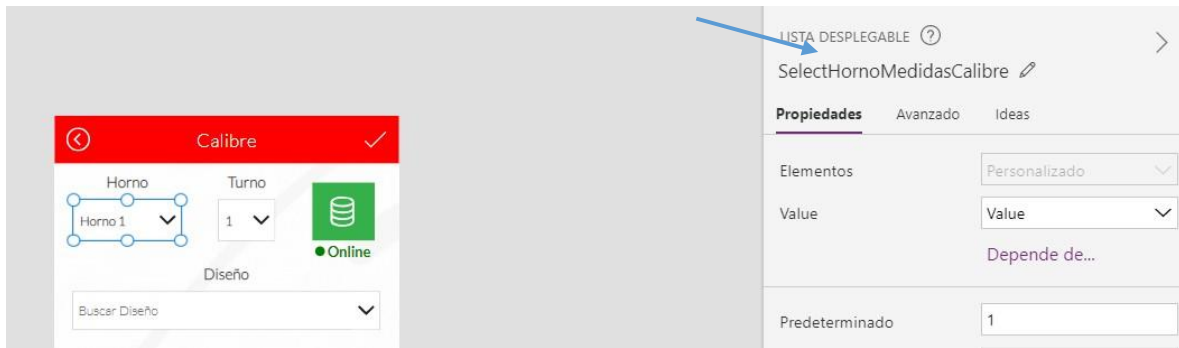


Ilustración 28 Componentes y como identificarlos para uso en formulas.

Listas desplegables.

El uso de listas desplegables permite dar opciones de elección a los usuarios al momento de ingresar información, por lo que para agregar estas listas deberemos ir a la sección de componentes que ya hemos visto y en el buscador podemos escribir “Lista” y nos aparecerán las posibles coincidencias a ese componente, deberemos seleccionar “Lista desplegable” y la podremos colocar en el lienzo.

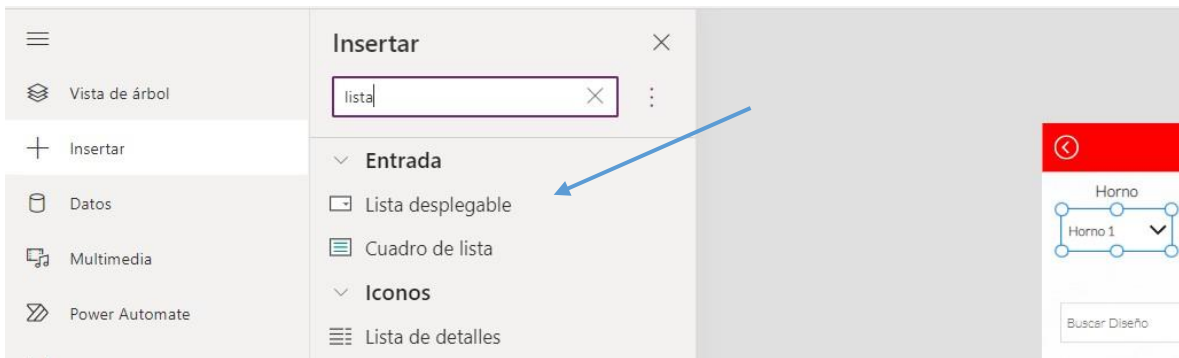


Ilustración 29 Uso de listas desplegables.

Ahora tenemos nuestra lista, pero ¿Cómo agregamos valores u opciones a la lista?, es muy sencillo agregar datos a una lista, para esto debes saber que hay dos opciones de hacerlo, la primera es mediante un origen de datos (Bases de datos, colecciones) y la otra sería ingresar los datos de manera manual, dependiendo los casos es conveniente de una u otra forma, por ejemplo; si es una lista que se repetirá constantemente los datos en otras pantallas, lo mejor es usarlas con orígenes de datos, ya que esto permite modificar o agregar datos en el origen y esto hará que se modifique en las listas que hagan uso de dicho origen de datos, por lo que resumidamente solo deberemos modificar datos en un solo lugar, ahora de forma manual podemos usarla si solo lo usaremos en una sola pantalla, ya que para agregar o modificar datos no será algo complicado ya que no se usa en varias pantallas o son pocos datos que podemos ingresar de forma manual.

Agregar datos manualmente a listas desplegables.

Para agregarlos de manera manual deberemos seleccionar la lista a la que vamos a agregar datos y escribiremos los datos en la barra de fórmulas de Power apps con el siguiente formato:

```
[  
    "Horno 1";  
    "Horno 2";  
    "Horno 3 INF";  
    "Horno 3 SUP";  
    "Horno 4";  
    "Horno 5"  
]
```

Ilustración 30 Forma manual para ingreso de datos a listas

Agregar datos mediante orígenes de datos.

Para agregar datos mediante orígenes de datos, solo deberemos colocar el nombre de la base de datos o la colección en la barra de fórmulas de la lista seleccionada y Power apps mostrara los datos en la lista.

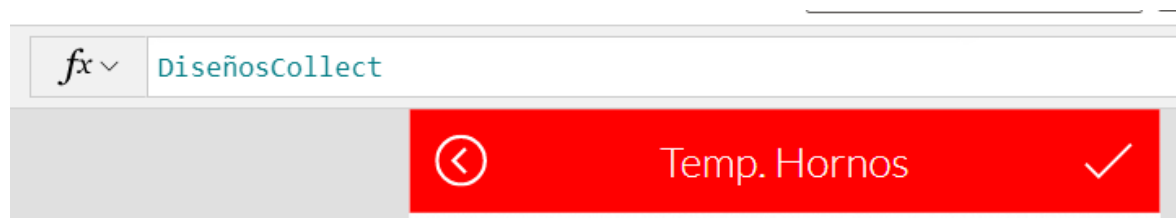


Ilustración 31 Agregar datos a listas mediante orígenes de datos o colecciones

Recomendación: Es importante mencionar que las listas solo sean usadas cuando se mostraran pocos datos al usuario (10 opciones máx.), ya que, para mostrar grandes cantidades de datos al usuario, tenemos otras opciones mejores como "ComboBox" el cual nos permite pasarle un origen de datos y poder buscar dichos datos mediante nombre, iniciales, etc. Es decir que el usuario podrá buscar datos específicos en una base de datos o colección de Power apps

Ejemplo: En la aplicación de SCI se usan listas para mostrar opciones como turno, maquinas, fases, entre otras porque solo son pocos datos, pero para dar a seleccionar al usuario los diseños, supervisores, operadores, compuestos es mejor usar combo box ya que como se menciona tiene funciones de buscador y permite al usuario seleccionar información específica de un conjunto grande de datos como se muestra en la siguiente imagen:

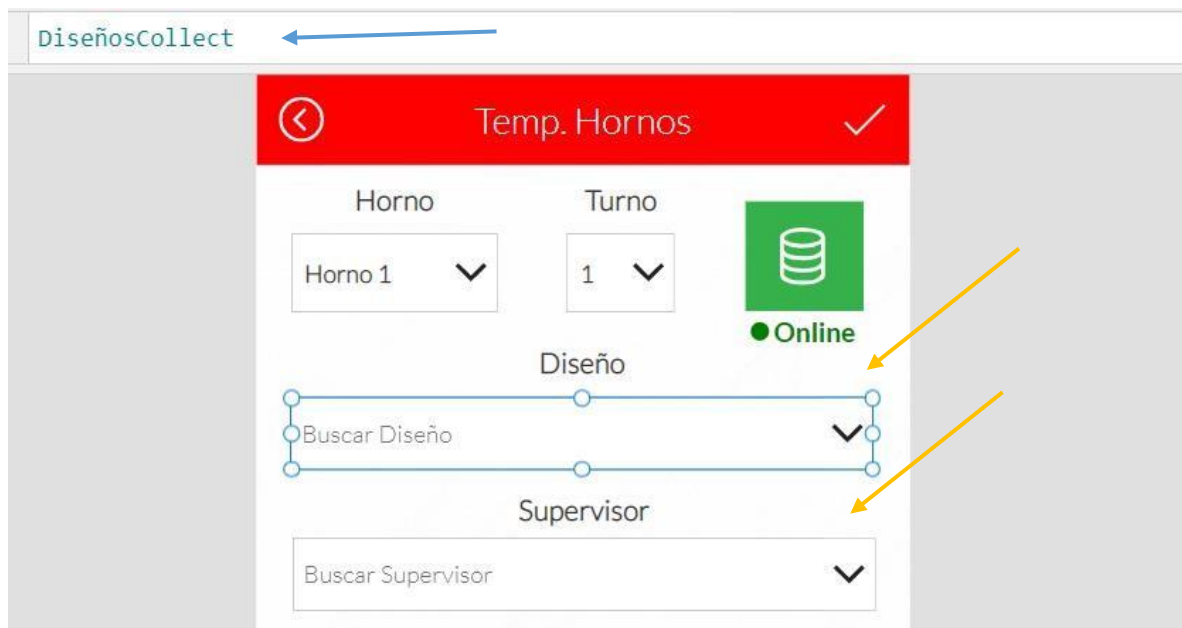


Ilustración 32 Demostración de componente Combo Box con colecciones

Envío de información a bases de datos.

Para el ingreso de información a la base de datos, deberemos situarnos en los formularios de la aplicación, primero deberemos insertar los inputs referentes a los que se quiere almacenar en la base de datos, para esto deberemos establecer si el formulario contendrá inputs, listas, combo box, etc., esto para saber cómo el usuario ingresará la información, ya construido el formulario quedaría algo similar a la siguiente imagen:

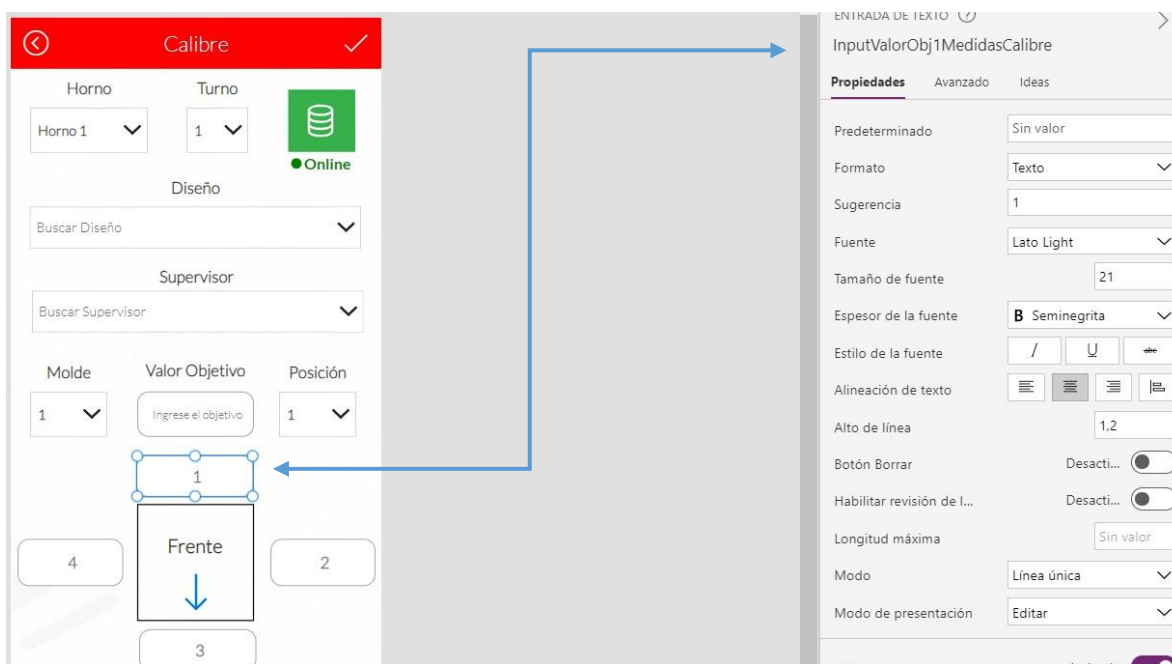


Ilustración 33 Identificador de componente para uso en el código de envío de datos

Una vez que tenemos los inputs identificados con un nombre referente a lo que se guardara en la base de datos como se muestra con la flecha color azul, deberemos situarnos en el icono que se encuentra en la esquina superior derecha, el cual es un check que permite el envío de la información a la base de datos, pero antes de eso deberemos escribir unas líneas de código para poder hacer el envío a la base de datos. Para hacer el envío seleccionaremos el icono no accederemos a su propiedad OnSelect.

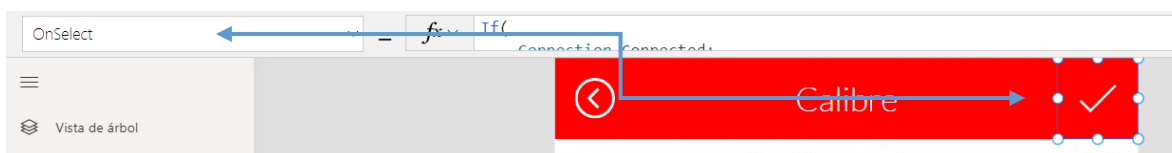


Ilustración 34 Propiedad OnSelect de componentes

Escribiremos las siguientes líneas de código para enviar los datos, cabe mencionar que para futuras implementaciones o nuevos formularios se puede copiar el código y adaptarlo a el nuevo formulario, ya que él envió sigue la misma lógica en la mayoría de los formularios.

La lógica del código es la siguiente: Si la aplicación está conectada a internet, mandaremos la información mediante un “patch” pasándolo como argumento la tabla o base de datos a donde vamos a enviar la información, posterior a eso con la función defaults, pondremos los nombres de las columnas de la base de datos y delante pondremos el valor que guardará, en este caso ponemos el nombre que le dimos al input que tendrá la información.

```
If(
    Connection.Connected;
    Patch(
        CalibreHornos;
        Defaults(CalibreHornos);
        {
            'Fecha ': Text(
                Now();
                "mm/dd/yyyy hh:mm:ss"
            );
            Turno: SelectTurnoHornosMedidasCalibre.SelectedText;
            Horno: SelectHornoMedidasCalibre.SelectedText;
            Diseño: SelectDiseñoHornoMedidasCalibre.Selected;
            Supervisor: SelectSuperHornoMedidasCalibre.Selected;
            'Molde ': SelectMoldeMedidasCalibre.SelectedText;
            Posición: SelectPosicionMedidasCalibre.SelectedText;
            'Valor objetivo': InputValorObjCalibre.Text;
            '1': InputValorObj1MedidasCalibre.Text;
            '2': InputValorObj2MedidasCalibre.Text;
            '3': InputValorObj3MedidasCalibre.Text;
            '4': InputValorObj4MedidasCalibre.Text
        }
    );
);
```

Ilustración 35 Código para envío de información a base de datos.

En la siguiente imagen, creamos una colección, esta colección se crea con la finalidad de almacenar la información en caso de que no haya una conexión a internet, por lo que creamos la colección con los mismos campos que arriba, ya que si no se cumple la condición de que haya internet, enviara los datos a la colección creada para esos datos.

```

Collect(
    DatosCalibreHorSinCargar;
    {
        'Fecha ': Text(
            Now();
            "mm/dd/yyyy hh:mm:ss"
        );
        Turno: SelectTurnoHornosMedidasCalibre.SelectedText;
        Horno: SelectHornoMedidasCalibre.SelectedText;
        Diseño: SelectDiseñoHornoMedidasCalibre.Selected;
        Supervisor: SelectSuperHornoMedidasCalibre.Selected;
        'Molde ': SelectMoldeMedidasCalibre.SelectedText;
        Posición: SelectPosicionMedidasCalibre.SelectedText;
        'Valor objetivo': InputValorObjCalibre.Text;
        '1': InputValorObj1MedidasCalibre.Text;
        '2': InputValorObj2MedidasCalibre.Text;
        '3': InputValorObj3MedidasCalibre.Text;
        '4': InputValorObj4MedidasCalibre.Text
    }
);
SaveData(
    DatosCalibreHorSinCargar;
    "DatosCalibreHorParaCargar"
);

```

Ilustración 36 Creación de colección para datos offline

Finalmente, la última parte del código es la que se encarga de resetear los inputs que indiquemos para que se elimine la información una vez que se haya enviado la información a la base de datos y notifica que los datos se enviaron correctamente.

```

SaveData(
    DatosCalibreHorSinCargar;
    "DatosCalibreHorParaCargar"
);
Reset(InputValorObjCalibre);
Reset(InputValorObj1MedidasCalibre);
Reset(InputValorObj2MedidasCalibre);
Reset(InputValorObj3MedidasCalibre);
Reset(InputValorObj4MedidasCalibre);
Notify(
    "Datos enviados correctamente";
    Success;
    6000
);

```

Ilustración 37 Reset de componentes

Colecciones.

Como vimos anteriormente, hicimos uso de colecciones para guardar datos capturados sin una conexión a internet, las colecciones nos permiten almacenar información temporal en la aplicación, es decir que se almacena en la memoria del dispositivo hasta que la información sea enviada.

Para que la aplicación pueda cargar las colecciones que creamos para almacenar la información sin conexión a internet y también colecciones que usamos para guardar información interna de la aplicación deberemos colocar ciertas líneas de código cada que creamos una colección en la aplicación de esta forma, cuando inicia la aplicación también se cargan las colecciones para trabajar con ellas.

Para acceder al inicio de la aplicación deberemos seleccionar la sección de APP en Power apps y acceder a la propiedad OnStart

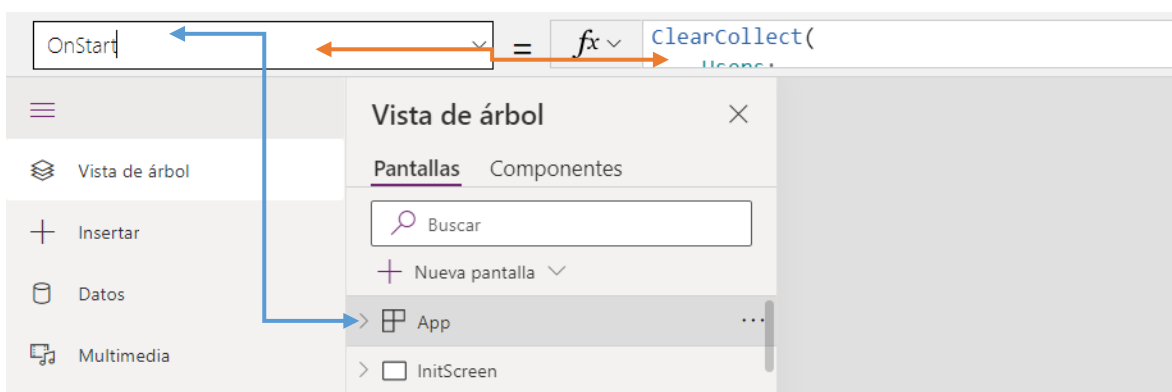


Ilustración 38 Propiedad OnStart del archivo de inicio de la aplicación para carga de información.

En la barra de fórmulas deberemos establecer las colecciones de la siguiente manera, de esta forma cuando se inicie la aplicación los datos de las colecciones se cargarán igualmente, cabe recalcar que anteriormente se muestra el cómo se crean las colecciones mediante la función “Collect” y “ClearCollect” en la parte de envío de los datos, por lo que en esta sección solo tendremos que poner la función “LoadData” y pasarle como argumentos la colección creada y el nombre del archivo donde se guardara la información y quedaría como se muestra en la siguiente imagen.

```
ClearCollect(
    Users;
    {
        Nombre: "ADMIN";
        Contraseña: "123"
    }
);
LoadData( //OfflineControlAditivos
    DatosContolAditivosSinCargar;
    "DatosControlAditivosParaCargar";
    true
);
LoadData( //OfflineMedidasAlabeoHor
    DatosAlabeoHorSinCargar;
    "DatosAlabeoHorParaCargar";
    true
);
LoadData( //OfflineMedidasCalibreHor
    DatosCalibreHorSinCargar;
    "DatosCalibreHorParaCargar";
    true
);
LoadData( //OfflineTempHornos
    DatosTempHorSinCargar;
    "DatosTempHorParaCargar";
    true
);
```

Ilustración 39 Instrucciones para carga de información guardada en el dispositivo.

Datos sin conexión a internet.

Para esta funcionalidad haremos uso de galerías, colecciones y métodos de envío distintos a lo que ya se mencionó anteriormente

Iniciaremos colocando un icono que nos permita acceder a la pantalla donde se almacenaran y mostraran los datos capturados sin una conexión a internet, por lo que quedaría de la siguiente manera.



Ilustración 40 Icono de envío a vista de datos offline

También observamos un indicador que nos permite visualizar si la aplicación se encuentra conectada a internet, ya que se coloca de color rojo cuando no hay conexión y en verde cuando hay una conexión.

Le indicamos la navegación a otra pantalla en su propiedad OnSelect como ya se vio anteriormente, redirigiendo al usuario a la galería de datos sin conexión.

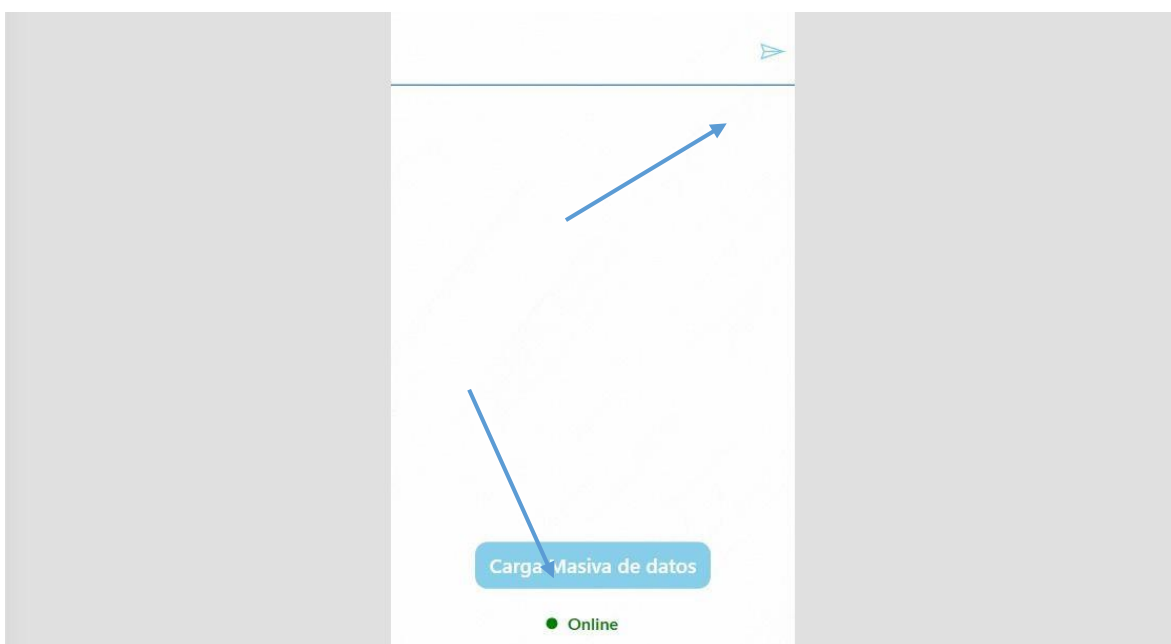


Ilustración 41 Pantalla de visualización de datos offline

Para enviar los datos sin conexión a la base de datos cuando el dispositivo ya tenga internet usamos dos botones para el envío de información, un botón para mandar datos de manera individual y el botón más grande para enviar una gran cantidad de datos a la vez.

Para el botón de envío individual, tendremos que escribir unas líneas de código para enviar la información en su propiedad OnSelect, seleccionamos el botón pequeño de la galería y escribimos lo siguiente en la barra de fórmulas:

```
Patch(
    LineasEsmaltado;
    Defaults(LineasEsmaltado);
    {
        Fecha: ThisItem.Fecha;
        Turno: ThisItem.Turno;
        'Línea de esmaltado': ThisItem.'Línea de esmaltado';
        Supervisor: ThisItem.Supervisor;
        Diseño: ThisItem.Diseño;
        Tipo: ThisItem.Tipo;
        Etapa: ThisItem.Etapa;
        Objetivo: ThisItem.Objetivo;
        'Valor obtenido': ThisItem.'Valor obtenido'
    }
);
If(
    IsEmpty(Errors(LineasEsmaltado));
    Remove(
        DatosLEsmaltadoSinCargar;
        ThisItem
    );
    SaveData(
        DatosLEsmaltadoSinCargar;
        "DatosLEsmaltadoParaCargar"
    )
);
Notify( "Datos enviados correctamente"; Success; 3000 );;
```

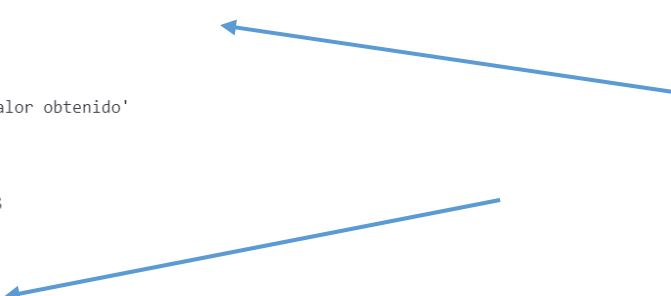


Ilustración 42 Método de envío de datos almacenados en el dispositivo con ThisItem

En la imagen anterior se muestra el código que hace que el botón pequeño de la galería envíe los datos de manera individual a la base de datos, esta ocasión como ya tenemos los datos guardados en una colección, no es necesario colocar el nombre del input delante del nombre de la columna donde se guardará la información, sino ahora colocaremos "ThisItem.Nombre de la columna donde enviaremos la información" de esta forma recuperamos los datos de la colección y los enviamos a la base de datos cuando haya internet, también en el código encontraremos un condicional que funciona para verificar que los datos se envíen a la base de datos correctamente, ya que de lo contrario guardara en otra colección los datos no se hayan enviado y borrarlos para que no haya datos duplicados, finalmente se envía una notificación si los datos se han cargado exitosamente.

Para el botón más grande con el texto de "Carga masiva de datos", deberemos usar el siguiente código para enviar los datos de forma rápida.

El código es similar a lo anterior mencionado para el envío de manera individual, solo que en este caso en lugar de usar “ThisItem” usaremos “ThisRecord” Esto nos permite enviar todos los datos que haya en la colección, finalmente enviando una notificación de envío exitoso.

```

ForAll(
    DatosLEsmaltadoSinCargar;
    Collect(
        DatosLEsmaltadoCargadosConExito;
        Patch(
            LineasEsmaltado;
            Defaults(LineasEsmaltado); {
                Fecha:ThisRecord.Fecha;
                Turno:ThisRecord.Turno;
                'Linea de esmaltado':ThisRecord.'Linea de esmaltado';
                Supervisor:ThisRecord.Supervisor;
                Diseño:ThisRecord.Diseño;
                Tipo:ThisRecord.Tipo;
                Etapa:ThisRecord.Etapa;
                Objetivo:ThisRecord.Objetivo;
                'Valor obtenido':ThisRecord.'Valor obtenido'
            }
        );
    );
);

If(
    IsEmpty(Errors(LineasEsmaltado));
    Clear(DatosLEsmaltadoSinCargar);
    SaveData(
        DatosLEsmaltadoSinCargar;
        "DatosLEsmaltadoParaCargar"
    );
    ClearCollect(
        DatosLEsmaltadoSinCargar;
        Filter(
            DatosLEsmaltadoSinCargar;
            !(Fecha in DatosLEsmaltadoSinCargar.Fecha)
        )
    );
    SaveData(
        DatosLEsmaltadoSinCargar;
        "DatosLEsmaltadoParaCargar"
    );
);
Notify( "Datos enviados correctamente"; Success; 3000 );

```

Ilustración 43 Uso de función ThisRecord para envío de datos de manera masiva

Agregar colecciones o datos a una galería.

Para agregar una colección o datos a una galería, deberemos seleccionar la galería y en la parte de orígenes de datos seleccionaremos la colección o el origen de datos que queramos conectar a la galería, de esta forma es como podemos acceder a la colección y mostrar los datos que contenga en la galería.

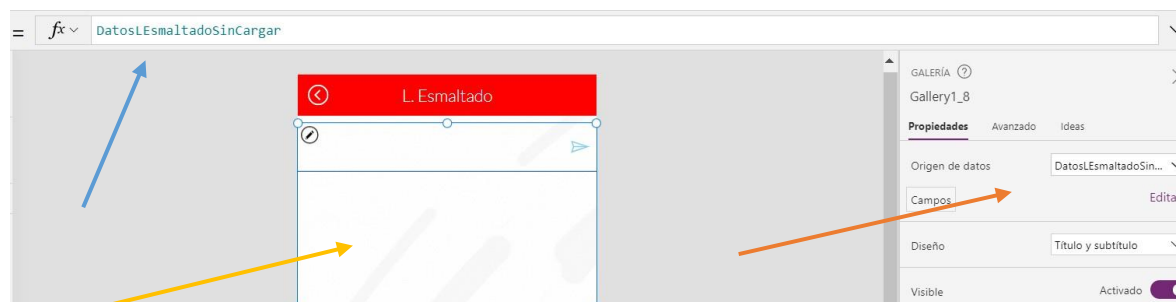


Ilustración 44 Conexión de galerías a orígenes/colecciones de datos

Desarrollo del backend

Para el backend se usó el lenguaje propio de Power apps, ya que al ser una plataforma low code, que como su nombre indica, el código empleado en la aplicación es mínimo, pero aun así para ciertas funcionalidades como el envío de información a la base de datos, la funcionalidad offline/online, y la creación de colección sí que se usan líneas de código para dar estas instrucciones a la aplicación.

Para la creación de la funcionalidad offline/online del sistema, lo primero que se hizo fue establecer la lógica que seguiría para funcionar, después fue implementarlo en la aplicación, lo cual, al inicio fue un poco complicado debido a la falta de experiencia con la tecnología, una vez que funciono para un área el siguiente paso era simplemente adecuarlo y replicarlos a las demás áreas, de esta forma es como se hizo en su mayoría el backend de la aplicación, otro punto también importante fue la creación de las colecciones, las cuales sirven para almacenar los datos ingresados sin conexión a internet, por lo que establecimos en la pantalla de inicio de la aplicación métodos que permite la carga de estas colecciones a la aplicación desde que se inicia, ya que al iniciar la aplicación toma los datos almacenados en el dispositivo como se establece en el código

El siguiente código es el que permite el envío de la información a la base de datos, específicamente a la tabla establecida con sus respectivas columnas y los componentes de donde obtendrán la información que se enviara, al igual que se establece que se cree una colección cuando no haya una conexión a internet mediante un condicional (if)

```
If(Connection.Connected;
Patch(MoliendaPP; Defaults(MoliendaPP); {
    Fecha:Text(Now());"mm/dd/yyyy hh:mm:ss";
    Turno:SelectTurnoPPMolienda.SelectedText;
    Molino:SelectMolinoPPMolienda.SelectedText;
    Supervisor:SelectSuperPPMolienda.SelectedText;
    'Densidad objetivo':InputDensidadObjPPMolienda.Text;
    'Densidad real':InputDensidadRealPPMolienda.Text;
    'Viscosidad real':InputViscosidadRealPPMolienda.Text;
    'Viscosidad objetivo':InputViscosidadObjPPMolienda.Text;
    'Rechazo objetivo':InputRechazoObjPPMolienda.Text;
    'Rechazo real':InputRechazoRealPPMolienda.Text;
    'Ajuste real':InputAjusteRealPPMolienda.Text;
    Ajuste:SelectDetAjusteMoliendaPP.SelectedText
});
Collect(DatosMoliendaPePSinCargar; {
    Fecha:Text(Now());"mm/dd/yyyy hh:mm:ss";
    Turno:SelectTurnoPPMolienda.SelectedText;
    Molino:SelectMolinoPPMolienda.SelectedText;
    Supervisor:SelectSuperPPMolienda.SelectedText;
    'Densidad objetivo':InputDensidadObjPPMolienda.Text;
    'Densidad real':InputDensidadRealPPMolienda.Text;
    'Viscosidad real':InputViscosidadRealPPMolienda.Text;
    'Viscosidad objetivo':InputViscosidadObjPPMolienda.Text;
    'Rechazo objetivo':InputRechazoObjPPMolienda.Text;
    'Rechazo real':InputRechazoRealPPMolienda.Text;
    'Ajuste real':InputAjusteRealPPMolienda.Text;
    Ajuste:SelectDetAjusteMoliendaPP.SelectedText
});
```

Ilustración 45 Código para envío de información a la base de datos

En la siguiente imagen, se muestra el código que permite enviar los datos almacenados sin conexión de manera individual, obteniendo los datos de la colección creada anteriormente, y en caso de error, almacenar los datos en otra colección y eliminar los que ya estaban, esto para evitar la duplicidad de los datos en la aplicación y al final, cuando los datos se hayan enviado correctamente aparecerá una notificación de color verde indicando que el envío de la información fue exitoso.

```
Patch(
    MoliendaPP;
    Defaults(MoliendaPP); {
        Fecha:ThisItem.Fecha;
        Turno:ThisItem.Turno;
        Molino:ThisItem.Molino;
        Supervisor:ThisItem.Supervisor;
        'Densidad objetivo':ThisItem.'Densidad objetivo';
        'Densidad real':ThisItem.'Densidad real';
        'Viscosidad real':ThisItem.'Viscosidad real';
        'Viscosidad objetivo':ThisItem.'Viscosidad objetivo';
        'Rechazo objetivo':ThisItem.'Rechazo objetivo';
        'Rechazo real':ThisItem.'Rechazo real';
        'Ajuste real':ThisItem.'Ajuste real';
        Ajuste:ThisItem.Ajuste
    }
);;

If(
    IsEmpty(Errors(MoliendaPP));
    Remove(DatosMoliendaPePSinCargar; ThisItem);;
    SaveData(DatosMoliendaPePSinCargar; "DatosMoliendaPPParaCargar")
);;

Notify( "Datos enviados correctamente"; Success; 3000 );;
```

Ilustración 46 Código para envío de información offline - individual

Ahora en la siguiente imagen, se muestra el código que enviar de forma masiva un conjunto de datos almacenados en el dispositivo que fueron ingresados en un momento si conexión a internet.

```
ForAll(
    DatosMoliendaPePSinCargar;
    Collect(
        DatosMoliendaPPCargadosConExito;
        Patch(
            MoliendaPP;
            Defaults(MoliendaPP); {
                Fecha:ThisRecord.Fecha;
                Turno:ThisRecord.Turno;
                Molino:ThisRecord.Molino;
                Supervisor:ThisRecord.Supervisor;
                'Densidad objetivo':ThisRecord.'Densidad objetivo';
                'Densidad real':ThisRecord.'Densidad real';
                'Viscosidad real':ThisRecord.'Viscosidad real';
                'Viscosidad objetivo':ThisRecord.'Viscosidad objetivo';
                'Rechazo objetivo':ThisRecord.'Rechazo objetivo';
                'Rechazo real':ThisRecord.'Rechazo real';
                'Ajuste real':ThisRecord.'Ajuste real';
                Ajuste:ThisRecord.Ajuste
            }
        );;
    );;
);;

If(
    IsEmpty(Errors(MoliendaPP));
    Clear(DatosMoliendaPePSinCargar);;
    SaveData(
        DatosMoliendaPePSinCargar;
        "DatosMoliendaPPParaCargar"
    );;
    ClearCollect(
        DatosMoliendaPePSinCargar;
        Filter(
            DatosMoliendaPePSinCargar;
            !(Fecha in DatosMoliendaPePSinCargar.Fecha)
        );;
    );;
    SaveData(
        DatosMoliendaPePSinCargar;
        "DatosMoliendaPPParaCargar"
    );;
);;

Notify( "Datos enviados correctamente"; Success; 3000 );;
```

Ilustración 47 Código para envío de información Offline - Carga masiva de datos

Finalmente se muestra el código implementado en la propiedad OnStart de la aplicación para cargar los datos almacenados en las colecciones al inicio de la aplicación, el código se muestra de manera general, abarcando ciertas líneas de código ya que al ser demasiadas instrucciones se vuelve un código extenso

```

);
LoadData(//OfflineControlAditivos
    DatosControlAditivosSinCargar;
    "DatosControlAditivosParaCargar";
    true
);
LoadData(//OfflineMedidasAlabeoHor
    DatosAlabeoHorSinCargar;
    "DatosAlabeoHorParaCargar";
    true
);
LoadData(//OfflineMedidasCalibreHor
    DatosCalibreHorSinCargar;
    "DatosCalibreHorParaCargar";
    true
);
LoadData(//OfflineTempHornos
    DatosTempHorSinCargar;
    "DatosTempHorParaCargar";
    true
);
LoadData(//OfflineAprobacionPdE
    DatosPrepEsmaltadoAproSinCargar;
    "DatosPrepEsmaltadoAproParaCargar";
    true
);
LoadData(//OfflineDepositoPdE
    DatosPrepEsmaltadoDepSinCargar;
LoadData(//OfflineMolinoPdE
    DatosPrepEsmaltadoSinCargar;
    "DatosPrepEsmaltadoParaCargar";
    true
);
LoadData(//OfflineCintaPP
    DatosCintaSinCargar;
    "DatosCintaParaCargar";
    true
);
LoadData(//OfflineSilosPP
    DatosSiloSinCargar;
    "DatosSiloParaCargar";
    true
);
LoadData(//OfflineMoliendaPP
    DatosMoliendaPePSinCargar;
    "DatosMoliendaPPPParaCargar";
    true
);
LoadData(//OfflineATHPP
    DatosATHPPSinCargar;
    "DatosATHPPPParaCargar";
    true
);
LoadData(//OfflineCisternaPP
    DatosCisternaPPSinCargar;
// coleccion ajuste
ClearCollect(
    'Determinación de ajuste';
    [
        "Ajustado";
        "No ajustado"
    ]
);

// coleccion Compuestos
ClearCollect(
    Compuestos;
    [
        "COMPUESTO BGOB 001-C";
        "COMPUESTO BGSM 005-C18";
        "COMPUESTO BGTC 003-C";
        "COMPUESTO CCM 52084";
        "COMPUESTO SEMIBRILLO COTTOFORTE";
        "COMPUESTO EGOM COTTOFORTE";
        "COMPUESTO MGB 00008";
        "CRISTALINA MGC-00157 TORRECID";
    ]
);

```

Ilustración 48 Código para la carga de datos almacenados al inicio de aplicación

A modo de resumen, eso fue el código implementado en la aplicación para la realización de ciertas funcionalidades necesarias, además adecuarlo a las demás áreas.

Pruebas.

Las pruebas del sistema fueron divididas por etapas, ya que al principio se realizaban las pruebas de manera interna, con el equipo de desarrollo, agregando información y observando el rendimiento, accesibilidad, eficiencia, etc. Normalmente siempre que se implementaba una nueva funcionalidad al sistema se probaba para comprobar que funcionara correctamente, una vez que la funcionalidad funcionara podíamos replicarla para los demás módulos con la certeza de que no habría problema con su funcionamiento, pues ya había sido revisada y aprobada.

Conforme el desarrollo del sistema, decidimos implementar un módulo de la aplicación a un área, de esta manera podríamos comprobar si el sistema era eficiente y funcional en producción, además que de encontrar errores podríamos corregirlos antes de implementarlo en todas las líneas de producción.

Para este caso en particular, decidimos probar en el área de hornos, específicamente para la medición y captura de los tiempos muertos en hornos, para esto fue necesario dar una capacitación al personal para mostrar y enseñar cual sería el rol de ellos en el sistema, además de capacitarlos para que pudiesen hacer uso de la aplicación y capturar la información requerida que necesitábamos.

Durante la capacitación del personal, algunos dieron puntos de mejora a la aplicación como por ejemplo componentes que les permitieran buscar los diseños por nombre, lo cual agilizaría el método de búsqueda y captura de la información, por lo que inmediatamente se hizo la implementación de los buscadores de información.

El periodo de estabilización del sistema en producción fue largo, ya que duro 1.5 meses en esa área, para revisar el comportamiento del personal y el sistema, los resultados que obtuvimos fueron los esperados, aunque al inicio el personal ingresaba la información de manera errónea o en ocasiones ni siquiera la ingresaban, conforme avanza el tiempo y la constante insistencia por parte de supervisores, jefes y el equipo de desarrollo, poco a poco comenzaron el ingreso de información de mejor manera, siendo más constante con los datos hasta el día de hoy que se ha normalizado el sistema para el personal de esa área y ahora ya no tienen ningún problema con la aplicación de Power apps. Ahora, respecto al rendimiento del sistema, específicamente de la aplicación, no se presentó ningún inconveniente, la aplicación funciona correctamente en todo momento hasta el día de hoy.

Cuando se finalizó con el desarrollo del sistema tanto de Power BI como la aplicación con Power apps, el siguiente paso era implementarlo en toda una línea en producción, de esta forma podríamos probar el rendimiento total del sistema en una línea piloto, la cual fue seleccionada mediante un análisis de personal que estaría participando en esta etapa de prueba. La distribución del sistema, y los dispositivos que se establecieron para el uso de la aplicación en las líneas al igual de roles establecido de las personas que se encargarían de ingresar la información al sistema quedo de la siguiente manera como se muestra en la siguiente imagen:

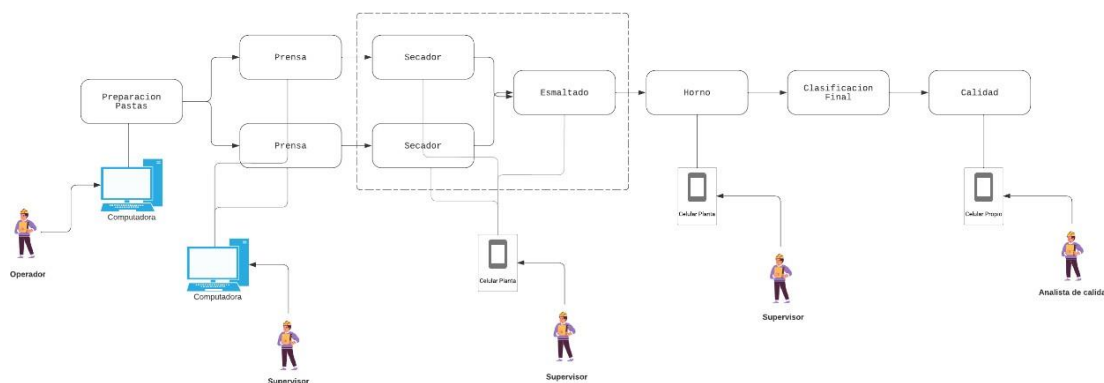


Ilustración 49 Distribución del sistema en línea piloto para pruebas.

Como podemos observar, en la imagen anterior, se muestra la distribución de la línea piloto para la prueba en producción de SCI, estableciendo que en el área de pastas el dispositivo que se usará será una computadora y el encargado del ingreso de la información al sistema será el operador, para el área de prensas también se usarán computadoras y el encargado de ingreso de información será el supervisor, en el secador y línea de esmaltado será mediante un dispositivo móvil el cual será usado únicamente por el supervisor en turno, seguidamente el área de hornos, los cuales ya cuentan con cierta experiencia en el uso de la aplicación solo que ahora se le implementa el uso de un celular para la captura de información, finalmente tenemos el área de calidad que al igual que las demás usará un celular y el encargado de ingresar la información será el analista de calidad.

Cabe recalcar que, para la implementación en esta línea piloto, se realiza la capacitación correspondiente al personal, para dicha capacitación se hace uso de material como videos, presentación y demostración en vivo del uso del sistema, por lo que no habría problema en la implementación.

Pruebas de eficiencia.

Se mide la eficiencia de la aplicación mediante las pruebas realizadas que se mencionan anteriormente, el resultado es que la eficiencia del personal operativo agiliza de cierta forma la actividad de captura de información, ya que, anteriormente escribían la información en un papel, y la ingresaba hasta el final de turno, por lo que este proceso era poco eficiente además que se perdía la continuidad en la fecha y hora en la que fue ingresada la información, cosa que no pasa en este sistema, ya que cada que ingresan un conjunto de datos, la fecha y hora se colocan de manera automática sin importar que hayan ingresado la información sin tener una conexión a internet. Ahora directamente ingresan la información sin la necesidad de escribir en papel y espera al final de turno para ingresar la información.

Pruebas de rendimiento.

El rendimiento de la aplicación y los dashboard de Power BI, son bastante buenos, ya que al menos durante el periodo de prueba, no hubo problemas ni quejas del personal por un mal rendimiento, lo cual nos da como resultado que el sistema funciona correctamente y tiene un excelente rendimiento en producción, aspecto que ayuda en gran medida a tener una muy buena experiencia de usuario.

Pruebas de usabilidad

La usabilidad de la aplicación es considerablemente buena, puesto que, de acuerdo a los comentarios recibidos por el personal, mencionan que la usabilidad del sistema es buena, pues cuenta con interfaces intuitivas y amigables para el usuario, por lo que al personal no le resulta complejo el uso de la aplicación, al contrario les fácil el uso, y son capaces de usar la herramienta al salir de la capacitación, ciertamente hay oportunidades de mejora en la usabilidad y en el resto de la funcionalidades, pero considerando que el sistema se encuentra en una fase inicial, se encuentra dando buenos resultado y consiguiendo buenas críticas por parte de los usuarios.

Pruebas de accesibilidad

Finalmente, para la accesibilidad en la aplicación, se cuenta con ayudas visuales, las cuales permiten al usuario ubicarse y saber que hacer en la aplicación, por ejemplo, cuando el usuario ingresa información en la aplicación, las letras cambian a un color que le permita identificar al usuario la información que ha ingresado, al enviar la información se muestran notificaciones de envió exitoso de dato al igual que se les muestran notificaciones de error en el caso que se presenten, esto permite que el usuario se sienta seguro al usar la aplicación, ya que no hay forma de que se pueda equivocar si es que sigue las instrucciones mencionadas en la aplicación.

Resultados

Como se observó durante el avance y desarrollo de este documento, pudimos notar que los resultados obtenidos de este proyecto son notablemente buenos, ya que el sistema permite una mejor trazabilidad de las variables de proceso de la planta, optimización en la captura de información en producción y KPI's, disponibilidad y centralización de la información, etc.

Los objetivos establecidos al inicio del proyecto fueron cumplidos exitosamente en la finalización del proyecto SCI.

Las interfaces desarrolladas son las mismas que observamos en las ilustraciones contenidas en la sección del desarrollo del frontend y el desarrollo de backend, que, a consideración personal, son amigables e intuitivas para los usuarios, para la aplicación, como se comentó anteriormente se desarrollaron aproximadamente 86 vistas en la aplicación, y aproximadamente 15 para los dashboard desarrollados en Power BI, por lo que el desarrollo fue bastante extenso, lográndose en un periodo de 4 meses.

Dicho proyecto fue presentado a el equipo de control de proceso de planta, esto con el objetivo de presentar y dar a conocer el nuevo desarrollo e innovación para control de proceso, además de buscar su apoyo para realizar la capacitación a todo el personal de líneas de producción e las que se implementará el sistema.

Conclusiones.

Para la finalización de este proyecto, queda más que decir que durante el proceso de desarrollo se encontraron ciertas dificultades que en ciertas ocasiones retasaron el proyecto, pero también, durante la implementación de este proyecto, hemos superado diversos desafíos y obstáculos que se presentaron en el camino, demostrando nuestra capacidad para adaptarnos y encontrar soluciones creativas. Además, hemos trabajado de manera colaborativa y hemos fomentado una comunicación abierta y transparente entre los miembros del equipo, lo que ha contribuido significativamente al logro de los resultados positivos.

Los beneficios obtenidos a través de este proyecto son evidentes. Hemos logrado optimizar el proceso de captura de información, mejorado la trazabilidad del negocio, la centralización, estandarización y disponibilidad de la información, etc. Estos resultados han impactado positivamente en la empresa y han demostrado el valor y la relevancia de este proyecto.

En conclusión, este proyecto ha sido un logro significativo para la organización. Los resultados obtenidos, la experiencia adquirida y los beneficios generados nos permiten avanzar hacia un futuro prometedor. Agradecemos todos los involucrados por su arduo trabajo, dedicación y compromiso en la realización de este proyecto, y esperamos seguir construyendo sobre este éxito en proyectos futuros.

Competencias desarrolladas.

En cuanto al aprendizaje obtenido, hemos adquirido nuevas habilidades, conocimientos y experiencia que nos servirán en futuros proyectos y nos permitirán mejorar nuestra eficiencia y efectividad. Además, hemos identificado áreas de mejora que podrían ser abordadas en proyectos futuros, y hemos aprendido valiosas lecciones sobre la importancia de la gestión del tiempo, la comunicación efectiva y la coordinación del equipo.

Algunos de los conocimientos y habilidades adquiridos durante este periodo de residencias y la realización fueron el uso de herramientas low code, como Power apps y Power BI, Power Automate, además de habilidades como el trabajo en equipo, comunicación, investigación, expresión de ideas, implementación de proyectos y finalmente la experiencia de trabajo y la realización de un proyecto real para una empresa de gran renombre, adquiriendo todas las experiencias y habilidades que conlleva el desarrollo de un nuevo sistema que permite la optimización de muchos procesos repetitivos del día.

Fuentes de información.

- ❏ KumarVivek. (2023, 16 marzo). *¿Qué es Power Apps?* - *Power apps*. Microsoft Learn. <https://learn.microsoft.com/es-es/power-apps/powerapps-overview>.
- ❏ Mihart. (2023, 23 marzo). *¿Qué es Power BI?* - *Power BI*. Microsoft Learn. <https://learn.microsoft.com/es-es/power-bi/fundamentals/power-bi-overview>
- ❏ Bartumeu, G. (2022). *¿Qué es Power Automate? Crea flujos de trabajo automatizados*. Aglaia. <https://aglaia.es/blog/power-automate/power-automate/>
- ❏ dbdocs.io - Database Documentation and Catalog Tool. (n.d.). <https://dbdocs.io/>
- ❏ IBM Documentation. (2021.). <https://virements-defining-use-cases>.
- ❏ Robledano, A. (2023, April 13). *Qué es MySQL: Características y ventajas*. OpenWebinars.net. <https://openwebinars.net/blog/que-es-mysql/>
- ❏ *¿Qué es SQL? - Explicación de lenguaje de consulta estructurado (SQL)* - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/sql/>
- ❏ Platzi. (2020.). *Modelo de Arquitectura "4+1."* <https://platzi.com/tutoriales/1248-pro-arquitectura/4142-modelo-de-arquitectura-41/>