**DISEÑO DEL MOCKUP PARA APLICACIÓN MÓVIL**

**CALIBRACIÓN DE ABONADORAS AP**

**ORLANDO CORREA MARTÍNEZ**

**1130631792**

**COMPUTACIÓN MÓVIL**

**HECTOR MANUEL VANEGAS SOLIS**

**UNIVERSIDAD SANTIAGO DE CALI**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS**

**PLANTEAMIENTO DEL PROBLEMA**

En la actualidad el sector agroindustrial presenta una gran oportunidad para empezar a implementar herramientas que encaminen hacia la revolución tecnológica, la era digital, tradicionalmente la información que se recopila en campo se hace manualmente empelando matrices en hojas de papel y escritos con lapicero, posteriormente estos documentos son entregados a un tercero para que lo digite en hojas de cálculo o en la plataforma BIOSALC (base de datos corporativa) para centralizar la información.

En la sección de aplicación de fertilizantes sólidos para el cultivo de la caña de azúcar, se han implementado tecnologías de precisión para mejorar la distribución del producto en el campo, esto a través de salidas reguladas controladas por un motor hidráulico; el giro de este equipo dispone la salida de material de forma no uniforme, lo que requiere diariamente un proceso de calibración, permitiendo ajustar los cálculos del equipo para entregar adecuadamente el producto en el suelo, que está sujeto a los cambios de velocidad.

En primera instancia se propuso llenar una matriz en hojas tamaño carta donde se escriben los valores manualmente, se toma una foto y se envía por WhatsApp, en la oficina se descarga la imagen y se digita en hojas de cálculo para analizar el comportamiento de los datos y por ultimo estimar el comportamiento del equipo. Este proceso se realiza semanalmente, por tanto, las correcciones son tardías y con mala interpretación. De forma emergente la calidad de los datos empezó a afectar la calidad en el análisis, lo que llevo a cuestionar qué medidas se debían tomar, a lo que se concluyó que la baja calidad de las imágenes combinado con una caligrafía deficiente causaba problema en interpretaciones de los datos.

El problema identificado a raíz de este análisis es la baja calidad de entrega de datos, lo que requiere una estandarización digital del procedimiento de captura de información, por lo que se propone implementar una aplicación que mitigue totalmente las ambigüedades en el registro y que permita al usuario hacer un análisis en tiempo real de la información permitiendo hacer correcciones en la estructura de la abonadora para mejorar la aplicación de una forma oportuna.

**JUSTIFICACIÓN**

Las herramientas se crearon para facilitar el diario ejercer en la ingeniería, en el campo aún más, la fuerza física paso a segundo plano con la revolución industrial, y en nuestra era digital y su cuarta revolución es justo que el campo se vea beneficiado de instrumentos que garanticen el nuevo insumo mundial, los datos, calidad y oportunidad. Implementar una app que permita de forma intuitiva recopilar la información, controlar los datos que se digitan y centralizar en una base de datos, será una gran oportunidad para dar garantías de un buen y oportuno análisis de los equipos en campo, lo que en consecuencia generará una mejora en la aplicación del insumo agrícola (normalmente categorizado como uno de los más costosos para el cultivo).

Una aplicación móvil es una herramienta diseñada para ser ejecutada en Celulares, tabletas y otros dispositivos inteligentes, lo cual permite al usuario ejecutar tareas de cualquier tipo, sea de modo profesional, educativo, de servicio o afines. En esta propuesta se empleará el SO Android, ya que todo el personal cuenta con al menos una tableta asignada por la empresa y con un teléfono personal de soporte.

**OBJETIVO**

Proponer un diseño de una aplicación móvil para gestionar la información de las calibraciones de las abonadoras de precisión marca Verion y Vcom

**OBJETIVOS ESPECÍFICOS**

* Crear una base de datos que se conecte a una API y permita el tráfico de datos con la APP y permita hacer CRUD.
* Proponer un modelo estandarizado del proceso de digitación a partir de la experiencia obtenida en campo.
* Construir la primera versión funcional de Login y captura de información con persistencia.

**ESTUDIO TÉCNICO**

Se evaluarán los aspectos técnico operativos que se requieren para determinar la viabilidad y posteriormente la factibilidad del desarrollo del aplicativo móvil que permita la validación e interpretación de los datos capturados.

* 1. **CARACTERISTICAS DEL SOFTWARE**

Debe ser un software que brinde una alta funcionalidad, confianza, compatibilidad y facilidad en el manejo, priorizando la interfaz de usuario acorde a la diaria exigencia del proceso, fiel a la comodidad y practicidad para quien lo usa y eficiente para quien trate los datos posteriormente.

* + 1. Comunicación eficiente con la base de datos.
    2. Interpretación y sugerencias para el usuario cuando la calidad de datos no cumpla con los estándares de calidad.
    3. Despliegue de registros y gráficos de tendencia de cada equipo usado
    4. Sistema de identificación y validación de usuarios.
    5. Debe validar todo datos de entrada, además posterior a cada registro debe entregar una interpretación de los datos y permitir los ajustes necesarios para mejorar el estado de la calibración, mitigando es sesgo estadístico.
    6. Debe permitir al usuario comunicarse de forma rápida con los encargados de tratamiento de datos para optimizar los tiempos de ejecución de actividades en campo.
    7. Tener en cuenta los colores adecuados para trabajo bajo el sol.
  1. **CARACTERISTICAS DEL HARDWARE**

Computador para programar

* + 1. Procesador Intel Core i5 (9th Gen)
    2. Memoria RAM 12GB
    3. Disco estado sólido 128 GB
    4. Tarjeta aceleradora de gráficos 4GB
    5. Dos pantallas 19”
    6. Celular Android + Calble USB tipo C
  1. **ETAPAS Y TIEMPO DE DESARROLLO DE SOFWARE**

|  |  |  |
| --- | --- | --- |
| ETAPA | RESPONSABLE | TIEMPO |
| *Requerimientos*: El objetivo es obtener los requerimientos del aplicativo, así como la mayor cantidad de funciones que se deben ofrecer a los distintos tipos de usuarios, (Capturado de información, Directores de procesos, Analistas de datos) | **Estudiante ingeniería Orlando Correa Martínez** | **1 semana** |
| *Planteamiento*: Se revisa la estructura de la aplicación, el tiempo y los detalles de las funciones, se debe escribir y presentar claramente las funcionalidades y comportamiento que se va a desarrollar. | **Estudiante ingeniería Orlando Correa Martínez** | **2 semanas** |
| *Diseño*: Se determina el cómo se va a desarrollar la APP, la estructura de la base de datos, el lenguaje de programación, determinar las librerías y permisos necesarios para el desarrollo. | **Estudiante ingeniería Orlando Correa Martínez** | **2 semanas** |
| *Implementación*: Se hace la instalación en dispositivos celulares y tabletas, puesta en marcha de la aplicación y la conexión a la API y la base de datos | **Estudiante ingeniería Orlando Correa Martínez** | **8 horas** |
| *Pruebas*: Se realizan para dar garantía del funcionamiento de la APP y encontrar los errores que esta presente, así como la validación de funcionalidades no percibidas durante el diagnóstico y diseño. | **Estudiante ingeniería Orlando Correa Martínez** | **1 semana** |

* 1. **CARACTERISTICAS TECNICAS IMPORTANTES**
     1. ***PostgreSQL*** es un sistema de base de datos relacional de alta disponibilidad. Es capaz de funcionar de manera estable en el servidor y, por lo tanto, resulta robusto, una de las principales características que buscan las empresas. Además, es consistente y tolerante a fallos. Es compatible con el modelo relacional, ya que asegura siempre su integridad referencial. Es un sistema que puede trabajar en sistemas operativos como Windows, Linux y Mac. El sistema soporta interfaces como Delphi, PHP, Java y .net
     2. ***pgAdmin*** es una herramienta indispensable para gestionar y administrar PostgreSQL, la base de datos de código abierto más avanzada del mundo. Por lo tanto, pgAdmin es la herramienta para gestionar nuestras bases de datos espaciales PostGIS. La interfaz de usuario es más flexible, permite que las pestañas se acoplen y reorganicen en más formas que anteriormente. Ahora es posible disponer de pestañas desmontables, lo que permite que la Herramienta de consulta (Query Tool) y el Depurador se abran en pestañas nuevas y luego se puedan mover a pantallas alternativas. Importante tener en cuenta: La arquitectura de pgAdmin 4 requiere, como es natural, más CPU y memoria que una aplicación de escritorio tradicional.
     3. ***Docker*** te permite crear una imagen de contenedor y usar esa misma imagen a lo largo de todo el proceso de despliegue. Una gran ventaja de esto es la capacidad de separar pasos no dependientes del proceso y ejecutarlos en paralelo. El tiempo que va desde la compilación a la producción se puede acelerar bastante. Esto se debe al hecho de que crea un contenedor para cada proceso y no arranca un sistema operativo. Los datos se pueden crear y eliminar sin temer que el coste de tenerlo que arrancarlo todo otra vez sea mayor de lo que se pueda permitir.
     4. ***SpringBoot*** es un módulo del proyecto de Spring que fue creado para simplificar el desarrollo de aplicaciones con Spring Framework bajo licencia Apache 2.0. Pone a nuestro alcance una infraestructura para el desarrollo de aplicaciones en una plataforma de lenguaje Java de código abierto, que hace mucho más fácil la vida de los programadores ahorrando tiempo y costes, sin por ello sacrificar control sobre el código ni rendimiento. Con Spring Boot es rápido desarrollar y es rápido de ejecutar. Apoyado en todo el framework de desarrollo Spring, las aplicaciones hechas con esta tecnología está optimizadas para altas cargas de trabajo con un mínimo consumo de memoria, gracias a que la mayoría de entidades son singletons y por tanto objetos java reutilizados entre threads. Esta aproximación mantiene mínima la memoria RAM necesaria por cada cliente simultáneo, convirtiéndolo en sistemas óptimos para entornos web con miles de peticiones por segundo.
     5. ***Android Studio*** es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:
* Un sistema de compilación flexible basado en Gradle
* Un emulador rápido y cargado de funciones
* Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
* Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla
* Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
* Variedad de marcos de trabajo y herramientas de prueba
* Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
* Compatibilidad con C++ y NDK
* Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine
  + 1. ***GitHub*** es una plataforma de alojamiento, propiedad de Microsoft, que ofrece a los desarrolladores la posibilidad de crear repositorios de código y guardarlos en la nube de forma segura, usando un sistema de control de versiones llamado Git. Facilita la organización de proyectos y permite la colaboración de varios desarrolladores en tiempo real. Es decir, nos permite centralizar el contenido del repositorio para poder colaborar con los otros miembros de nuestra organización. GitHub está basada en el sistema de control de versiones distribuida de Git, por lo que se puede contar con sus funciones y herramientas, aunque GitHub ofrece varias opciones adicionales y su interfaz es mucho más fácil de manejar, por lo que no es absolutamente necesario que las personas que lo usan tengan un gran conocimiento técnico. Aquí puedes conocer más sobre su historia.
    2. ***Funcionalidades***: Se debe especificar las cosas que se pueden hacer y las que no desde el aplicativo. Se debe ser muy claro y utilizar medios lícitos para probar lo que no se podrá hacer a través de la app.
    3. ***Licencia y condiciones de uso***: Se deben estipular licencias de uso y condiciones para el usuario para aceptar el uso de la aplicación donde no podamos eximir de responsabilidades que puedan reclamar por el mal uso del aplicativo.
    4. ***Información y permisos***: Se debe ser altamente explícito al solicitar permisos al usuario. Muchas veces los aplicativos van a necesitar permisos con contactos de agenda y contenidos del móvil, ya sea simplemente por compartir contenidos.
  1. **RECURSO HUMANO**

Debido a que es un desarrollo educativo, el único recurso humano responsable del desarrollo y demás actividades es el estudiante quien presenta el documento, sin embargo, el apoyo de otros compañeros es fundamental para solucionar problemas de uso y percepción de funcionalidades requeridas, lo cual al final en las conclusiones se hará la respectiva mención en agradecimiento por la participación indirecta del proceso.

**DISEÑO DE MOCKUP Y EXPLICACION DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES**

**FASE 1:**

|  |  |
| --- | --- |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1-1.png** | **INGRESO:** Se requiere un inicio de sesión por parte del usuario para validar su identificación y poder relacionar los datos que este genere. Dispone un cuadro de texto para el correo electrónico y otro para la contraseña. Posteriormente un botón de inicio para enviar la consulta.  De fondo el botón debe enviar el correo electrónico como parámetro a una consulta a la API creada especialmente para esta APP, la cual consulta directamente la base de datos y devuelve un mensaje de error si no lo encuentra, y, si lo encuentra, trae todo el registro en JSON del usuario, compara la contraseña con la digitada por el usuario y si todo coincide entonces accede a la APP.  **TAB**: En la parte superior se dispone de una TAB para navegar entre el ingreso y el registro de usuario. |

|  |  |
| --- | --- |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1-2.png** | **REGISTRO:** Dispone cuadros de texto para nombre de usuario, correo electrónico, su confirmación, contraseña y su confirmación. Internamente se hacen las validaciones de que correspondan los datos ingresados y de no ser iguales se exponen mensajes y enfoques sobre los cuadros de texto inválidos. El botón de registro permite ejecutar las acciones mencionadas, además, una vez validaos los datos de ingreso envía una consulta con el correo como parámetro para validar la existencia de dicho usuario, de existir, se envía un mensaje de error, si no, envía todo el registro hacia la API para crear el nuevo registro, una vez terminado muestra un mensaje y se pasa a la vista de inicio de sesión automáticamente. (No inicia sesión con el registro) |

**FASE 2:**

|  |  |
| --- | --- |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-0.png | **VISTA PRINCIPAL**: Incluye un menú, dos botones para acceder a las herramientas propósito de la aplicación y un botón menú flotante para acceder a las redes sociales de la organización. |

|  |  |
| --- | --- |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1.png | **MENU**: Se necesita un menú para funciones que se puedan consultar en todas las vistas, en ellas se especifica el nombre del usuario, La opción de contacto para comunicarse con el equipo técnico o el personal encargado de los datos, la galería de fotos para enviar información relevante sobre los equipos, el manual de instrucciones para consultar procedimientos, un calificativo de la app y el botón de cierre de sesión. |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1-1-1.png | **CONTACTO**: Se dispone una vista para informar a los usuarios los medios de comunicación con los desarrolladores y el analista de datos. Cuenta con dos botones para encaminar la consulta al responsable adecuado. |

|  |  |
| --- | --- |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1-1-A.PNG | **CONTACTO DESARROLLADOR**: En esta vista se proporciona el contacto WhatsApp del desarrollador, con la intención de abrir la aplicación con un mensaje previo. |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1-1-B.PNG | **CONTACTO ANALISTA**: Tiene el mismo funcionamiento que la vista anterior, solo que se contacta con el analista de datos responsable del tratamiento de la información. |

|  |  |
| --- | --- |
|  | **GALERÍA**: Vista que contacta con la galería de imágenes del usuario y así poder compartir información relevante sobre el estado de un equipo. Permite seleccionar una o más imágenes y compartirlas. |
|  | **MANUAL**: Es una vista con barra deslizadora y poder visualizar el paso a paso del uso de la app, tiene tarjetas informativas con detalle de procesos puntuales. |

|  |  |
| --- | --- |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1-4.png | **CIERRE DE SESIÓN**: Boton que muestra la inforacipon del usuario, su actividad con la app en cuanto al promedio de uso e inicios de sesión total. Cuenta con un botón de cierre de sesión, el cual cierra todos los procesos d la app y obliga al inicio de sesión nuevamente. |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-1-5.png | **CALIFICACIÓN**: Opción para calificar la app, el cual mide el nivel de satisfacción del usuario final, se permite un cuadro de texto para ingresar las observaciones necesarias y con esto evaluar si se hacen mejoras en futuras actaulizaciones. |

|  |  |
| --- | --- |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-2-1.png** | **REDES SOCIALES**: Se crea un botón flotante que dispone de un menú con las redes sociales de la organización, de esta forma poder compartir los logros de los usuarios. |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-3-A.PNG** | **CALIBRACIÓN**: Vista con el desarrollo de la primera herramienta funcional, consiste en un formulario con varios tipos de captura de información. Cuenta con listados desplegables para elegir información fija, para producto (Urea o Mezcla), Estado (soca o plantilla) y Abonadora (código interno del equipo), un botón que lleva a una nueva vista y permite capturar la información de posicionamiento geográfico. Tirulos para las 4 salidas que representan cada uno de los descargues de abono solido en el equipo, una grilla de 12 capturadores de información de tipo numérico, lo que representa 3 repeticiones de la calibración, cada casilla debe validar si el valor digitado corresponde a un digito valido, de no ser así, se mostrará un mensaje indicando el error. Si el valor es considerado como válido para el usuario a pesar de la advertencia se deja un registro de esta decisión para que el analista de datos pueda tener en cuenta. El botón calcular se encarga de tomar todos los datos y validar estadísticamente si la calibración es válida y muestra el resultado en la casilla de texto no editable llamada “FACTOR”. El botón “GUARDAR” toma toda la información digitada y la adiciona a la base de datos a través de la API. |

|  |  |
| --- | --- |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-3-B.PNG** | **UBICACIÓN**: Vista que emplea la API de google Maps para capturar la ubicación del dispositivo, la guarda y confirma visualmente al usuario si es real su ubicación. La marca queda reportada en el mapa y se puede consultar en cualquier momento, no se puede editar esta ubicación ya que se busca que los datos sean capturados en campo y no en oficina. Requerimiento para fomentar el uso de al app. |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-3-C.PNG** | **LISTADO HISTORICO CALIBRACIÓN**: La vista en general cuenta con un TAB para navegar entre la digitación y los registros. Cuenta con un listado y una barra vertical deslizable, el cual automáticamente es actualizado con la comunicación a la base de datos, empela un método Get All. Se organiza de más actualizado al más antiguo. |

|  |  |
| --- | --- |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-3-D.PNG** | **INFORMACIÓN COMPLETA DEL REGISTRO**: Cuando se presiona sobre algún registro aparece una ventana emergente que muestra toda la información al detalle, un botón para compartir de forma puntal. Se requiere también que exista una forma de editar los registros. Con la presión fuera del cuadro se cierra el dialogo. |
| **C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-4-A.PNG** | **REGISTROS DE SUERTES**: Interfaz que permite registrar los datos del lote a abonar. Cuenta con listas desplegables para cada variable: Nombre de la hacienda, numero de la suerte, indicador si tiene mapa o no de prescripción, tipo de producto aplicado, valor de la dosis, una posición para crear una dosis nueva, el cual valida si los datos están acorde a los mínimos y máximos posibles manejados a nivel general. El botón guardar permite compilar toda la información y enviarla a la base de datos a través de la API. Una vez se competa en envío se retorna a la vista principal y se limpian los campos. |

|  |  |
| --- | --- |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-4-B.PNG | **LISTADO HISTORICO DE SUERTES ABONADAS**: Cuenta con un listado y una barra vertical deslizable, el cual automáticamente es actualizado con la comunicación a la base de datos, empela un método Get All. Se organiza de más actualizado al más antiguo. |
| C:\Users\NickFury\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3-4-C.PNG | **INFORMACIÓN COMPLETA DE LA SUERTE ABONADA**: Cuando se presiona sobre algún registro aparece una ventana emergente que muestra toda la información al detalle, un botón para compartir de forma puntal. Se requiere también que exista una forma de editar los registros. Con la presión fuera del cuadro se cierra el dialogo. |