

# GUÍA DE DISEÑO DE HMIs BAJO EL SISTEMA OPERATIVO ANDROID.



**ALEXANDER DUEÑAS SALAZAR**  
**DIEGO JAIR BRAVO**

Director: Ing. Vladimir Trujillo Arias

*Universidad del Cauca*  
*Facultad de Ingeniería Electrónica y Telecomunicaciones*  
**Departamento de Electrónica, Instrumentación y Control**  
*Popayán, Septiembre de 2014*

# GUÍA DE DISEÑO DE HMIs BAJO EL SISTEMA OPERATIVO ANDROID.



**ALEXANDER DUEÑAS SALAZAR  
DIEGO JAIR BRAVO**

**Monografía presentada como requisito parcial para optar por el título de  
Ingenieros en Automática Industrial.**

Director: Ing. Vladimir Trujillo Arias

*Universidad del Cauca*  
*Facultad de Ingeniería Electrónica y Telecomunicaciones*  
*Departamento de Electrónica, Instrumentación y Control*  
*Ingeniería en Automática Industrial*  
*Popayán, Septiembre de 2014*

## TABLA DE CONTENIDO

INTRODUCCION.....	1
1. Conceptos Generales .....	3
1.1 Supervisión de procesos.....	3
1.2 HMI (Interfaz Hombre Maquina). .....	3
1.3 Diseño centrado en el usuario (DCU). .....	3
1.4 MPlu+a (modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad). 5	
1.5 Sistema operativo Android.....	5
1.5.1 Kernel de Linux.....	7
1.5.2 Bibliotecas o Librerías. ....	7
1.5.3 Entorno de Ejecución. (Android Runtime). .....	7
1.5.4 Marco o Framework de aplicaciones. ....	7
1.5.5 Aplicaciones. ....	8
1.6 Ciclo de Vida de una Actividad. ....	9
1.7 Interfaz de Usuario. ....	10
2. GUIA DE DISEÑO DE HMIS BAJO EL SISTEMA OPERATIVO ANDROID. ....	11
2.1 Presentación.....	11
2.2 Análisis de Requisitos.....	11
2.2.1 Análisis Etnográfico. ....	12
2.2.2 Análisis contextual de las tareas.....	13
2.2.3 Objetos. ....	13
2.2.4 Plataforma .....	13
2.3 Diseño. ....	14
2.3.1 Diseño de la actividad.....	14
2.3.2 Diseño de la información. ....	15
2.3.3 Estilo. Estrategias de diseño de la información a nivel industrial.....	15
2.3.4 Ventanas informativas. ....	16
2.3.5 Señalización. ....	22
2.3.1 Elementos Gráficos. ....	26
2.4 Prototipado. ....	54
2.5 Evaluación.....	57

2.5.1	Evaluación Heurística.....	62
2.6	Implementación.....	65
2.7	Lanzamiento.....	65
2.8	Resumen de la guía propuesta.....	66
2.8.1	Análisis de requisitos.....	66
2.8.2	Diseño de la interacción.....	68
2.8.3	Implementación.....	71
2.8.4	Lanzamiento.....	72
3.	Aplicación de la guía propuesta a un caso de estudio.....	73
3.1	Análisis de Requisitos.....	73
3.1.1	Clasificación de los usuarios.....	73
3.1.2	Análisis Etnográfico.....	73
3.1.3	Análisis contextual de las tareas.....	76
3.1.4	Objetivos.....	80
3.2	Diseño de la interacción.....	81
3.2.1	Diseño de la actividad.....	81
3.3	Diseño de la información.....	86
3.3.2	Story board navegacional.....	91
3.4	Implementación.....	94
3.5	Evaluación del prototipo Software.....	98
3.6	Lanzamiento.....	100
4.	Mecanismo de comunicación entre una HMI Android y un proceso gobernado por un PLC.....	101
4.1	Estándar De Comunicación OPC.....	101
4.1.1	Arquitectura OPC.....	102
4.1.2	Estándar OPC en LabVIEW.....	102
4.2	Comunicación TCP/IP Bidireccional entre LabVIEW y el dispositivo móvil.....	104
4.2.1	Arquitectura TCP/IP.....	104
4.2.2	Protocolo TCP/IP y configuración en LabVIEW.....	105
4.2.3	Protocolo TCP/IP en JAVA.....	108
4.2.1	Mecanismo de comunicación final.....	110
5.	Conclusiones.....	112
6.	Trabajos Futuros.....	114

7. Contribuciones.....	114
8. Bibliografía.....	115

### TABLA DE ILUSTRACIONES

Figura 1. 1. Etapas del DCU. Se realiza una iteración constante hasta satisfacer los objetivos propuestos. ....	4
Figura 1. 2. MPlu+a. ....	5
Figura 1. 3. Arquitectura de Android. ....	6
Figura 1. 4. Ciclo de vida de una Actividad .....	10
Figura 1. 5. Alerta con título. ....	19

Figura 2. 1. Primera fase de análisis de requisitos, el análisis etnográfico influye en las demás actividades de esta fase. ....	12
Figura 2. 2. Prototipo de baja fidelidad (Boceto) .....	13
Figura 2. 3. Cuadro de mensaje común en una aplicación de Windows.....	17
Figura 2. 4. Estructura general de un dialogo en Android. ....	18
Figura 2. 5. Cuadro de dialogo con acciones específicas. ....	18
Figura 2. 6. Ejemplos de diálogos típicos utilizados en el sistema operativo Android.....	19
Figura 2. 7. Alerta sin título. ....	19
Figura 2. 8. Figura 1. 5. Alerta con título. ....	19
Figura 2. 9. Figura. Mensaje toast en la aplicación WordPress.....	20
Figura 2. 10. Alerta por encendido de la planta.....	20
Figura 2. 11. Mensaje Toast para el encendido de un equipo o planta. ....	21
Figura 2. 12. Principio de coherencia.....	21
Figura 2. 13. Aspecto coherente en los diálogos a nivel industrial. ....	21
Figura 2. 14. Información clara y concisa.....	21
Figura 2. 15. Información poco clara.....	21
Figura 2. 16. Claridad en el dialogo .....	22
Figura 2. 17. Información Concisa. ....	22
Figura 2. 18. Dialogo Personalizado en Android. ....	22
Figura 2. 19. Ejemplos de señales a nivel industrial.....	24
Figura 2. 20. Tubería de agua potable. ....	25
Figura 2. 21. Tubería con riesgo de explosión y fuego.....	25
Figura 2. 22. Tubería de gas con radiación ionizante.....	25
Figura 2. 23. Tubería de aceite con indicación de dirección de flujo. ....	25
Figura 2. 24. Hmi Rockwell, las grandes áreas vacías tienen un color neutro: gris. ....	26
Figura 2. 25. Colores de referencia en Android.....	27
Figura 2. 26. Paleta de colores del sistema operativo Android.....	27
Figura 2. 27. Representación por color y texto de los estados de un motor. ....	28
Figura 2. 28. Representación por color de los estados de una válvula.....	28
Figura 2. 29. Pre-alarma para la velocidad de un motor.....	29
Figura 2. 30. Alarma para la velocidad de un motor. ....	30
Figura 2. 31. Límites para las alarmas de señales analógicas. ....	30

Figura 2. 32. Fondo de la aplicación sobre el estado del tiempo WeatherPro .....	32
Figura 2. 33. Fondos de pantalla recomendados a nivel industrial para Android.....	32
Figura 2. 34. Fondo real con superposiciones (WinCC, Siemens). .....	33
Figura 2. 35. Fondo real con superposiciones .....	33
Figura 2. 36. Patrón Z, Regla de Gutenberg. ....	34
Figura 2. 37. Zonas de pantalla Recomendadas.....	34
Figura 2. 38. Postura común para manejar una Tablet en Portrait y Landscape. ....	35
Figura 2. 39. Postura común para manejar un Smartphone en Portrait.....	35
Figura 2. 40. Distribución de pantalla propuesta en [22] .....	35
Figura 2. 41. Distribución de pantalla para un dispositivo móvil en orientación landscape (izquierda) y portrait (derecha). En esta última orientación, la entrada de los datos se realiza en otra ventana.....	36
Figura 2. 42. Action Bar en el tema Holo light. ....	37
Figura 2. 43. Cursor de navegación del Action Bar (Izquierda) y Botón de Atrás (Derecha). .....	37
Figura 2. 44. Navegación en la aplicación Gmail. 3 pantallas: Home del dispositivo móvil (Izquierda), Lista de conversaciones (centro) y detalles de conversación. ....	37
Figura 2. 45. Ejemplos de un Spinner en un Action Bar. ....	38
Figura 2. 46. Icono Action Overflow. Se observa que se encuentra siempre a la derecha del Action Bar, en la imagen se despliega un Menú con varias acciones. ....	38
Figura 2. 47. Manejo de pestañas en orientación portrait.....	40
Figura 2. 48. Manejo de Pestañas en orientación Landscape. ....	40
Figura 2. 49. Diseño aceptable. RSVIEW32.....	40
Figura 2. 50. Diseño muy pobre .....	41
Figura 2. 51. Al hacer contacto sobre los campos de texto el teclado aparece automáticamente.....	43
Figura 2. 52. TextFields que cuentan con la propiedad Hint, para ingresar Email y Password. ....	43
Figura 2. 53. Indicadores de mando convencionales. ....	44
Figura 2. 54. Funcionalidad en indicadores de mando. ....	44
Figura 2. 55. Mandos deslizantes. ....	45
Figura 2. 56. . Movimiento coherente.....	45
Figura 2. 57. Protección del mando. ....	45
Figura 2. 58. Seguridad en el mando. ....	45
Figura 2. 59. Posición por secuencia. ....	46
Figura 2. 60. Posición por sentido único. ....	46
Figura 2. 61. Mandos deslizantes, dos sentidos.....	46
Figura 2. 62. Tabla de alarmas y eventos de InTouch.....	47
Figura 2. 63. Tabla de alarmas y eventos Ignition Mobile. ....	48
Figura 2. 64. Distribución de Views dentro de TableRows que a su vez están contenidos en un TableLayout. ....	48
Figura 2. 65. Superposición de varios Views (grises) en un TableLayout (fondo rojo).....	49
Figura 2. 66. Tabla de Alarmas en Android.....	49
Figura 2. 67. Gráfica de tendencias del software Ignition de Inductive Automation .....	50
Figura 2. 68. Tabla de tendencias del software Wonderware.....	50
Figura 2. 69. Ejemplos de graficas desarrolladas en AndroidPlot. ....	51
Figura 2. 70. Abuso de mayúsculas y subrayados. ....	52

Figura 2. 71. Tipo de fuente y color.....	53
Figura 2. 72. Escalas para el tamaño de fuente en Android. Se debe recordar que el tamaño de fuente se especifica en sp. ....	53
Figura 2. 73. Prototipo de papel (boceto).....	55
Figura 2. 74. Ejemplos Story board de papel, además de las diferentes pestañas se muestra flechas que significan navegabilidad en la aplicación. ....	56
Figura 2. 75. Jerarquía de ventanas básicas en una aplicación HMI industrial .....	57
Figura 2. 76. Clasificación de los métodos de evaluación. ....	59
Figura 2. 77. Reunión de implicados.....	66
Figura 2. 78. Actividades principales en el análisis de requisitos. El análisis etnográfico repercute sobre las otras actividades. ....	67
Figura 2. 79 Esquema general para la obtención de los objetivos de la aplicación a desarrollar. ....	68
Figura 2. 80. Proceso para el desarrollo de la maqueta digital.....	70
Figura 2. 81. Prototipo Story board para la supervisión de un proceso. ....	71
Figura 2. 82. Implementación de la HMI a desarrollar. ....	72
Figura 3. 1. Proceso general para la supervisión del proceso de temperatura. ....	74
Figura 3. 2. Interfaces para la supervisión del proceso de temperatura del LCP. Ventana principal del proceso (izquierda) y ventana de graficas (derecha). ....	76
Figura 3. 3. Primer prototipo de baja fidelidad: Boceto.....	78
Figura 3. 4. Focus Group para la evaluación del primer prototipo. ....	79
Figura 3. 5. Modelo HTA correspondiente al ingreso de a la aplicación .....	82
Figura 3. 7. HTA para monitorización del proceso .....	83
Figura 3. 6. HTA para monitorización de las gráficas del proceso.....	83
Figura 3. 8. HTA para configuración de parámetros PID y set point.....	84
Figura 3. 9. HTA para monitorear las alarmas.....	85
Figura 3. 10. HTA para observar descripción del proceso de temperatura.....	85
Figura 3. 11. HTA para observar la ayuda referente a los eventos y alarmas del proceso	86
Figura 3. 12. Representación de la baliza del proceso de temperatura. ....	87
Figura 3. 13. Representación de la planta de temperatura.....	87
Figura 3. 14. Ingreso de datos para el control de proceso de temperatura.....	87
Figura 3. 15. Primera maqueta digital del sistema. Desarrollada con la herramienta software Pencil Project. Utilizada para evaluar el diseño de la información.....	88
Figura 3. 16. Segunda Maqueta digital desarrollada en la herramienta software INKSCAPE.....	90
Figura 3. 17. Prototipo de Story Board (modelo de navegabilidad.) ....	91
Figura 3. 18. Recorrido cognitivo por parte de los usuarios.....	93
Figura 3. 19. Valores correspondientes a las alarmas del proceso. ....	95
Figura 3. 20. Interfaces en la orientación Landscape implementadas en el IDE Eclipse, para la aplicación móvil de supervisión del proceso de temperatura. ....	96
Figura 3. 21. Interfaces en la orientación Portrait implementadas en el IDE Eclipse, para la aplicación móvil de supervisión del proceso de temperatura.....	97
Figura 3. 22. Estructura de usabilidad de MUMMS] .....	99
Figura 3. 23. Distribución de pantalla final propuesta luego de aplicar la guía de diseño	100

Figura 4. 1. Objetivo principal de comunicación. Envío y recepción de datos de una manera bidireccional entre el dispositivo de control y el dispositivo móvil. ....	101
Figura 4. 2. Esquema de una Arquitectura OPC .....	102
Figura 4. 3. Servidor NI OPC Servers .....	103
Figura 4. 4. Cliente Rápido de NI OPC Servers .....	103
Figura 4. 5. Ejemplo de Uso de una Tag OPC en Código de LabVIEW .....	104
Figura 4. 6. Arquitectura TCP/IP y su familia de protocolos. ....	105
Figura 4. 7. Paleta de Funciones TCP/IP en LabVIEW .....	106
Figura 4. 8. Configuración de un Cliente en LabVIEW .....	106
Figura 4. 9. Configuración de un Servidor en LabVIEW .....	107
Figura 4. 10. .vi desarrollado para el envío y recepción de datos hacia y desde el dispositivo móvil. ....	108
Figura 4. 11. Diagrama de Flujo de Datos Usando Sockets en Java.....	109
Figura 4. 12. Mecanismo de Comunicación con el protocolo TCP/IP .....	110
Figura 4. 13. Mecanismo de Comunicación con TCP/IP Programas .....	111

## INDICE DE TABLAS

Tabla 1. 1. Framework de Android .....	8
Tabla 2. 1. Algunos métodos utilizados con sus principales características. ....	15
Tabla 2. 2. Niveles de percepción de colores, según ANSI. ....	23
Tabla 2. 3. Parámetros de color y forma para señales de seguridad según Norma NTC 1461.....	23
Tabla 2. 4. Colores de identificación de tuberías Norma Colombiana NTC 3458. ....	24
Tabla 2. 5. Representación por color de los Estados. ....	28
Tabla 2. 6. Representación por color de las Alarmas. ....	28
Tabla 2. 7. Niveles de prioridad de las alarmas. Dependiendo de la prioridad, el mensaje de la alarma aparecerá en pantalla con colores diferentes.....	31
Tabla 2. 8. Nro. de Ítems para el Action Bar en algunos dispositivos. “O” denota un Ítem del Action Bar, “=” denota un Icono Action Overflow. Portrait, la posición vertical del dispositivo y Landscape la posición horizontal. ....	39
Tabla 2. 9. Complementos para elementos de mando. ....	47
Tabla 2. 10. Clasificación de prototipos o técnicas de prototipado. ....	54
Tabla 2. 11. Técnicas de evaluación por inspección .....	59
Tabla 2. 12. Métodos de evaluación por indagación. ....	60
Tabla 2. 13 Métodos de evaluación por test.....	61
Tabla 2. 14. Métodos de evaluación con sus principales características.....	62
Tabla 2. 15. Información de los usuarios finales del sistema a desarrollar. ....	67
Tabla 3. 1. Resultados obtenidos con la herramienta MUMMS para evaluar usabilidad con los usuarios finales.....	99

## **GLOSARIO DE TERMINOS**

**GUI:** Interfaz Gráfica de usuario.

**HMI:** Interfaz Hombre – Máquina.

**MPIu+a:** Modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad.

**IDE:** Entorno de desarrollo integrado.

**OPC:** OLE para procesos de Control.

**OLE:** Enlace e incrustación de objetos.

**INSHT:** Instituto nacional de seguridad e higiene en el trabajo.

**ISO:** Organización internacional de Normalización.

**PVD:** Pantallas de visualización de datos.

**ANSI:** instituto Americano de Normalización

**NTC:** Norma Técnica colombiana.

**RTU:** Unidad terminal remota.

**API:** Interfaz de programación de aplicaciones.

**TCP:** Protocolo de control de transmisión.

**IP:** Protocolo de transmisión.

**LCP:** Laboratorio de control de procesos de la universidad del Cauca.

**OSI:** Sistema de interconexión abierta.

# INTRODUCCION

La comunicación e interacción (interacción hombre – máquina) entre los usuarios humanos y los procesos industriales que cuentan con plantas y sistemas técnicos dinámicos, ha sido reconocida como esencial para llevar a cabo una buena gestión de la seguridad, la calidad y la eficiencia de procesos de producción [1]. Esta interacción se lleva a cabo por medio de una herramienta que cobra un valor fundamental dentro de cualquier proceso industrial que cuente con algún grado de automatización: la interfaz hombre - máquina (HMI por sus siglas en inglés). Estas interfaces han logrado que la comunicación entre el usuario (operadores, ingenieros, personal de mantenimiento, gestores, etc.) y el proceso industrial se lleve a cabo de una manera eficiente y productiva, y lo que es más, han permitido que se lleve a cabo la supervisión y control del proceso en tiempo real, aspecto relevante dentro del ámbito productivo industrial.

El avance progresivo en lo referente a la automatización de procesos industriales ha hecho que las empresas que ofrecen soluciones de automatización se valgan de herramientas software como los sistemas SCADA (sistemas de adquisición, control y supervisión de datos dentro del proceso) que están compuestos de subsistemas como las HMIs que se convierten en la herramienta gráfica donde se despliegan todos los datos del proceso al usuario, para que este pueda manipularlo según lo crea conveniente (supervisión y control del proceso). En este sentido, la forma como se le presenten los datos al usuario se vuelve fundamental para que se pueda llevar a cabo una buena gestión de la información; el diseño de la interfaz toma mayor relevancia. Aunque este es un tema importante que se debe tener en cuenta, en la actualidad no existe una norma específica que brinde parámetros de diseño para elaborar HMIs industriales, en este punto es importante anotar que desde el año 2005 el comité ISA – SP 101 ha estado desarrollando un estándar: el SP 101<sup>1</sup>, que contempla varios criterios y recomendaciones para el diseño de estas interfaces.

Con todo lo planteado anteriormente y teniendo como premisa que el usuario humano se vuelve fundamental en la supervisión de los procesos industriales mediante una HMI, es que se puede inferir que el desarrollo de estas interfaces se debe realizar pensando en los modelos mentales de los usuarios, en sus necesidades y requerimientos, ya que son estos en últimas, los que interactúan con el proceso. “El diseño se debe llevar a cabo desde un punto de vista de la centralidad humana, es decir, con respecto a la orientación del usuario, sus procesos cognitivos, sus necesidades y capacidades” [1]. Lo anterior recae en la inclusión de la filosofía del diseño centrado en el usuario (DCU) para elaborar las interfaces Hombre – Máquina.

Por otra parte, debido al desarrollo exponencial que han tenido las comunicaciones a nivel industrial, la supervisión de los procesos ha llegado a tal punto, que en la actualidad se lleva a cabo la supervisión y control de una manera remota a través de herramientas tecnológicas como los servicios WEB o en menor medida, los dispositivos móviles. La evolución de estos últimos ha ido acompañada del desarrollo de sistemas operativos independientes (Android, IOS, Windows Mobile, etc.) con capacidad de procesamiento de información, conexión constante con la web, interacción agradable con el usuario final, etc., siendo Android el más

---

<sup>1</sup> Esta planeado que para el año 2015 sea publicado un primer borrador de este estándar.

consolidado a nivel mundial<sup>2</sup>, razón por la cual se tomó como referencia para el desarrollo del proyecto.

Esto último, unido al desarrollo hardware de los dispositivos móviles que ha sobrepasado fronteras con los Smartphones y tablets - ya que pueden llevar a cabo infinidad de tareas cada vez más complicadas- ha hecho que se abran nuevas líneas de investigación encaminadas a explorar nuevos entornos para las aplicaciones móviles: medicina, robótica, domótica, automatización y aunque hay grandes avances en el ámbito de la domótica y la medicina, las aplicaciones a nivel industrial son muy pocas<sup>3</sup>.

En este orden de ideas lo que se pretende con este proyecto es dar un primer paso hacia el diseño de HMIs industriales bajo el sistema operativo Android, teniendo como base la teoría del DCU, los patrones de diseño de Android y las recomendaciones y directrices a nivel industrial para el diseño de estas interfaces.

## **Estructura del Documento.**

El presente documento está organizado en 4 capítulos, cuyos contenidos son los que se describen a continuación:

En el capítulo 1 se describe de una manera muy general los conceptos necesarios para contextualizar al lector en la guía de diseño de HMIs propuesta.

En el capítulo 2 se expone la guía de diseño de HMIs bajo el sistema operativo Android propuesta, en esta, se hace una relación estrecha entre la teoría del MPLu+a, las recomendaciones para el diseño de HMIs a nivel industrial y los parámetros de diseño del sistema operativo Android para desarrollar aplicaciones. Debido a la extensión de la Guía, en el presente documento se omiten algunos conceptos teóricos que se detallan en el Anexo A. Aun así, en los anexos digitales se encuentra la versión extendida de la guía propuesta.

El capítulo 3 hace referencia a la aplicación de la guía propuesta en un caso de estudio: la creación de un sistema interactivo para la supervisión del proceso de temperatura del LCP de la universidad del Cauca.

Por ultimo en el capítulo 4, se detalla un mecanismo de comunicación propuesto, para el intercambio de datos entre un proceso industrial gobernado por un PLC y un dispositivo móvil Android.

---

<sup>2</sup> Un estudio de la internacional Data corporation (IDC) arrojo que en el año 2013, un 78.6 % de todos los Smartphones tenían integrado este sistema operativo, seguido por un 15.2 % de su competidor IOS.

<sup>3</sup> Solo unas cuantas empresas han dado los primeros pasos hacia la supervisión de procesos industriales en dispositivos móviles, entre estas se encuentran Inductive Automation, Indusoft, Integraxor.

# 1. Conceptos Generales

Lo que se pretende en este espacio es contextualizar al lector respecto a varios aspectos que se tratan en el documento.

## 1.1 Supervisión de procesos.

El concepto de supervisión en el ámbito industrial se entiende como el conjunto de acciones desempeñadas con el propósito de asegurar el correcto funcionamiento del proceso, incluso en situaciones anómalas. Estas acciones se llevan cabo por medio de usuarios especializados, que detectan la presencia de comportamientos anómalos y actúan según lo crean necesario, ajustando parámetros, cambiando consignas y activando accionamientos para prevenir un mal superior o conservar la capacidad operativa del proceso [2]. Este proceso de supervisión se lleva a cabo por medio de herramientas tecnológicas como los sistemas SCADA, que permiten el envío, la recepción, el transporte, la organización y presentación de la información importante para que el usuario pueda llevar a cabo sus tareas [3].

## 1.2 HMI (Interfaz Hombre Maquina).

Una HMI (interfaz hombre – Máquina, por sus siglas en inglés) es la forma interactiva de contacto entre el usuario y el proceso que se esté llevando a cabo. Esta informa al usuario acerca del estado del proceso, muestra tendencias, alarmas y permite llevar a cabo procedimientos para que el operario realice acciones en tiempo real para mitigar situaciones anómalas [4]. En la actualidad son muchas las empresas que dentro de sus soluciones de automatización ofrecen paquetes software para la elaboración de interfaces gráficas que cuentan con propiedades de animación, combinación de colores, sinópticos para representar el proceso, etc., y aunque estas son poderosas herramientas con las que cuenta el usuario, no se ha llevado a cabo un buen uso de estas debido a que no se cuenta con criterios de estandarización para la elaboración de las HMIs, lo que conlleva a la no existencia de una normativa a aplicar [2].

Teniendo en cuenta aspectos importantes como, que el diseño de las HMIs es fundamental dentro de todo proceso industrial que cuente con esta herramienta, y que el diseño de estas interfaces se tiene que realizar teniendo siempre en cuenta al usuario y sus necesidades, sus procesos cognitivos, su modelo mental etc. se puede abordar el tema del diseño centrado en el usuario (DCU).

## 1.3 Diseño centrado en el usuario (DCU).

Es una aproximación al diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso. Así, se puede entender el DCU como una filosofía cuya premisa es que, para garantizar el éxito de un producto, hay que tener en cuenta al usuario en todas las fases del diseño [5].

Uno de los objetivos fundamentales del DCU es brindar el mayor índice de usabilidad en el diseño que se lleve a cabo, entendiéndose como usabilidad como una medida de la utilidad, facilidad de uso, facilidad de aprendizaje, la satisfacción y apreciación con que las personas realizan sus tareas gracias al producto con el que se está trabajando. En este sentido el

grado de usabilidad de un sistema interactivo<sup>4</sup> es un aspecto relacionado con la interfaz de usuario [6].

El DCU es visto como un proceso cíclico en donde el diseño se realiza siempre teniendo al usuario presente y donde la usabilidad del diseño es evaluada de forma iterativa y mejorada incrementalmente.

El Estándar ISO 13407 (Human-Centred design processes for interactive systems) define 4 actividades principales durante el ciclo de vida del desarrollo (Figura 1. 1).

- Entender y especificar el contexto de uso.
- Especificar los requisitos de usuario y de la organización.
- Producir soluciones de diseño.
- Evaluar los diseños en base a los requisitos.

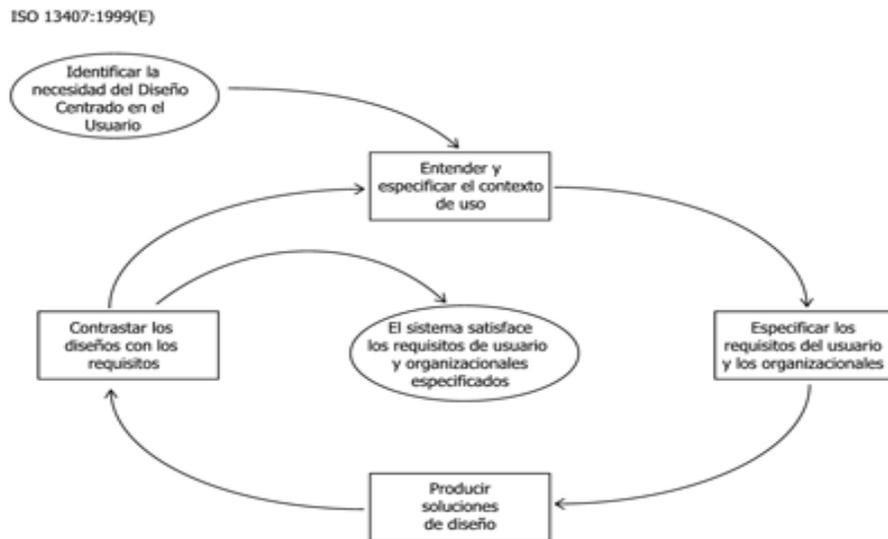


Figura 1. 1. Etapas del DCU. Se realiza una iteración constante hasta satisfacer los objetivos propuestos.

[Fuente: <http://labda.inf.uc3m.es/awa/en/node/102>]

Se debe tener en cuenta que estas son definiciones de la ISO dentro de una filosofía de diseño por lo cual, no está completamente definido. No se muestra una metodología de desarrollo “estándar” para llevar a cabo en un desarrollo propuesto.

Debido a la complejidad de aplicar los estándares ISO en un desarrollo , otras aproximaciones han definido modelos y marcos de trabajo para integrar las actividades de Diseño Centrado en el Usuario dentro del proceso de desarrollo, como el framework LUCID (Kreitzberg; 2008) o el modelo **MPlu+a** (Granollers; 2004) [7]. Siendo este último el que se tendrá en cuenta para el desarrollo del proyecto.

---

<sup>4</sup> Un sistema interactivo es un sistema informático que se interrelaciona y depende de las acciones de un usuario para realizar una tarea, es decir, todo sistema en el que interactúan persona y máquina

## 1.4 MPlu+a (modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad).

Es un Modelo propio de DCU desarrollado por el grupo de investigación GRIHO de la universidad de Lledia en España, que además de tener en cuenta la usabilidad en su metodología, integra el concepto de accesibilidad. El modelo presenta 3 fases importantes (3 pilares básicos) que se relacionan entre si y que tienen al usuario siempre presente en todo el proceso del desarrollo (Figura 1. 2):

- La Ingeniería del Software, en el formato "clásico" de ciclo de vida en cascada iterativo o evolutivo (columna de la izquierda de color azul).
- El prototipado (columna central de color verde), como metodología que engloba técnicas que permitirán la posterior fase de evaluación.
- La evaluación (columna de la derecha, de color amarillo) que engloba y categoriza a los métodos de evaluación existentes.

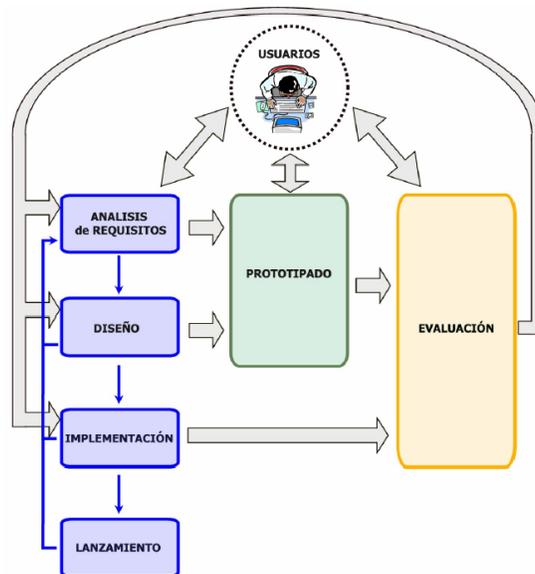


Figura 1. 2. MPlu+a.  
[Fuente: [6] ]

El modelo está estructurado de una manera sencilla y no establece pasos ni iteraciones específicas a seguir, por lo que lo convierte en un modelo flexible que se puede utilizar libremente por los equipos desarrolladores para alcanzar el objetivo de elaborar sistemas interactivos altamente usables y accesibles. Este modelo será la herramienta principal para desarrollar la guía de diseño propuesta, por lo cual se abordará de una manera más profunda en el siguiente capítulo.

## 1.5 Sistema operativo Android.

Inicialmente fue desarrollado por Android Inc. Firma que fue adquirida por Google en 2005 y no fue sino hasta 2008 que se popularizó, gracias a la unión del proyecto de Open Handset Alliance que es un consorcio en el cual se encuentran 48 empresas de desarrollo hardware, software y telecomunicaciones, quienes promocionaron el software libre, aunque es Google quien ha publicado la mayor parte del código fuente del sistema operativo [8].

Android es un sistema operativo y una plataforma software basada en Linux para dispositivos móviles<sup>5</sup>. Para realizar aplicaciones en este, se programa en un entorno de trabajo (framework) de Java y una máquina virtual Dalvik con compilación en tiempo de ejecución.

En la figura 1.3 se observa la arquitectura del sistema operativo, la cual está dividida en capas que serán descritas a continuación de una manera muy general:

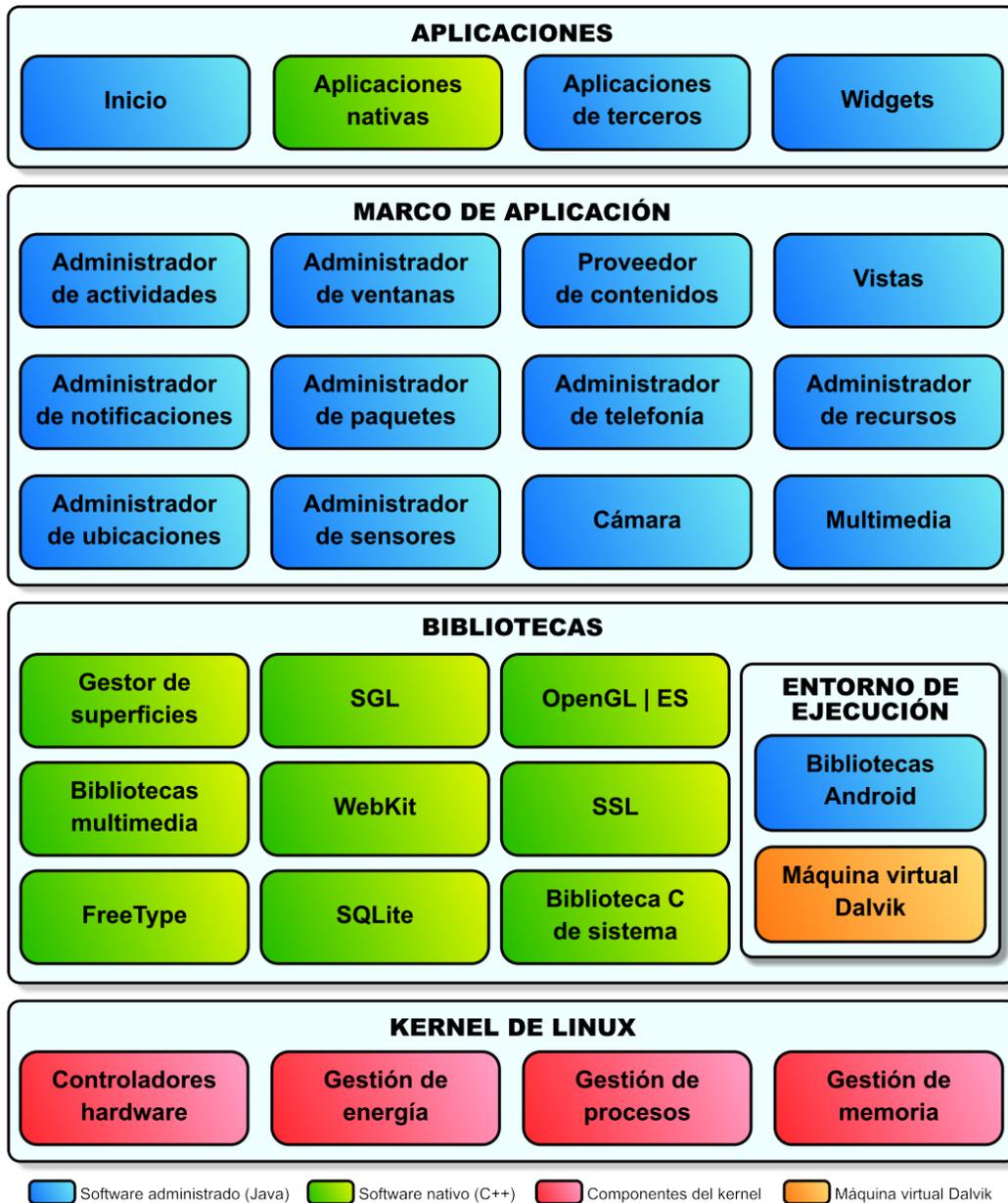


Figura 1. 3.Arquitectura de Android.  
[Fuente: [9]]

<sup>5</sup> En un principio se diseñó para Smartphones, luego se expandió para soportar dispositivos como tablets, televisores y hasta relojes.

### **1.5.1 Kernel de Linux.**

Situada en la parte inferior de la pila (rojo), proporciona una capa fundamental de abstracción del hardware, así como servicios básicos, tales como gestión de procesos, memoria y sistema de archivos. En el Kernel es donde están implementados los drivers específicos del hardware, para Wi-Fi o Bluetooth por ejemplo. La pila de Android está diseñada para ser flexible con muchos otros componentes opcionales que dependerán en gran medida de la disponibilidad de hardware específico en un dispositivo determinado. Estos componentes incluyen características tales como el tacto en las pantallas, cámaras, receptores GPS y acelerómetros [10].

### **1.5.2 Bibliotecas o Librerías.**

En la siguiente capa (en verde), se observan las librerías, las cuales fueron creadas para el desarrollo de aplicaciones Android desde las cuales las aplicaciones acceden a los servicios que provee el kernel [11].

### **1.5.3 Entorno de Ejecución. (Android Runtime).**

Es el entorno de ejecución, ubicado en la misma capa de las librerías, está compuesto por las librerías de núcleo las cuales son librerías con múltiples clases java y la máquina virtual Dalvik [12].

### **1.5.4 Marco o Framework de aplicaciones.**

Representa el conjunto de herramientas de desarrollo de cualquier aplicación. Cualquier aplicación desarrollada en Android utiliza el mismo conjunto de API y el mismo framework que se presenta en esta capa.

En el framework de aplicaciones se encuentran un conjunto de servicios y subsistemas que incluyen:

- Componentes gráficos, como listas, grillas, campos de texto, botones y también vista Web que puede utilizarse para la construcción de aplicaciones.
- Proveedores de contenido, que permiten a las aplicaciones Android acceder a información que proviene de otras aplicaciones y también compartir su propio contenido.
- Administración de notificaciones, que permite a las aplicaciones desplegar alertas en una barra de estado en el dispositivo.
- Administración de actividades, donde gestiona el ciclo de vida de las aplicaciones. [8].

En la tabla 1.1 se observan las APIs más importantes de esta capa.

Nombre	Descripción
Activity Manager	Gestiona el ciclo de vida de las aplicaciones
Window Manager	Gestiona las ventanas de las aplicaciones
Telephone Manager	Gestiona las funcionalidades propias del teléfono (llamadas, mensajes, etc.)
Content Provider	Permite a una aplicación compartir sus datos con otras aplicaciones.
View System	Proporciona elementos para realizar interfaces de usuario (GUI)
Location Manager	Permite a las aplicaciones obtener información de localización y posicionamiento y funcionar según esta información.
Notification Manager	Permite comunicar al usuario los eventos que ocurren durante la ejecución de una aplicación.
Package Manager	Permite la recuperación de diversos tipos de información relacionados con los paquetes de aplicaciones que se encuentran instalados en el dispositivo.
Resource Manager	Permite el acceso a recursos sin código, como cadenas localizadas, gráficos y archivos de diseño.

*Tabla 1. 1. Framework de Android  
[Fuente: [12]]*

### 1.5.5 Aplicaciones.

Finalmente en la parte superior se encuentra la capa de aplicaciones la cual contiene las aplicaciones a las cuales se tiene acceso desde el dispositivo, estas aplicaciones tienen acceso a las APIs (Interfaz de programación de aplicaciones) del framework de aplicaciones y a las librerías de los niveles anteriores.

Todas las aplicaciones se escriben en el lenguaje de programación Java y se construyen a partir de componentes, que son puntos de entrada por medio del cual el sistema puede acceder a una aplicación, aunque, no todos los componentes son verdaderos puntos de entrada para el usuario, estos son una pieza única que ayuda a definir el comportamiento general de una aplicación. Existen cuatro tipos de componentes los cuales cumplen un propósito y tienen ciclos de vida diferentes.

- **Actividades (Activities):** Es una ventana en donde el usuario puede realizar una actividad (tomar una foto, ver un mapa, marcar un número, etc.) a través de una interfaz de usuario. Una aplicación puede tener muchas actividades que trabajan juntas para formar una experiencia coherente dentro del sistema, aunque las actividades son independientes unas de otras.
- **Servicios (Services):** Es un componente de una aplicación que puede realizar operaciones de larga ejecución en segundo plano y no proporciona una interfaz de usuario [13], también es usado para comunicarse con procesos remotos. Un

ejemplo de un servicio es reproducir música en segundo plano mientras el usuario está en otra aplicación.

- Proveedores de contenido (Content Providers): Gestionan un conjunto de datos compartidos de una aplicación. Pueden almacenar información en el dispositivo, en una base de datos SQLite, en la Web, o en cualquier otro lugar al que la aplicación tenga acceso. Otras aplicaciones pueden consultar o modificar la información de su interés por medio del proveedor de contenido.
- Receptores Broadcast (Broadcast Receivers): Responden a anuncios generados en el sistema, estos anuncios podrían ser: que el sistema informe el apagado de la pantalla, la batería es baja, que una imagen fue tomada con la cámara, etc.

Las actividades, servicios y receptores broadcast, se activan mediante un mensaje asíncrono llamado intención (Intent). Las intenciones unen los componentes individuales en tiempo de ejecución, son mensajeros que solicitan una acción de otros componentes, aun sin importar que el componente pertenezca a otra aplicación [8].

## 1.6 Ciclo de Vida de una Actividad.

Aunque los componentes de las aplicaciones Android tienen ciclos de vida diferentes, es muy importante priorizar en conocer el ciclo de vida de una actividad (activity) el cual es el componente principal de las aplicaciones. Estas actividades son ventanas que contienen interfaces de usuario, y permiten al usuario realizar tareas. Una aplicación se compone generalmente de varias actividades, que se comunican por medio de intenciones (intents), las cuales le permiten al usuario una experiencia de interacción con la aplicación.

El ciclo de vida de una aplicación está controlado principalmente por las actividades, debido a que el usuario no cambia de aplicación si no de actividad. Es importante mencionar que en Android el ciclo de vida de las aplicaciones es controlado principalmente por el sistema. En este sentido, el sistema es quien decide qué proceso debe ser destruido basándose en el conocimiento de los componentes que están corriendo (principalmente actividades y servicios), que tan importantes son para el usuario y con cuanta memoria se cuenta en el momento.

Si al eliminar un proceso de una aplicación, el usuario vuelve a ella, se creara de nuevo el proceso, pero se habrá perdido el estado que tenía la aplicación, en este caso si será responsabilidad del programador si se quiere almacenar el estado de las actividades, para cuando la aplicación sea reiniciada.

Una actividad en Android tiene cuatro estados:

- Activa (Running): La actividad es visible y puede ser utilizada por el usuario, o está en uso.
- Visible (Paused): La actividad es visible pero no ocupa toda la pantalla. Esto sucede cuando otra actividad pasa a “activa” de una forma transparente o que no ocupa toda la pantalla.
- Parada (Stopped): Cuando una actividad es no visible.
- Destruída (Destroyed): Cuando la actividad termina usando para esto el método finish(), o es “eliminada” por el sistema Android (Sale de la pila de actividades).

En la Figura 1. 4 se observa el ciclo de vida de una actividad y los métodos que capturan los eventos que en ella suceden.



Figura 1. 4. Ciclo de vida de una Actividad  
 [Fuente: <http://jarroba.com/activity-entender-y-usar-una-actividad/>]:

## 1.7 Interfaz de Usuario.

La interfaz de usuario es la principal sección de interacción entre el usuario y el dispositivo, es por medio de ésta que se accede a todas la funcionalidades disponibles. Como bien se conoce dentro de una actividad se encuentra la interfaz de usuario, pero para construir esta interfaz se emplean diferentes objetos los cuales son llamados Views, estos se componen de botones, etiquetas, etc.

Para construir estos Views y organizarlos según se requiera se utiliza un layout el cual es un recurso en el cual se puede describir lo que se quiere mostrar en la interfaz de usuario, los layouts se programan en código XML, que tiene un formato similar a HTML. Para asociar el layout a la actividad, este debe ser cargado desde el código de la actividad en el método onCreate(). [30]

Teniendo en cuenta todos estos conceptos teóricos – introductorios se plantea la guía de diseño de HMIs industriales bajo el sistema operativo Android, que es el resultado del desarrollo del proyecto.

## 2. GUIA DE DISEÑO DE HMIS BAJO EL SISTEMA OPERATIVO ANDROID.

### 2.1 Presentación.

La presente guía de diseño de HMIs industriales bajo el sistema operativo Android, se ha realizado bajo criterios y recomendaciones planteados por varios autores y organizaciones a través de los años. Además de pautas, sugerencias, y parámetros creados y utilizados por los desarrolladores del sistema operativo, para la implementación de interfaces gráficas de usuario (GUI, por sus siglas en inglés) –parte fundamental dentro del desarrollo de cualquier aplicación móvil–.

Uno de los objetivos fundamentales de la guía es adaptar toda una serie de criterios y recomendaciones industriales, con los patrones de diseño existentes en el sistema operativo Android para el desarrollo de cualquier aplicación. Con lo anterior, y teniendo en cuenta que la HMI desarrollada es una interfaz centrada en el usuario<sup>6</sup>; se ha tenido como punto de referencia el modelo **MPlu+a (Modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad)**, descrito en el capítulo 1, para la elaboración de cualquier clase de HMI digital industrial con la guía propuesta.

La guía no pretende ser una norma establecida para el diseño de interfaces hombre – máquina digitales, sino una herramienta con unos criterios claros acerca de cuáles son los principales aspectos a tener en cuenta a la hora de realizar un HMI industrial, todo esto bajo el sistema operativo Android. Además, pretende ser una herramienta flexible para el desarrollador, en tanto que no se debe seguir estrictamente todos los pasos propuestos ni utilizar todas las recomendaciones de diseño expuestas. Su contenido tendrá el mayor número de sugerencias para que pueda ser utilizada en cualquier clase de proyecto a nivel industrial.

El MPlu+a establece una serie de fases (Figura 1. 2) que serán el punto de partida en el desarrollo de la siguiente guía de diseño de HMIs bajo el sistema operativo Android.

### 2.2 Análisis de Requisitos.

Esta primera fase está relacionada con la Ingeniería de requisitos, la cual comprende todas aquellas tareas encaminadas a la obtención de las necesidades (requerimientos que por lo general son cambiantes) de los usuarios con respecto a un software, aplicación o producto en concreto [14].

Ya que en esta primera fase prima el usuario, sus requerimientos y el contexto en que realiza sus tareas, se deben llevar a cabo una serie de técnicas y recomendaciones para definir de una manera clara y específica todos los requisitos que se puedan generar, esto debido a que el usuario por lo general no sabe en primera instancia qué es lo que quiere. De esta manera se obtendrán resultados positivos que reducirán el grado de error de la aplicación en las posteriores fases del desarrollo.

---

<sup>6</sup> Es considerada una interfaz centrada en el usuario en el sentido de que es precisamente el usuario (el operador, ingeniero encargado del proceso, etc.) el que realiza las tareas de supervisión y control del sistema por medio de la interfaz.

El proceso de análisis de requisitos, está dividido en diferentes actividades.

### 2.2.1 Análisis Etnográfico.

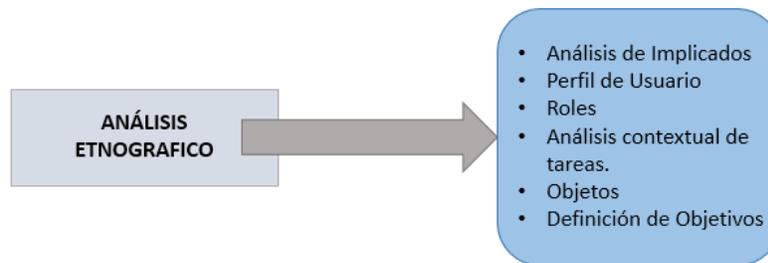


Figura 2. 1. Primera fase de análisis de requisitos, el análisis etnográfico influye en las demás actividades de esta fase.

[Fuente: [15]]

El trabajo etnográfico<sup>7</sup> se centra en realizar un estudio de observación del entorno del usuario con la finalidad de concebir los modelos mentales de estos [14]. Con este estudio se logrará al final describir el contexto en el que se sitúa el usuario, además de aumentar la información relativa a la organización de las tareas y a su consecución, ya que cada usuario ve de una manera diferente cada tarea.

En primera instancia se debe llevar a cabo una interacción con el usuario para conocerlo y tener un primer acercamiento a su manera de pensar, el objetivo es generar confianza en él, con el fin de que realice su trabajo de una manera natural y sin contratiempos. Para esto, se debe disponer de un tiempo razonable.

Una vez se ha interactuado y conocido al usuario, se debe prestar atención a cada una de sus actividades, al mismo tiempo que se irán guardando notas acerca de cómo realiza las tareas referentes a la supervisión del proceso en el cual está involucrado, para luego sacar conclusiones con ellos mismos, a través de entrevistas abiertas<sup>8</sup>. Es en estas primeras entrevistas se conoce cuáles son sus principales necesidades; qué desean ver en el supervisorio a desarrollar –dentro del marco de reglas de diseño que se verán más adelante–, por esta razón se debe hablar lo más claro posible, con un lenguaje fluido y entendible.

#### 2.2.1.1 Análisis de implicados (Stakeholders)

Se puede entender por implicado, aquella persona o grupo de personas que van a verse involucradas por, o en el sistema, de manera directa o indirecta, y las decisiones de las cuales, pueden influenciar en el desarrollo del mismo [14]. Teniendo en cuenta esta definición, el objetivo de esta segunda actividad es descubrir los principales implicados que

<sup>7</sup> La etnografía es una técnica de observación que se puede utilizar para entender los requerimientos sociales y organizacionales. El analista o desarrollador se integra al entorno laboral donde el usuario utilizará el sistema. Observa el trabajo diario y anota las tareas reales del usuario [4] I. Sommerville, *Software engineering*, 9th ed. ed. Boston, Mass. ; London: Pearson, 2011.

<sup>8</sup> esta entrevista preferiblemente se debe realizar en el entorno en el que se está trabajando y desarrollando, con esto se obtiene un acercamiento más fiel a sus necesidades, y por tanto, datos con un grado mayor de fiabilidad.

se verán afectados por el desarrollo del sistema software, partiendo de que el principal implicado es el usuario final.

## 2.2.2 Análisis contextual de las tareas.

Este análisis se realiza teniendo en cuenta todos los datos recogidos en el análisis etnográfico. A nivel industrial para la supervisión de procesos, estaría encaminado al análisis acerca de, en qué contexto (métodos, herramientas, objetos, etc.) los operarios, ingenieros o potenciales usuarios realizan las tareas de supervisión del proceso que se esté llevando a cabo.

## 2.2.3 Objetos.

Son las cosas que intervienen directa o indirectamente en la interacción entre el usuario y el sistema interactivo [15]. El concepto de objeto que el modelo MPlu+a describe es de dos tipos:

- **Conceptuales:** pueden ser gestos, mensajes, passwords, firmas, etc.
- **Físicos:** Bolígrafo, papel, calculadora, reloj, etc.

## 2.2.4 Plataforma

Se relaciona con la plataforma tecnológica escogida para albergar el sistema. Para el caso específico de esta guía se debe tener en cuenta el entorno para el desarrollo de la aplicación Android, además de Servidores OPC y entornos de desarrollo industrial.

En la presente guía se recomienda que antes de definir los objetivos del sistema se realice un primer prototipo de baja fidelidad (Ver sección 2.4) teniendo en cuenta solo los requisitos obtenidos hasta este momento<sup>9</sup>. Esto con el fin de mostrar a los usuarios un panorama (a grandes rasgos) de la aplicación que se desarrollará, para especificar los objetivos de una manera menos dispendiosa.

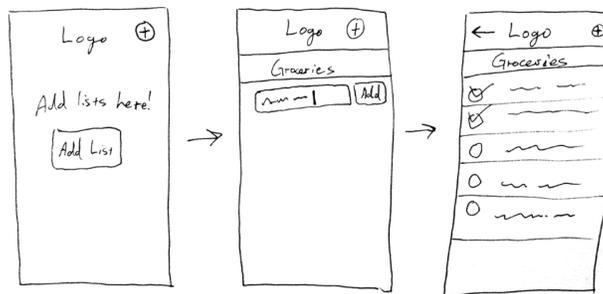


Figura 2. 2. Prototipo de baja fidelidad (Boceto)

[Fuente: <http://blog.teamtreehouse.com/rapidly-prototype-websites>]

En base a las necesidades del usuario se especifican unos objetivos que pueden ser funcionales (lo que el sistema debe hacer), no funcionales (utilización de un determinado

<sup>9</sup> Una de las ventajas de utilizar el modelo MPlu+a es su flexibilidad. Este no determina de una manera concisa en que parte del desarrollo se tiene que aplicar una u otra fase, lo que conlleva a que por ejemplo en este caso, se pueda hacer referencia a la fase de prototipado sin ningún inconveniente, siempre teniendo en cuenta al usuario final del sistema.

lenguaje de programación, de un IDE, etc.), de usabilidad (que el sistema sea fácil de aprender, efectivo y agradable para el usuario) y/o accesibilidad. En el anexo A, se amplía la definición de los objetivos descritos por el MPLu+a.

Es importante tener en cuenta que, aunque los objetivos se definen en esta fase del modelo, a medida que avanza el desarrollo del sistema, pueden ir cambiando debido a las condiciones y exigencias de los usuarios.

## **2.3 Diseño.**

Una vez finalizada la fase de análisis de requisitos se procede a realizar la fase del diseño de la interacción. Esta fase se realiza en repetidas ocasiones tomando en cuenta los requisitos que se obtuvieron en la fase previa. El análisis de estos, proporcionará al equipo desarrollador la información necesaria para llevar a cabo un buen modelo del sistema para su posterior codificación.

El diseño de un sistema interactivo se divide en dos actividades:

- Diseño de la actividad.
- Diseño de la información.

### **2.3.1 Diseño de la actividad.**

Directamente relacionado con la especificación funcional (funciones que debe cumplir el sistema). Se consigue analizando todas las funcionalidades y las tareas necesarias que permiten realizarlas (análisis de las tareas) y el modelado a nivel conceptual buscando una aproximación al modelo mental de los usuarios [15]. Lo que se busca con esta parte del diseño es que se defina de una manera clara un modelo de las tareas que se asemeje en gran medida al modelo mental del usuario. En el anexo A del documento se realiza una ampliación de conceptos como modelo mental del usuario, modelo conceptual y análisis de tareas enfocadas al ambiente industrial.

#### **2.3.1.1 Métodos para el análisis de tareas.**

Los métodos para llevar a cabo el análisis de las tareas se agrupan en función de sus características:

##### **Métodos cognitivos:**

- Identifican secuencias de comportamiento correctas.
- Representan el tipo de conocimiento que debe tener el usuario acerca del uso del sistema.
- Generan una especificación del usuario a partir de la descripción de tareas.

##### **Métodos predictivos:**

- Evalúan el rendimiento humano.
- Describen secuencias de comportamiento.
- Análisis centrado en rutinas de comportamiento.

##### **Modelos descriptivos:**

Permiten obtener una descripción más o menos completa del sistema a partir de la información obtenida de las tareas. En la Tabla 2. 1 se muestran algunos métodos utilizados con sus principales características:

Método	Tipo	Notación	Especificación	Comentarios
HTA	Cognitivo	Gráfico	Semi-Informal	Modelo de descomposición del conocimiento
GOMS	Cognitivo	Textual	Semi-Informal	Familia de lenguajes para describir el conocimiento
UAN	Cognitivo	Gráfico	Semi-Informal	Notación para el estilo de manipulación directa
KLM	Predictivo	Textual	Tiempo	Medición del rendimiento humano
TAG	Predictivo	Textual	Esquemas	Medida de la consistencia
CTT	Descriptivo	Gráfico	Lógica temporal	Herramientas de soporte al análisis y verificación.

*Tabla 2. 1. Algunos métodos utilizados con sus principales características.  
[Fuente: [16]]*

En la presente guía se utilizará el método del análisis jerárquico de las tareas (HTA). Para mayor información acerca de este método, se puede consultar el anexo A.

### **2.3.2 Diseño de la información.**

Esta segunda actividad está enmarcada dentro del diseño de las interfaces, la comunicación visual, la presentación de la información y la tipografía.

Su principal objetivo es dar soporte a la percepción, todo lo relacionado con la parte física de la interacción (la utilización de colores, la organización de los diferentes elementos, tamaño de las imágenes, utilización de metáforas etc.), y el lenguaje (parte visual y auditiva del sistema), todo esto enmarcado dentro de los objetivos de usabilidad y accesibilidad que se tengan [15].

Para llevar a cabo un buen diseño siempre se debe tener en cuenta la relación existente entre el diseño de interfaces de usuario y factores humanos como la percepción<sup>10</sup>, la comprensión, la memoria, etc.<sup>11</sup>

### **2.3.3 Estilo. Estrategias de diseño de la información a nivel industrial.**

Este es quizá uno de los apuntes más relevantes de esta guía, ya que, es en este apartado donde se plasma la relación entre las recomendaciones que existen actualmente para el

<sup>10</sup> La percepción obedece a los estímulos cerebrales logrados a través de los 5 sentidos, vista, olfato, tacto, auditivo y gusto, los cuales dan una realidad física del entorno. Es la capacidad de recibir por medio de todos los sentidos, las imágenes, impresiones o sensaciones para conocer algo.

<sup>11</sup> Estos aspectos relacionados con el diseño de la información se tratan de una manera más amplia en el Anexo A.

diseño de HMIs a nivel industrial, los patrones de diseño del sistema operativo Android y las estrategias de diseño del MPlu+a.

Se debe partir de dos aclaraciones que hace el modelo:

- Suele confundirse el estilo de una interfaz, con el diseño gráfico de la misma; el concepto de interfaz es mucho más amplio, y su diseño gráfico es tan solo una parte de la misma.
- No se debe confundir el diseño de la interfaz de usuario con la propia usabilidad: el estilo de la interfaz es tan solo una de las características de la usabilidad.

### **2.3.3.1 Interfaz gráfica a nivel industrial.**

El propósito de las interfaces graficas a nivel industrial es mantener informado al usuario de lo que está sucediendo en el proceso, para esto, debe existir una buena comunicación entre el usuario y el sistema por medio de la interfaz desarrollada, en sí, que haya una buena interacción.

Una HMI industrial debe contener tanto componentes gráficos, como componentes numéricos, además debe utilizar terminología estandarizada y clara para el usuario final. De ahí que existan muchas recomendaciones para la elaboración de interfaces gráficas para procesos industriales. Se recomienda que las variables de proceso: set point y variable de control sean lo más clara posibles para el usuario: así mismo, se debe mantener un registro histórico de las variaciones ocurridas, esto con el fin de estudiar su comportamiento y poder realizar las predicciones respectivas [17].

La Norma UNE – EN ISO 9241, en su parte 10: principios de dialogo, presenta una serie de ideas que pretenden ser una guía para la elaboración de interfaces graficas en equipos con pantallas de visualización de datos (PVD) [18].

- La aplicación debe estar adaptada a la tarea para la cual se ha diseñado; el dialogo con el usuario debe ser limpio, presentando y exigiendo solamente la información estrictamente necesaria.
- La aplicación debe informar el progreso al interlocutor de forma comprensible para éste (autodescriptividad).
- La aplicación debe poder adaptarse al nivel de capacitación del usuario.
- La aplicación debe ser controlable por el usuario, no al revés.
- Las respuestas de interface deben ser coherentes y adaptadas al nivel de capacitación del usuario.
- Debería ser clara y sencilla de utilizar.

Todas estas recomendaciones se elaboran siempre pensando en el usuario, enfocándose en satisfacer sus necesidades, y pensando en la usabilidad de la interfaz desarrollada.

### **2.3.4 Ventanas informativas.**

En toda interfaz gráfica los cuadros de diálogo o ventanas informativas son importantes para desplegar mensajes importantes, estas ventanas adicionales sirven para:

- Mostrar información específica a los usuarios.
- Recopilar información de los usuarios.

Dentro de los diferentes cuadros de dialogo, se puede encontrar los cuadros de mensaje, que se pueden utilizar para mostrar información contextual y permitir a los usuarios que tomen decisiones con botones. La Figura 2. 3 muestra un cuadro de mensaje el cual hace referencia a información textual, hace una pregunta y proporciona al usuario tres botones para que pueda responder a la pregunta [19].

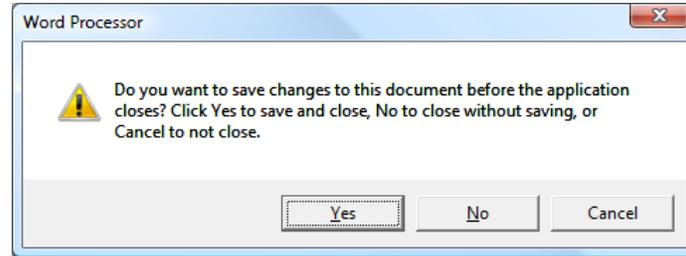


Figura 2. 3. Cuadro de mensaje común en una aplicación de Windows.  
[Fuente: [19]]

Los diálogos en una interfaz industrial son utilizados con gran frecuencia, ya que es una buena modalidad de comunicación con el usuario, estos se hacen a través de cajas de diálogos las cuales solicitan datos o respuestas al usuario. :

Entre las sugerencias más importantes para el manejo de diálogos a nivel industrial se tienen.

- Los diálogos se deben mostrar al operador como consecuencia de una acción de él.
- Los diálogos deben ser concisos y claros.
- Los diálogos se deben situar en una parte de la pantalla donde no interfieran con los datos del proceso.
- Solamente cuando el dialogo es de carácter critico se debe mostrar en el centro de la interfaz.
- Los diálogos de confirmación de las acciones del operador se deben mostrar solamente cuando se trate de operaciones críticas y/o irreversibles.

#### 2.3.4.1 Diálogos en Android.

En el sistema operativo Android estos cuadros de diálogos son llamados diálogos (una pequeña ventana) cuyo objetivo es solicitar al usuario que tome una decisión o introduzca información adicional. Estos diálogos no llenan la pantalla y se utilizan normalmente para los eventos modales<sup>12</sup>, aunque su diseño también puede llegar a ser complejos layouts si el desarrollo de la aplicación lo amerita. En la Figura 2. 4 se puede observar la estructura general de un dialogo.

---

<sup>12</sup> Se habla de una ventana modal si sólo se puede acceder a la ventana que se muestra, y no al resto de las ventanas de la aplicación [62]. En otras palabras, son las cajas que hasta que no se cierran no dejan continuar con la aplicación.



Figura 2. 4. Estructura general de un dialogo en Android.  
 [Fuente: <http://developer.android.com/design/building-blocks/dialogs.html>]

En la parte 1. Se indica el título del dialogo, este puede ser, por ejemplo, algún ajuste que el usuario desea cambiar, o algún requerimiento en el cual se debe tomar una decisión.

La parte 2 indica el área de contenido, los contenidos varían ampliamente entre aplicaciones, se pueden utilizar campos de texto, sliders, Spinners, CheckBoxes, o botones de opciones que permiten al usuario cambiar de aplicaciones o configurar el sistema en algún aspecto. En otros casos, como en las alertas, el contenido consiste en un texto informativo para que el usuario tome una decisión.

La parte 3 hace referencia a los action Buttons (botones de acción). Estos son típicamente Cancel (Cancelar) u Ok (Aceptar). Sin embargo si las opciones consisten en acciones específicas, como continuar, esperar confirmación, anular, editar, etc., entonces se deben utilizar verbos activos. En la Figura 2. 5 se ilustra un ejemplo.

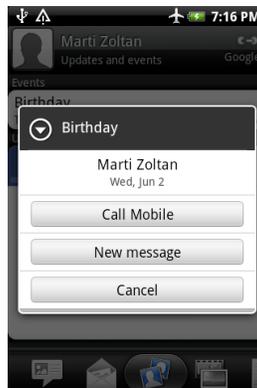


Figura 2. 5. Cuadro de dialogo con acciones específicas.  
 [Fuente: <http://stackoverflow.com/questions/2960646/how-to-create-dialog-with-custom-subtitle-and-some-buttons>].

Se deben considerar las siguientes reglas en las acciones de los botones.

- La acción despectiva de un dialogo siempre está a la izquierda, estas acciones devuelven al usuario a su estado anterior.
- Las acciones afirmativas se encuentran a la derecha, estas acciones siguen el progreso que el usuario desea para cumplir su objetivo.

En la Figura 2. 6 se muestran algunos ejemplos de cuadros de diálogo comunes.

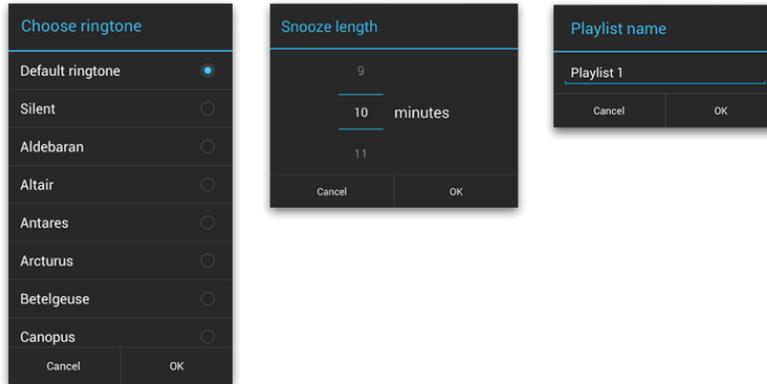


Figura 2. 6. Ejemplos de diálogos típicos utilizados en el sistema operativo Android.  
 [Fuente: <http://developer.android.com/design/building-blocks/dialogs.html>]

Las alertas informan al usuario acerca de una situación que requiere confirmación por parte del usuario antes de continuar. Difieren ligeramente en apariencia basada en la gravedad y el impacto del mensaje transmitido.

Se pueden diseñar **alertas sin título**; las cuales son muy comunes, por lo general la decisión no tiene un impacto grave y puede resumirse en una o dos frases. En el área de contenido se puede hacer una pregunta (como “Desea eliminar ítem?”), o una declaración clara cuya relación con los Action Buttons son evidentes. Ver Figura 2. 7.

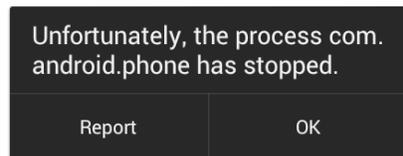


Figura 2. 7. Alerta sin título.  
 [Fuente: <http://developer.android.com/design/building-blocks/dialogs.html>]

Las alertas con título se deben utilizar con moderación. Son apropiadas solo cuando una operación es de alto riesgo e implica la pérdida potencial de datos, conectividad, cargos adicionales, etc., requiere una pregunta clara o declaración (el título) y alguna explicación adicional (en el área de contenido).

Se debe mantener la pregunta o afirmación lo más corta posible: por ejemplo, “Borrar el almacenamiento usb?” (ver Figura 1.5), evitar las disculpas. El usuario debe ser capaz de saltarse el contenido por completo y aun así tener una idea clara de lo que va a hacer basándose en el contenido y en los botones de Acción.

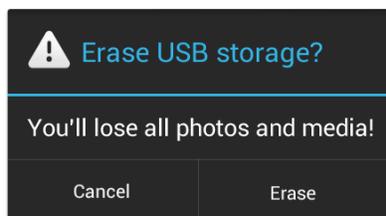


Figura 1. 5. Alerta con título.  
 [Fuente: <http://developer.android.com/design/building-blocks/dialogs.html>]

### 2.3.4.2 Toast.

Un toast es un mensaje que se muestra en pantalla durante unos segundos al usuario para luego volver a desaparecer automáticamente sin requerir ningún tipo de actuación por su parte, y sin recibir el foco en ningún momento (o dicho de otra forma, sin interferir en las acciones que esté realizando el usuario en ese momento). Aunque son personalizables, aparecen por defecto en la parte inferior de la pantalla, sobre un rectángulo gris ligeramente translúcido. Por sus propias características, este tipo de notificaciones son ideales para mostrar mensajes rápidos y sencillos al usuario, pero por el contrario, al no requerir confirmación por su parte, no deberían utilizarse para hacer notificaciones demasiado importantes [20].

Por ejemplo, en la aplicación Wordpress, un blog puede ser eliminado por el usuario, ver figura Figura 2. 8 (izquierda), cuando se confirma la eliminación del blog en el dialogo (centro), un mensaje toast informa que el usuario lo ha eliminado satisfactoriamente (derecha).

A nivel industrial las alertas están al orden del día en toda interfaz gráfica. Siempre que se desarrolle cualquier dialogo es importante que se notifique la acción que se ha realizado por medio de un Toast.

Por ejemplo si se quiere encender o apagar un equipo o planta, es muy importante mostrarle al usuario una alerta, especificando la acción que quiere realizar, Figura 2. 9, seguido de un Toast notificándole que la acción se ha llevado a cabo normalmente y sin contratiempos. Figura 2. 10.

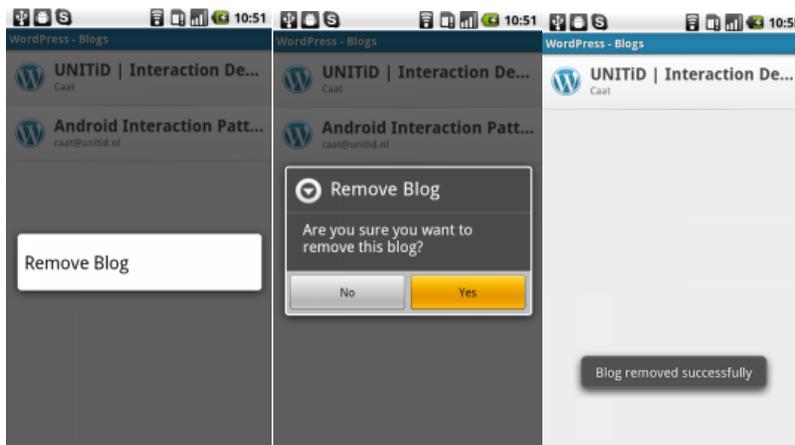


Figura 2. 8. Figura. Mensaje toast en la aplicación WordPress  
[Fuente: [http://www.androidpatterns.com/uap\\_pattern/toast-message](http://www.androidpatterns.com/uap_pattern/toast-message) ]



Figura 2. 9. Alerta por encendido de la planta  
[Fuente: Propia]

En la Figura 2. 9 se observa que además del contenido, el título, y los botones de acción, se introduce un icono de advertencia que es común en muchas aplicaciones, y más a nivel industrial, con esto lo que se pretende es que el usuario centre su atención en que es una alerta importante.

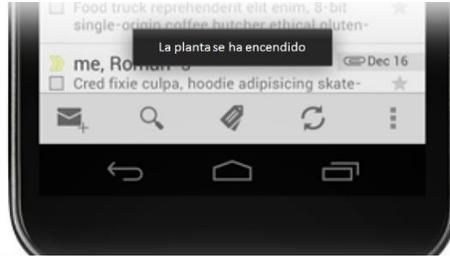


Figura 2. 10. Mensaje Toast para el encendido de un equipo o planta.  
[Fuente: propia]

Los patrones de Android anteriormente citados se pueden integrar con una serie de principios que se deben tomar en cuenta a la hora de diseñar los cuadros de diálogo de la interfaz gráfica industrial de la aplicación a desarrollar:

- Tener un aspecto coherente, que responda a las expectativas del usuario y agilice las respuestas.

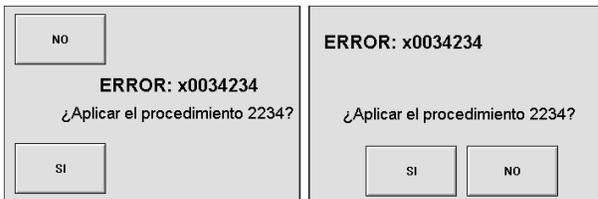


Figura 2. 12. Aspecto coherente en los diálogos a nivel industrial.  
[Fuente: [15]]

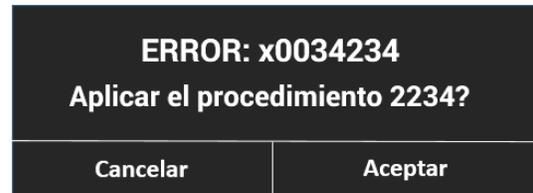


Figura 2. 11. Principio de coherencia.  
[Fuente: propia]

En la Figura 2. 12 (izquierda) se aprecia una distribución de los botones inusual en el cuadro de mensaje, lo cual hace que el mensaje sea chocante, ya que estamos acostumbrados a leer de arriba hacia abajo y de izquierda a derecha. En la Figura 2. 12 (derecha) se aprecia una mejor distribución, lo que generalmente se aprecia en cualquier interfaz. En Android se verá cómo se ilustra en la Figura 2. 11.

- Dar indicaciones con claridad, para proporcionar una asimilación correcta.

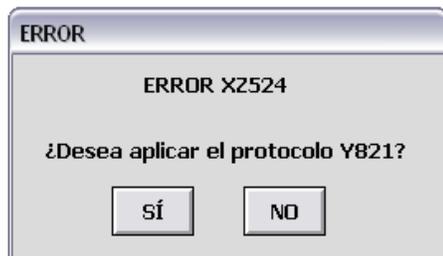


Figura 2. 14. Información poco clara.  
[Fuente: [15]]

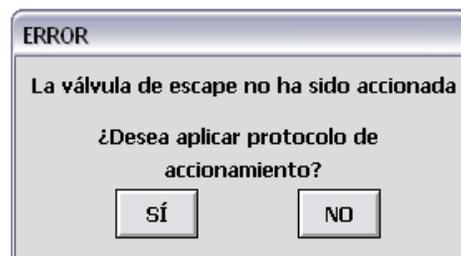


Figura 2. 13. Información clara y concisa  
[Fuente: [15]]

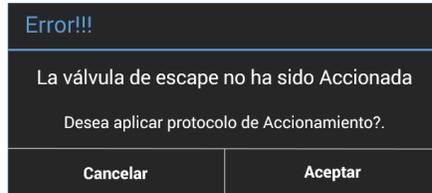


Figura 2. 15. Claridad en el dialogo  
[Fuente: propia]

La de Figura 2. 14, es habitual en muchos procedimientos, se observa que ha pasado algo, pero con información poco clara, cualquier usuario no se atrevería a realizar acción alguna. En la Figura 2. 13 ya se muestra información clara y concisa, apropiada para que el usuario tome la mejor decisión, en la Figura 2. 15 se observa la adaptación en Android.

- Ser concisa, evitar informaciones superfluas. Si se evitan las explicaciones innecesarias, la información será más clara. Ver Figura 2.17



Figura 2. 16. Información Concisa.  
[Fuente: propia].

Es importante resaltar que los diálogos Android de las figuras anteriores, son los que vienen por defecto después de la versión 14 (V14) del sistema operativo (Ice Cream Sandwich), en este caso en particular se muestran diálogos con el tema Holo dark, no obstante, se puede llevar a cabo la personalización de estilos y temas para diseñar cuadros de dialogo más complejos según lo requiera la aplicación. Ver Figura 2. 17



Figura 2. 17. Dialogo Personalizado en Android.  
[Fuente: Propia].

### 2.3.5 Señalización.

La señalización es una técnica que pretende mediante estímulos indicativos visuales, sonoros olfativos y táctiles, condicionar la actuación de la persona que los recibe, frente a los riesgos, peligros o circunstancias que desean ser resaltados para ser eludidos [21].

La normalización existente clasifica los colores en función de su contenido. Ya que los colores son de gran importancia en el manejo de la seguridad en cualquier ambiente laboral y más en el industrial. El instituto Americano de Normalización (ANSI, American National

Standars Institute) determinó una clasificación de colores en función de su grado de apreciación por el ojo humano. Presentó la siguiente tabla en la cual se ordenan los colores de mayor a menor grado de apreciación [15]:

Orden	Color	Fondo
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

Tabla 2. 2. Niveles de percepción de colores, según ANSI.  
[Fuente: [1]]

Los colores de las posiciones 2, 3, 4 y 7 aplicados de formas determinadas, dan lugar a las denominadas señales de seguridad. La norma NTC 1461 Higiene y Seguridad. Colores y señales de seguridad, es la encargada en Colombia de establecer los criterios para el diseño de las señales de seguridad (forma, color, símbolos, contraste y textos). En la Tabla 2. 3 se presentan los parámetros más importantes a tener en cuenta.

Forma Geométrica Color de seguridad					Descripcion
<b>Rojo, Contraste Blanco</b>	<b>Prohibicion</b>				Color de fondo siempre Blanco. La corona circular y la barra transversal en rojo. El simbolo de seguridad debe ser negro, estar ubicado en el centro y no superponer la barra transversal.
<b>Amarillo, Contraste Negro</b>		<b>Advertencia de Peligro o Prevencion</b>			Color de fondo siempre Amarillo. La banda triangular debe ser negra, y el simbolo de seguridad negro y ubicado en el centro.
<b>Azul, contraste Blanco</b>			<b>Obligacion</b>		Color de fondo azul, el simbolo de seguridad debe ser blanco y estar ubicado en el centro.
<b>Verde, Contraste Blanco</b>				<b>Informacion (Salida de emergencia, Primeros auxilios, situacion de seguridad, direcciones).</b>	Utilizados en equipos de seguridad, rutas de escape, etc. La forma puede ser cuadrada o rectangular, según convenga por la ubicación del simbolo de seguridad o el texto. El simbolo de seguridad debe ser blanco y el color de fondo siempre Verde.

Tabla 2. 3. Parámetros de color y forma para señales de seguridad según Norma NTC 1461.  
[Fuente: propia].

Algunos ejemplos prácticos de la implementación de los parámetros mostrados en la tabla, se visualizan en la siguiente ilustración:



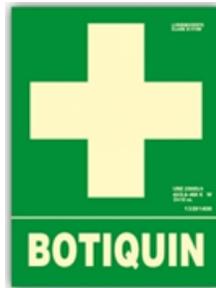
a). Prevención general.



b) Prohibido fumar



c) Obligación, utilizar protectores]



d) Señal Informativa.

[Fuente: <http://www.janfer.com/es/senales-evacuacion/727-senal-botiquin.html>]



Señalización de riesgo Permanente.

[Fuente: propia]

Figura 2. 18. Ejemplos de señales a nivel industrial

### 2.3.5.1 Señalización en conductos.

La norma técnica Colombiana NTC 3458 Higiene y Seguridad. Identificación de tuberías y servicios, especifica los colores para poder identificar los fluidos que se transportan en las tuberías terrestres y en navíos, además de los conductos de ventilación y los servicios eléctricos [4].

Los colores de identificación se ilustran en la Tabla 2. 4.

CONTENIDO DE LA TUBERIA	COLOR
Agua Potable	Verde y azul
Agua	Verde
Aguas residuales	Negro
Vapor	Gris – Plata
Aire	Azul Claro
Aceites (Mineral, Vegetal, Animal) – Combustibles líquidos	Marrón
Agua Sistema contra incendio	Rojo
Gases (en condición gaseosa o licuada) Excepto Aire	Amarillo ocre
Instalaciones eléctricas	Naranja
Red transmisión de datos	Azul oscuro
Conductos de ventilación	Blanco

Tabla 2. 4. Colores de identificación de tuberías Norma Colombiana NTC 3458.

[Fuente: propia]

La norma también hace referencia a que la información se debe suministrar con respecto a los contenidos del tubo, mediante el empleo de las siguientes sentencias, en forma individual o combinada.

- Nombre escrito en mayúscula.
- Abreviatura del nombre.
- Símbolo químico.
- Número del refrigerante (de acuerdo con la norma ISO 817).
- Bandas de color, según el código de indicadores pertinente. (Tabla 2. 4).

#### 2.3.5.1.1 Dirección Flujo.

Cuando es necesario mostrar la dirección del flujo, la norma enfatiza que esta se debe indicar mediante una flecha situada en la proximidad del color de identificación en color blanco o negro según sea necesario, además, es necesario indicar separadamente en los sistemas de calefacción y otros circuitos, el flujo y los tubos de retorno; esto se realiza con el uso de la palabra “Flujo” o la letra “F” sobre un tubo, y la palabra “Retorno” o la letra “R”, en el otro.

En las siguientes figuras se muestran algunos ejemplos aplicado:



Figura 2. 19. Tubería de agua potable.  
[Fuente Propia]



Figura 2. 20. Tubería con riesgo de explosión y fuego.  
[Fuente Propia]



Figura 2. 21. Tubería de gas con radiación ionizante.  
[Fuente Propia]



Figura 2. 22. Tubería de aceite con indicación de dirección de flujo.  
[Fuente propia]

Los parámetros de señalización anteriormente citados se pueden aplicar sin ninguna restricción sobre la aplicación desarrollada bajo el sistema operativo Android. Se debe prestar especial atención en que el diseño ya será personalizado, como el visto en los cuadros de diálogos de las figuras Figura 2. 16. Información Concisa. y Figura 2. 17. El diseño de tuberías se podrá hacer sin ningún contratiempo, al igual que la exportación de

cualquier señal en cualquiera de las ventanas de la aplicación. El desarrollo de Layouts es muy flexible en este sentido. Si lo que se desea es una interfaz modular que sirva para cualquier tipo de dispositivo, una buena herramienta es la utilización de Fragments<sup>13</sup>.

### 2.3.1 Elementos Gráficos.

Para diseñar aplicaciones de visualización industrial se cuenta con una serie de parámetros que son de gran importancia, al igual que para el diseño de interfaces graficas en el sistema operativo Android se cuenta con patrones con igual peso que los anteriores; ya que el objetivo es llevar a cabo una integración de estos parámetros, a continuación se presentan los más importantes enfocados siempre hacia el sistema operativo.

#### 2.3.1.1 Color y forma como fuentes de información.

Los colores sobre las pantallas y los elementos son importantes, se deben establecer los colores adecuados dependiendo del contexto y de los usuarios a los cuales va dirigido. La buena selección de la combinación de colores ayuda a disponer de una interfaz de usuario agradable [15]. Es así que por ejemplo a nivel de procesos industriales las grandes áreas de pantalla vacías deberán rellenarse con colores neutros para no forzar la vista con contrastes excesivos, ver Figura 2. 23. Así mismo es aconsejable que todas las pantallas de una misma categoría tengan el mismo color de fondo, llegando incluso a crear una respuesta automática al usuario [18].

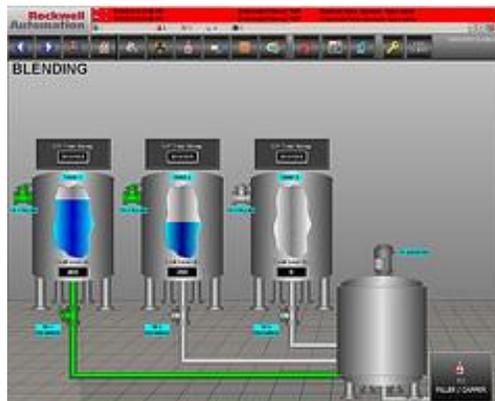


Figura 2. 23. Hmi Rockwell, las grandes áreas vacías tienen un color neutro: gris.  
[Fuente: <http://flickrhivemind.net/Tags/factorytalk,hmi/Recent>]

Teniendo presente lo anterior es importante recalcar que el uso excesivo de colores de muy alto contraste puede convertir una ventana de la aplicación en incómoda o casi ilegible, por ejemplo ciertas combinaciones de tonos azules y verdes pueden dar resultados explosivos [18].

La relevancia del color a nivel industrial hace que se tenga siempre presente en el desarrollo de la interfaz. En general el color en las HMIs industriales es utilizado para:

---

<sup>13</sup> Un fragment es un fragmento (por su tradición al español) de una actividad. Tiene su propio layout y su propio ciclo de vida. Dicho de otra manera, un fragment se puede considerar como una porción de la interfaz de usuario que puede añadirse o eliminarse de esta de forma independiente al resto de elementos de la actividad. Esto es una gran ventaja cuando se quiere diseñar para varias configuraciones de pantallas.

- Representar los estados de los diferentes elementos del proceso (marcha, para, falla, manual, etc).
- Representar los principales materiales y fluidos del proceso (agua, aire, gase, materias primas, productos terminados, etc.).
- Darle color a las alarmas (Criticas, pre alarmas, mensajes, etc.).
- Darle fondo a las pantallas.
- Dar Color de valores de proceso (Temperaturas, presiones, niveles, etc.).

Otras directrices que se deben tener en cuenta en el uso de los colores son las siguientes:

- Limitar el número de colores a cuatro para principiantes y no utilizar más de siete colores para los expertos en una pantalla y asegurarse que estos sean perfectamente diferenciables entre ellos.
- Cuando se combinen colores se debe maximizar el contraste entre ellos.
- Debido a problemas fisiológicos que pudieran tener los operadores respecto a la distinción de colores, reforzar estos con otros elementos: texto, tamaño, forma o posición, cuando sea necesario (evitar entonces las combinaciones de texto y color del tipo texto rojo sobre fondo verde, texto azul sobre fondo amarillo, etc.).
- El color debe usarse para indicar calidad y no cantidad.

Para que el color sea visible, se debe usar en objetos de buen tamaño [22].

Ya que bajo el sistema operativo Android se pueden utilizar todo los colores del espectro, es importante no excederse en la utilización de los mismos. Siempre se deben seguir las recomendaciones anteriormente planteadas. Estas se pueden tener en cuenta en el desarrollo de la interfaz en el dispositivo móvil sin ninguna restricción.

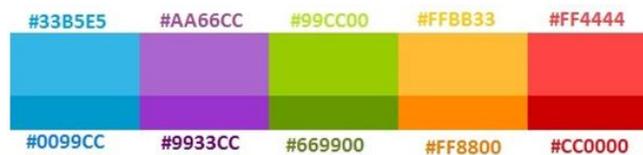


Figura 2. 24. Colores de referencia en Android.  
[Fuente: <http://developer.android.com/design/style/color.html>]

Se debe recordar que los colores en Android se manejan de una forma hexadecimal, partiendo de los colores básicos es sus tonalidades más fuertes, Rojo (#FF0000), Verde (#00FF00) y azul (#0000FF), hasta llegar a la ausencia de los tres colores: el negro (#000000), o la suma de los tres: el blanco (#FFFFFF).

El azul es el color primario estándar en la paleta de colores de Android. Cada color tiene un tono más oscuro correspondiente que se puede utilizar como un complemento cuando sea necesario.

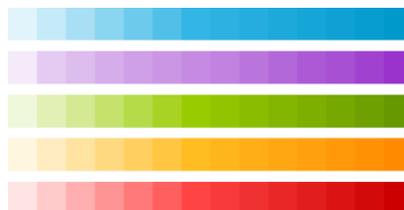


Figura 2. 25. Paleta de colores del sistema operativo Android.  
[Fuente: <http://developer.android.com/design/style/color.html>]

### 2.3.1.2 Los sucesos del sistema.

Las alarmas son los sucesos no deseables y requieren de la atención de un operario para la solución. El resto de situaciones (marcha, paro, cambios de consigna, etc.) serán los llamados eventos, sucesos o estados.

Para las señales de sistemas se acepta la siguiente convención de colores:

ESTADOS	COLOR
Marcha - Abierto	Verde
Parado - Cerrado	Gris
Paro (por falla) - Cerrado (por falla).	Rojo

Tabla 2. 5. Representación por color de los Estados.

[Fuente: [4]]

ALARMAS	COLOR
Atención – Pre-alarma	Amarillo
Alarma	Rojo
Sin Alarma	Gris, Invisible

Tabla 2. 6. Representación por color de las Alarmas.

[Fuente: [4]].

Como no siempre se utilizan los mismos criterios por parte de los diseñadores para representar los estados y las alarmas, es aconsejable siempre utilizar la combinación de elementos para la representación de los mismos. Por ejemplo: Color acompañado de texto aclarativo para que no hayan malas interpretaciones.

En la Figura 2. 26 y Figura 2. 27 se puede observar un ejemplo de lo anterior.

Color (motor en verde = MARCHA) y texto aclarativo.



Figura 2. 26. Representación por color y texto de los estados de un motor.

[Fuente: [4]]

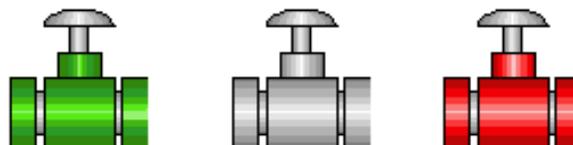


Figura 2. 27. Representación por color de los estados de una válvula

[Fuente: [4]]

Es preciso recordar que en Android se puede manejar todo tipo de animaciones, tanto en 2D como en 3D, sin embargo, para este tipo de aplicaciones industriales el exceso de animaciones podría ocasionar cansancio visual a largo plazo. Las animaciones a utilizar deben ser estrictamente las necesarias [13].

### 2.3.1.2.1 Alarmas.

Las alarmas junto con la representación del proceso y los valores analógicos del sistema constituyen los principales elementos con los que se informa al usuario sobre el estado de la planta. Las alarmas son muy importantes ya que alertan al operador sobre las situaciones anómalas que se presentan en el proceso e implican una intervención de él [22]. Siempre significan un evento o condición anormal, por tanto no pueden ser ignoradas. En general se representan mediante los colores de la Tabla 2. 6 y es conveniente clasificarlas en función del peligro que puedan advertir. No será lo mismo un aviso por un nivel bajo de líquido en un tanque que una presión máxima en un depósito de gas. Las alarmas se pueden dividir en las siguientes categorías.

### 2.3.1.2.2 Pre-alarmas.

Se consideran como alarmas críticas que no afectan la operación del sistema, pero los estados que las provocan son susceptibles de generar graves problemas a corto o mediano plazo, así que requieren atención por parte del personal de mantenimiento. Por lo general y según lo delicado de la máquina o sistema, las pre-alarmas se activan cuando se está cerca de los niveles críticos de operación, por ejemplo, un nivel bajo de aceite, una disminución de velocidad, un aumento de temperatura [4]. En Android estas se pueden representar de diferentes maneras, lo ideal es mostrar un dialogo personalizado con una alerta como la que se muestra en la Figura 2. 28, en ella se puede aclarar la clase de pre-alarma que se está presentando.



Figura 2. 28. Pre-alarma para la velocidad de un motor.  
[Fuente: Propia]

Estas son señales de falla crítica y se considera que una vez se presenten provocan inmediatamente el paro del equipo, sin importar el punto en que se halle el proceso. Se consideran alarmas mantenidas, esto significa que una vez presentes, permanecerán activas hasta que se elimine la causa de la falla. Por ejemplo niveles máximos de presión y temperatura activan alarmas de paro, al igual que el accionamiento de un paro de emergencia [4].

Al igual que las pre-alarmas, un dialogo es lo ideal para representar una alarma en Android. Ver Figura 2. 29. Alarma para la velocidad de un motor.



Figura 2. 29. Alarma para la velocidad de un motor.  
[Fuente: Propia]

Las alarmas se pueden dividir en dos grupos: alarmas de señales Analógicas y alarmas de señales Discretas.

**Alarmas de señales Analógicas:** Es generada cuando el valor de la señal excede su rango normal de operación sobrepasando sus límites. Dichos límites son constituidos por valores que pueden ser constantes o variables, información la cual se encuentra almacenada en la RTU.

En la Figura 2. 30 se puede observar los cuatro límites posibles de una señal analógica.

Se presentan casos en que el valor de una señal oscila cerca de un límite de alarma provocando que entre y salga del estado de alarma repetidamente, ocasionando al operador molestias innecesarias. Para evitar lo anterior, se coloca un rango dentro del cual estas variaciones no son consideradas como alarmas. Esto es conocido como banda muerta, estableciendo ésta según la condición del instrumento [23].

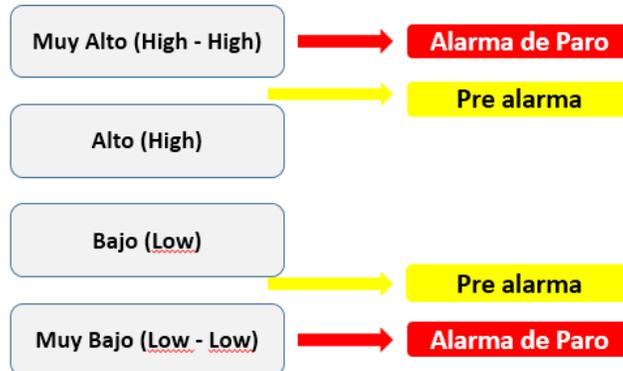


Figura 2. 30. Límites para las alarmas de señales analógicas.  
[Fuente: propia]

**Alarmas de señales discretas:** Estas alarmas representan un estado, tal como Abierto – cerrado, Verdadero – Falso, On – Off, etc.

**Prioridad de las alarmas:** Un último aspecto a tener en cuenta es la priorización de las alarmas, esto debido a que no todas las situaciones de alarma son igualmente importantes. Para determinar la prioridad apropiada para cada alarma, se deben combinar dos factores: la gravedad de las consecuencias si no se llegan a tomar medidas y cuanto es el tiempo disponible para que el operador pueda resolver y evitar esas consecuencias [24].

En la Tabla 2. 7 se ilustra los niveles de prioridad de las alarmas.

PRIORIDAD	DESCRIPCION
Critica	Indica peligro a equipo y/o personal. Se produce acompañada de la alarma audible
No Critica	Requiere que el operador tome la acción, pero no tiene que ser inmediata. (Prioridad Media).
Guía del operador	Suministra información al operador (Prioridad Baja)
Evento	Suministra información sobre eventos de baja prioridad. (Sin prioridad).

*Tabla 2. 7. Niveles de prioridad de las alarmas. Dependiendo de la prioridad, el mensaje de la alarma aparecerá en pantalla con colores diferentes.*

*[Fuente: [18]]*

Las señales de alarma informan de manera más eficiente si se organizan por grupos y se sitúan en lugares fácilmente visibles (lugares de privilegio, parte superior de la pantalla).

Si no se desea cargar el diseño, debería ser bien visible, en todas las pantallas, por lo menos, un indicador de estado de alarma activa con un acceso rápido a una pantalla con detalles sobre esta. Más adelante se verá más a fondo esta recomendación.

### **2.3.1.3 Fondo de Pantalla.**

Muchas aplicaciones desarrolladas sobre el sistema operativo Android, tienen un gran peso estético, esto con el fin de que el usuario se sienta a gusto y atraído con la aplicación; los fondos de pantalla por lo general son animados, con colores exuberantes, o con muchas figuras, siempre cuentan, por lo general, con un gran toque creativo. Un buen ejemplo de esto, son las aplicaciones sobre el tiempo atmosférico, que cuentan con interesantes fondos animados. Ver Figura 2. 31. No obstante, a nivel industrial el asunto cambia completamente, lo principal aquí, es la ergonomía; el descanso visual del usuario. Es por esto que, en las interfaces gráficas a color que se utilizan en procesos industriales, se manejan tonalidades claras de grises, azules, beige, y arena para los fondos o grandes áreas vacías, de esta manera el sinóptico (representación de la planta o proceso) hará un buen contraste y se percibirá mejor. Un buen ejemplo de esto es el color gris claro que se asocia fácilmente con las aplicaciones informáticas, porque la gran mayoría de las interfaces graficas son de este color o lo contienen. Es recomendable no usar blanco o negro en los fondos de las aplicaciones ya que son colores que generan mucho resplandor, causando fatiga visual [18].

El fondo para las aplicaciones industriales en dispositivos móviles no debe variar mucho respecto a los parámetros planteados anteriormente, se deben utilizar los colores neutros y degradados si se desea. Ya que en las interfaces gráficas de los dispositivos móviles se tiene que aprovechar al máximo el espacio, las áreas vacías serán pocas, a diferencia de las interfaces de escritorio.



Figura 2. 31. Fondo de la aplicación sobre el estado del tiempo WeatherPro  
 [Fuente: <http://www.xatakandroid.com/aplicaciones-android/los-mejores-gestores-de-wallpapers-para-android/>]

Algunas directrices adicionales que hay que tener en cuenta son las siguientes.

- Los colores de fondo deben ser contrastantes con los demás elementos.
- El uso de diferentes colores de fondo puede ser utilizado para diferenciar o agrupar procesos o áreas de la planta.
- Evitar el uso de colores primarios o fuertes en zonas grandes de la pantalla.

Algunos ejemplos de fondos de pantalla a nivel industrial en dispositivos móviles se ilustran en la Figura 2. 32.

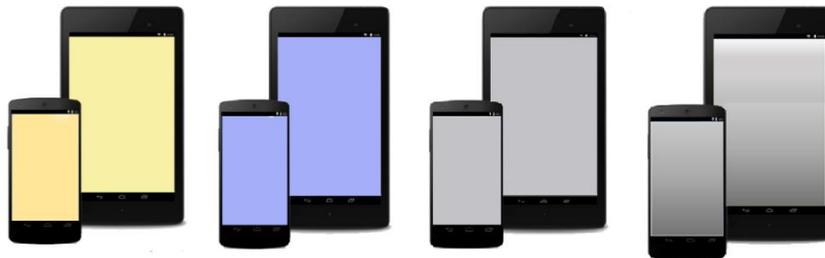


Figura 2. 32. Fondos de pantalla recomendados a nivel industrial para Android.  
 [Fuente: propia]

Otra estrategia que no es tan utilizada es colocar fotografías reales del proceso como fondo de pantalla, lo cual, no es aconsejable ya que solo contribuyen a llenar de colores la misma y hacer el diseño gráfico más difícil. Costará lograr un contraste adecuado para que los elementos que se coloquen encima de la fotografía sean bien visibles. Ver Figura 2. 33 y Figura 2. 34

“Se deberían poner fotografías si es absolutamente necesario y si estas son de alta calidad” [22].

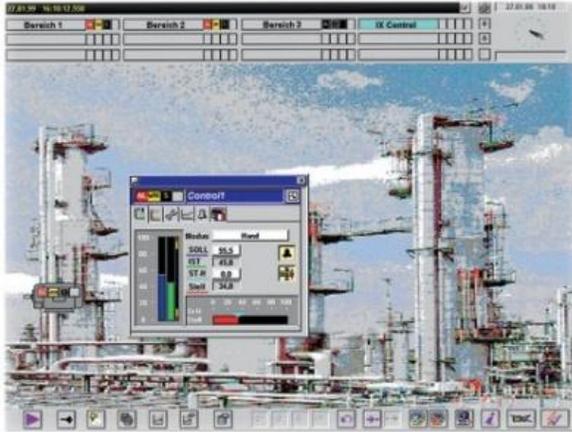


Figura 2. 34. Fondo real con superposiciones  
[Fuente: [18]]

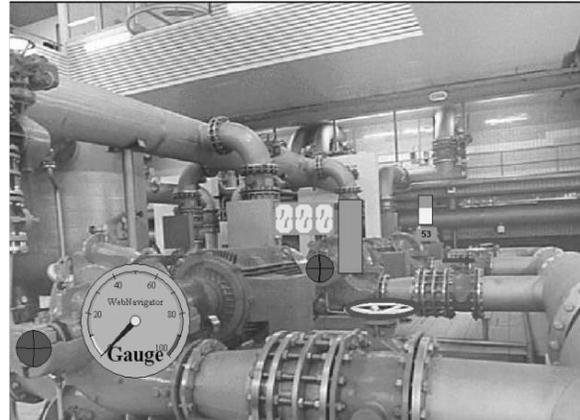


Figura 2. 33. Fondo real con superposiciones  
(WinCC, Siemens).  
[Fuente: [22]]

#### 2.3.1.4 Ubicación de Elementos.

En las interfaces gráficas industriales es muy común que se siga el diagrama de Gutenberg (o regla de Gutenberg) el cual describe algo llamado: la gravedad de la lectura. Este fenómeno consiste en un hábito de lectura que se da en el mundo occidental: de izquierda a derecha y de arriba hacia abajo. El diagrama de Gutenberg divide la página en cuatro cuadrantes: “Área óptica primaria” en el superior izquierdo, “Sector en barbecho fuerte”, el superior derecho; “Sector en barbecho débil”, el inferior izquierdo, y “Área terminal” el inferior derecho [25].

Para este patrón de diseño las zonas que presentan menor atención son la esquina superior derecha e inferior izquierda, y en caso de querer atraer atención a esas partes es necesario incluir un elemento llamativo que enfatice de alguna manera la zona. Es por esto que el contenido relevante debe seguir la diagonal que va de la esquina superior izquierda a la inferior derecha.

El patrón Z se encarga de definir el diseño de una interfaz en base a la forma Z. Los usuarios empezarán a percibir la información de la esquina superior izquierda, moviéndose horizontalmente hacia la esquina superior derecha, después de manera diagonal irán hasta la parte inferior izquierda para finalmente terminar con la parte inferior derecha [26]. Ver Figura 2. 35.

El área terminal 4 es el lugar donde se pueden llevar a cabo acciones. Por ejemplo menús de navegación [27].

Con lo anterior se puede tener una idea clara de cómo se pueden distribuir las zonas de influencia en la pantalla. Los elementos importantes, tales como alarmas o estados operativos, deben gozar de posiciones privilegiadas en pantalla, en los márgenes superior o inferior preferiblemente.

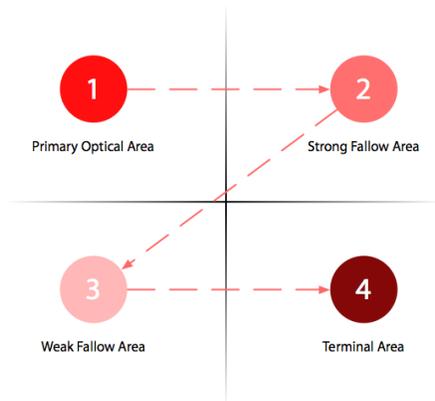


Figura 2. 35. Patrón Z, Regla de Gutenberg.  
[Fuente: [27]].

Teniendo en cuenta el principio de Gutenberg a nivel industrial, una interfaz convencional quedaría conformada de la siguiente manera.

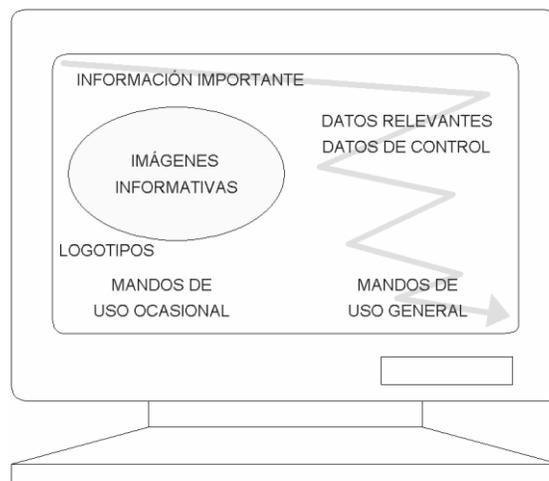


Figura 2. 36. Zonas de pantalla Recomendadas.  
[Fuente: [18]]

En [22] se describen una serie de directrices para realizar la distribución de los elementos en la pantalla.

- Considerar el diagrama de Gutenberg.
- La información importante debe ir arriba.
- El centro de la pantalla es un lugar de alta visibilidad.
- La información miscelánea (Escudos, logotipos, etc.) debe ir en la parte inferior izquierda.
- Funciones e información crítica deben tener un lugar fijo en la pantalla.
- La mejor posición para los gráficos es a la izquierda del campo visual.
- Se debe establecer una estructura de rejilla (grid) regular.
- Al desarrollar los prototipos de los sinópticos de procesos se debe controlar la densidad de los gráficos, la cual no debe sobrepasar el 50%, esto con el fin que no tome una apariencia aglutinada. (importante en dispositivos móviles).
- Se debe dar preferencia a las distribuciones simples sobre las complejas.

Las anteriores recomendaciones se pueden tomar tal cual para el desarrollo de la interfaz industrial en el dispositivo móvil a utilizar. En este se debe siempre tener en cuenta la manipulación por parte del usuario. Es importante que el diseño del layout satisfaga la postura con la que se maneja el dispositivo, ya sea un Smartphone (su manipulación por lo general se realiza con una mano), o una Tablet (Se utilizan las dos manos).

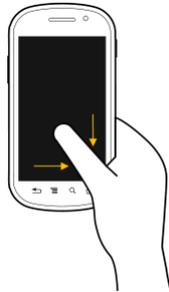


Figura 2. 38. Postura común para manejar un Smartphone en Portrait.  
[Fuente: propia]

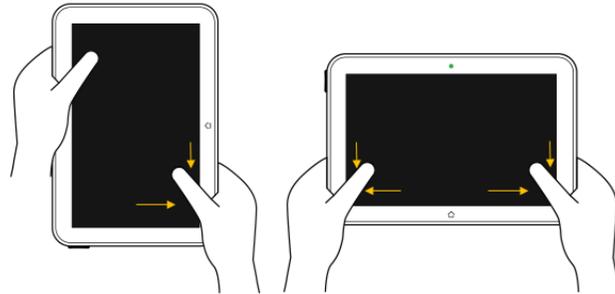


Figura 2. 37. Postura común para manejar una Tablet en Portrait y Landscape.  
[Fuente: propia]

En la Figura 2. 37 y Figura 2. 38 se puede observar que los campos de acción deben estar situados en la parte inferior del dispositivo, esto debido a las posturas que toma el usuario para su manipulación. Es así que, por ejemplo, los submenús para ir a alarmas o a las gráficas del proceso deben ir ubicados en este lugar, ya que el usuario podrá acceder a ellos rápidamente, conservando así, la ergonomía. Ponsa y Granollers proponen una distribución de pantalla como la ilustrada en la Figura 2. 39.



Figura 2. 39. Distribución de pantalla propuesta en [22]  
[Fuente: [22]]

Se observa un espacio amplio destinado para la entrada de comandos y datos por parte del operario, para que el sistema cumpla los objetivos establecidos. Las características principales que deben tener los comandos son su visibilidad y su facilidad de operación, para esto es imprescindible que su área de acción en pantalla sea de buen tamaño, perfectamente etiquetada y por ello reconocible fácilmente por el usuario. La ubicación de los comandos e ingreso de datos se puede establecer ya sea en pantallas específicamente diseñadas para este fin, o pueden ser localizados junto con los sinópticos del proceso tal como se muestra en la Figura 2. 39. En la presente guía de diseño se ha establecido que en la orientación portrait se especifique una ventana para la entrada de los datos de configuración.

Las directrices para especificar los comandos del operador se mencionan a continuación:

- Los comandos deben ser claramente visibles para el operador.
- Los comandos deben ser claramente etiquetados.
- El área de acción sobre el comando deber ser suficientemente grande como para que sea fácilmente operado.
- El operador debe ser realimentado inmediatamente del resultado de su acción.
- Los comandos que activan una acción crítica o de riesgo deben estar claramente etiquetadas y no deben estar cerca de comandos de uso frecuente.

Teniendo en cuenta el principio de Gutenberg (Figura 2. 35), las posturas comunes para el manejo de Smartphones y Tablets (Figura 2. 37 y Figura 2. 38) y la distribución de pantalla propuesta por Ponsa y Granollers (Figura 2. 39), se plantea la distribución de pantalla en los dispositivos móviles (Figura 2. 40) para las orientaciones landscape y portrait.



*Figura 2. 40. Distribución de pantalla para un dispositivo móvil en orientación landscape (izquierda) y portrait (derecha). En esta última orientación, la entrada de los datos se realiza en otra ventana. [Fuente: propia]*

Es importante aclarar que en la presente guía se recomienda que el desarrollo se lleve a cabo en dispositivos móviles con pantallas touchscreen mayores a 3.7 pulgadas. Esto debido a que si se toman tamaños menores al recomendado no se visualizara de una forma clara todos los elementos que a nivel de interfaces industriales se manejan (sinópticos, datos legibles, graficas, botones, etc.)

En la distribución de la Figura 2. 40 se contemplan aspectos importantes como el aprovechamiento del espacio: sinóptico-mímico del proceso, el modo de la distribución de

las alarmas y algunos comandos. Por ejemplo, en esta distribución se puede observar que el título de la pantalla se sitúa en el Action Bar de la aplicación.

#### 2.3.1.4.1 Action Bar

En Android el Action Bar (o barra de Acción), es la barra superior que se encuentra comúnmente en muchas de las aplicaciones que se utilizan día a día; según el grado de complejidad de la aplicación se le puede dar un mayor o menor uso. Por lo general se organiza como se muestra en la Figura 2. 41.



Figura 2. 41. Action Bar en el tema Holo light.  
[Fuente: [13]]

En el primer segmento se ubica el icono de la aplicación, que puede ser un icono predeterminado, un logotipo o una marca si se desea. El cursor que se encuentra a la izquierda del icono, es utilizado para navegar dentro de la aplicación basándose entre las relaciones jerárquicas entre pantallas [13]. Por ejemplo si se tiene una pantalla principal A (Figura 2. 43 - centro) que muestra una serie de elementos y la selección de uno de ellos conduce a la pantalla B (Figura 2. 43 - Izquierda), la cual muestra los detalles de ese elemento, la pantalla B deberá ofrecer el cursor de navegación en el action Bar, al igual que el convencional botón de “atrás” ver Figura 2. 43.



Figura 2. 42. Cursor de navegación del Action Bar (Izquierda) y Botón de Atrás (Derecha).  
[Fuente: [13]]

En la Figura 2. 43 se ilustra un ejemplo de navegación en la aplicación Gmail.

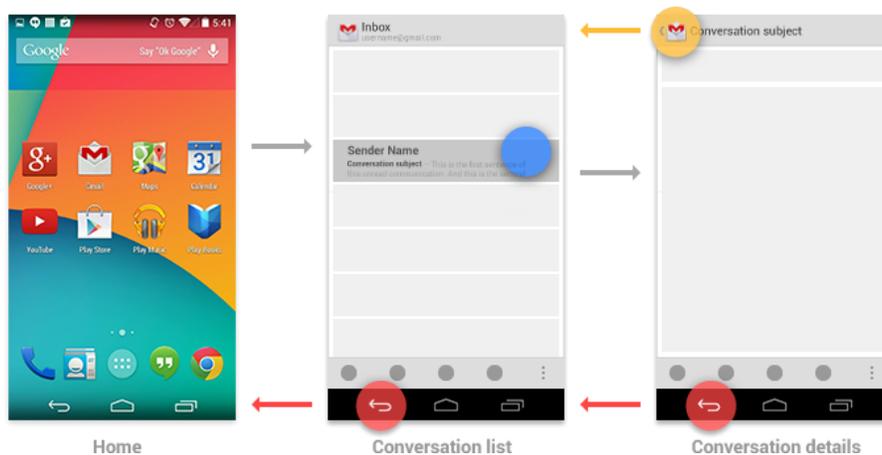


Figura 2. 43. Navegación en la aplicación Gmail. 3 pantallas: Home del dispositivo móvil (Izquierda), Lista de conversaciones (centro) y detalles de conversación.  
[Fuente: [13]]

Nótese que el botón de navegación del Action Bar en la lista de conversaciones de Gmail no está (Figura 2. 43 - centro), ya que se encuentra en la pantalla principal de la aplicación; al indicar un elemento (una conversación) se traslada a detalles de la conversación. En este punto el cursor de navegación del action bar (en el círculo amarillo) se muestra con el objetivo de que el usuario pueda volver a la pantalla principal. De manera similar el botón “Back” (en el círculo rojo) puede también cumplir esta función, con la diferencia de que con este, además de poder navegar a través de la aplicación, se puede salir de la misma, trasladándose a la pantalla principal del dispositivo móvil.

En segundo lugar en el action bar (Figura 2. 41) se encuentra un View Control, este es utilizado si se cuenta con muchas vistas para que el usuario pueda cambiar entre ellas si desea. Un ejemplo es el Spinner. (Ver Figura 2. 44 ).

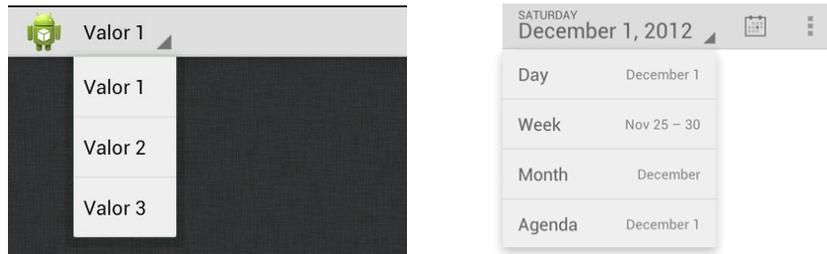


Figura 2. 44. Ejemplos de un Spinner en un Action Bar.  
[Fuente: [13]]

Si la aplicación no necesita un View control simplemente se coloca el título de la aplicación o la información que se desee.

En la tercera posición (Figura 2. 41) se encuentran los Action Buttons (o botones de Acción). En este espacio se pueden poner los botones que se crean más importantes para la aplicación.

En la última posición se encuentra el Boton Action Overflow, éste proporciona acceso a más acciones de la aplicación (Menú de opciones). Es importante tener en cuenta que este icono no aparece en los teléfonos que cuentan con una tecla hardware de “Menú” [13].

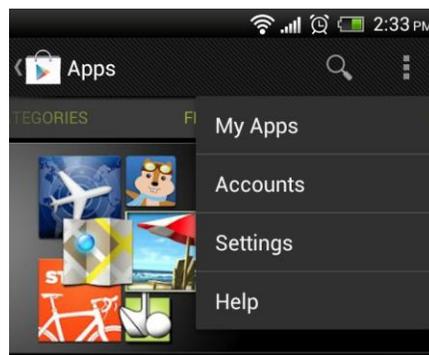


Figura 2. 45. Icono Action Overflow. Se observa que se encuentra siempre a la derecha del Action Bar, en la imagen se despliega un Menú con varias acciones.

[Fuente: <http://www.xda-developers.com/android/remove-unnecessary-overflow-buttons-with-nooverflow/>].

El número de items que se pueden ubicar en el Action Bar viene dado por las siguientes directrices.

- Los Action Buttons no pueden ocupar más del 50% del Action Bar.
- El ancho de la pantalla en pixeles independientes de la densidad (dp) determina el número de ítems que se pueden ubicar en el Action Bar [13].
  - Pantallas más pequeñas que 360 dp = 2 iconos
  - Pantallas entre 360 y 499 dp = 3 iconos.
  - Pantallas entre 500 y 599 dp = 4 iconos.
  - Pantallas desde 600 dp en adelante = 5 iconos [13].

En la tabla se observan algunos dispositivos con el numero de ítems que se pueden colocar en el Action Bar [13].

Dispositivo	Orientación	Dp. Horizontal	Iconos	Ejemplo
Nexus S	Portrait	320	2	OO
Galaxy Nexus	Portrait	360	3	OO=
Nexus S	Landscape	534	4	OOOO
Tablet 7"	Portrait	600	5	OOOO=
Galaxy Nexus	Landscape	640	5	OOOO=
Tablet 10"	Portrait	800	5	OOOO=
Tablet 7"	Landscape	1024	5	OOOO=
Tablet 10"	Landscape	1280	5	OOOO=

Tabla 2. 8. Nro. de Ítems para el Action Bar en algunos dispositivos. "O" denota un Ítem del Action Bar, "=" denota un Icono Action Overflow. Portrait, la posición vertical del dispositivo y Landscape la posición horizontal.

[Fuente: [13]]

Es importante recordar que el Action Bar se introdujo por primera vez en la versión 3.0 de Android (API 11). Debido a que este componente no se incluyó dentro de la librería de compatibilidad Android – Support no era posible utilizarlo en versiones anteriores a la 3.0. Solo fue hasta julio de 2013 que se introdujo en la revisión 18 de esta librería. A partir de entonces se cuenta con la versión de Action Bar conocida como ActionBarCompat la cual es compatible con cualquier versión Android a partir de la 2.1 Eclair (API 7) [20]. En [13] se explica de una manera detallada el cómo importar la librería Android-support-v7 y utilizarla en el proyecto que se esté realizando si se trabaja en una versión anterior a la 3.0 (Honeycomb).

Otro aspecto que no es menos importante al momento de desarrollar la aplicación, y siguiendo los parámetros de diseño para tratar de ahorrar el mayor espacio posible en pantalla, es el de trabajar con pestañas para la navegación a través de la aplicación; si este es el caso, se debe tener presente que el action bar en orientación portrait luce como en la Figura 2. 46, y en landscape como en la Figura 2. 47.



Figura 2. 46. Manejo de pestañas en orientación portrait.  
[Fuente: propia]



Figura 2. 47. Manejo de Pestañas en orientación Landscape.  
[Fuente: propia]

En landscape (Figura 2. 47), la pantalla es lo suficientemente amplia, lo cual hace que las pestañas se puedan ubicar en el action Bar al lado de los Action Buttons, esto es común en las tablets, mientras que en una pantalla estrecha como la que se muestra en la Figura 2. 46, las pestañas aparecen en una barra separada, conocida como stacked action bar (Barra de acción apilada) [13].

#### 2.3.1.4.2 Cuadro sinóptico.

En este se representa la planta del proceso. Cuando se esté realizando el diseño de este es recomendable ser minimalista, el realismo es aceptable en pequeñas proporciones. Los elementos que se dibujen en pantalla deben tener el propósito de brindar información por sí mismos, a ser posible de tipo cualitativo. (Por ejemplo, amarillo = fuego, rojo = paro, azul = aire, agua, etc.).

Los elementos en pantalla se pueden representar como se deseen. Desde un simple esquema de flujo con etiquetas, hasta una elaborada representación 3D de una planta con todo lujo de detalles, pero el uso excesivo de detalles conducirá a un resultado práctico poco eficiente.

En la Figura 2. 48 se puede observar un diseño aceptable.

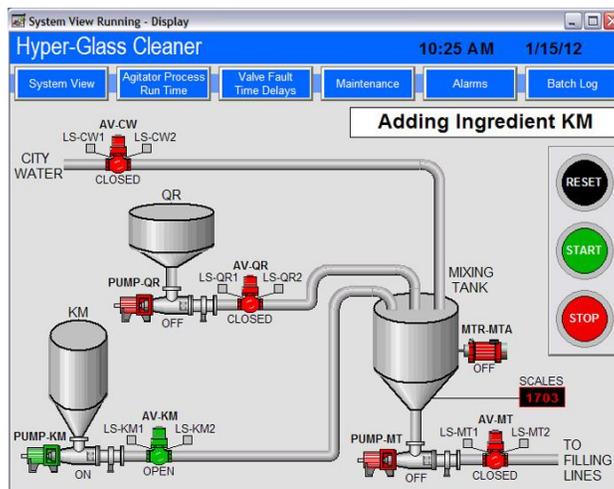


Figura 2. 48. Diseño aceptable. RSView32.  
[Fuente: <http://www.engineer-and-technician.com/how-to-program-rsview32/>]

Los rasgos más destacados son:

- Los colores indican el estado de los equipos (Verde = marcha o abierto, Rojo = parado o cerrado).
- El equilibrio de los colores pretende mostrar una imagen suave, sin brusquedades.
- Cada depósito tiene una identificación numérica y funcional.
- Los puntos de entrada y salida de proceso están convenientemente identificados.
- La distribución en pantalla intenta reflejar la realidad, dentro de lo posible.
- Los botones están agrupados en una zona común.
- Un indicador en zona preferente refleja el estado del sistema (alarmas).
- Las indicaciones numéricas, alineadas, dan la información de forma cómoda.
- Todos los elementos están debidamente etiquetados para evitar ambigüedades.

En la Figura 2. 49 se observa otro diseño semejante pero con un diseño demasiado pobre.

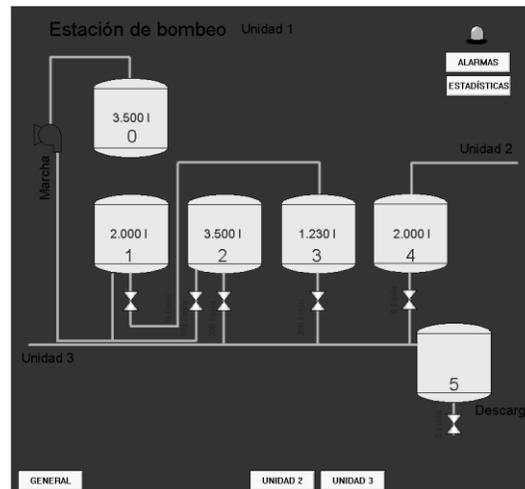


Figura 2. 49. Diseño muy pobre  
[Fuente: [18]]

Sus principales características son:

- En cuanto a los colores, no siguen ningún principio humanitario.
- Los depósitos tienen solamente identificación numérica y son mudos.
- Los puntos de entrada y salida de proceso no son claros.
- Esta distribución parece que intenta reflejar, de forma práctica, la teoría del caos.
- Los botones están en algún sitio de la pantalla.
- Las indicaciones numéricas no son asimilables a primera vista. [18].

Respecto al tratamiento de imágenes para el desarrollo de cualquier aplicación en Android, se puede decir que el sistema cuenta con una gran variedad de potentes APIs para el procesamiento de animaciones en la interfaz de usuario, además del dibujo personalizado en 2D y los gráficos en 3D. Debido a esto, es importante considerar exactamente cuáles son las necesidades gráficas para el desarrollo de la aplicación, teniendo en cuenta las directrices industriales que se mencionaron anteriormente. (Se debe recordar que aunque en Android se pueden hacer grandes diseños cubriendo así cualquier necesidad grafica de una aplicación moderna, abusar de estas herramientas a nivel industrial es muy poco viable).

Android es compatible con varios formatos de imagen, tales como PNG, JPG, GIF. En el desarrollo de aplicaciones es aconsejable siempre utilizar el formato PNG. Estos recursos siempre se deben definir en la carpeta res/drawable del proyecto. En el sistema también se pueden manipular gráficos vectoriales, realizar animaciones al estilo Flash, trabajar con texto y gráficos en 3D.

Entre las clases más importantes para la creación de gráficos en 2D están:

- **Canvas:** Representa una superficie donde se puede dibujar.
- **Paint:** Permite definir el color, estilo o grosor del trazado de un gráfico vectorial.
- **Path:** Permite definir un trazado a partir de segmentos de rectas y curvas.
- **Drawable:** Abstracción que representa “algo que puede ser dibujado”. Esta es quizás la clase más importante.

Dentro de los drawables se tienen:

- **BitmapDrawable:** Imagen basada en mapa de bits (JPG, GIF, PNG).
- **ShapeDrawable:** Gráfico a partir de primitivas vectoriales (se pueden escalar fácilmente sin perder calidad).
- **LayerDrawable:** array de Drawables que son visualizados por capas.
- **GradientDrawable:** Degradado de color que puede ser usado en botones o fondos.
- **TransitionDrawable:** Interpola una transición entre dos drawables.
- **AnimationDrawable:** Animación frame a frame a partir de una serie de objetos Drawable.
- **AnimationSet:** animación a partir de gráficos vectoriales.

Por último las APIs para el tratamiento de gráficos en 3D se tienen dos alternativas:

- OpenGL.
- Renderscript. [28].

Por último la plataforma Android proporciona cuatro tipos de animaciones para poder desarrollar la aplicación:

- Imágenes GIF animadas. Los GIF animados son ficheros gráficos que contienen varios fotogramas (frames).
- Animaciones fotograma a fotograma. Mediante la clase **AnimationDrawable**, el programador suministra los fotogramas.
- Animaciones de interpolación (tweening). Estas animaciones pueden resolverse con código XML y se aplican a cualquier vista.
- Animaciones en la biblioteca OpenGL. [29].

Como se puede observar el tratamiento de imágenes en Android es muy extenso. Por otra parte, debido a que a nivel industrial el manejo de las imágenes tanto en 2D y 3D, además de las animaciones no debe ser tan avanzado, el desarrollo se puede llevar a cabo teniendo en cuenta la clase **AnimationDrawable**, la cual contiene muchísimas herramientas para satisfacer las necesidades de la interfaz.

#### 2.3.1.4.3 Ingreso de datos y comandos.

De acuerdo a los objetivos del usuario, este le suministra datos al sistema, en general las operaciones que este efectúa son: ejecutar comandos, seleccionar opciones e ingresar

datos de consigna y parámetros del proceso. Las principales características de los comandos son su visibilidad y su facilidad de operación. Para cumplir con estos dos parámetros es imprescindible que su área de acción en la pantalla sea de buen tamaño, con sus respectivas etiquetas, y por tanto fácilmente reconocida por el usuario [18]. Android cuenta con TextFields (campos de texto) los cuales permiten al usuario escribir cualquier texto en su aplicación. Estos pueden ser de una o varias líneas. Entre las propiedades más importantes de este elemento se encuentran:

- Si se toca cualquier área del texto ya sea con el cursor o con el dedo, se muestra automáticamente el teclado (Figura 2. 50).
- Con la propiedad Hint se puede mostrar un texto informativo, el cual desaparece cuando se escribe la información requerida.
- Los Campos de texto se pueden personalizar como se desee (fondo, color, tamaño, Bordes, etc.).
- Además de escribir, los campos de texto permiten otras actividades como la selección de texto, cortar, copiar y pegar.
- Tiene la propiedad de autocompletar.

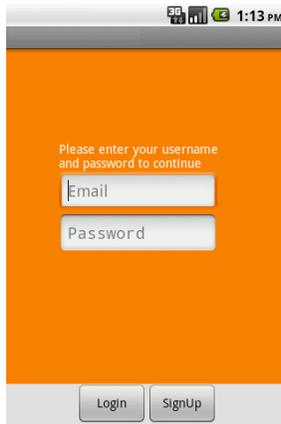


Figura 2. 51. TextFields que cuentan con la propiedad Hint, para ingresar Email y Password.  
[Fuente: [69]]

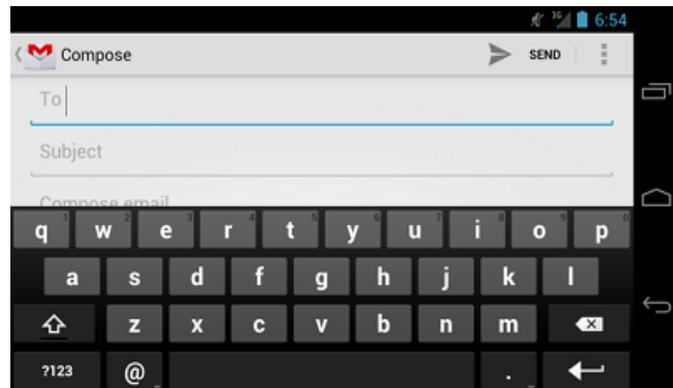


Figura 2. 50. Al hacer contacto sobre los campos de texto el teclado aparece automáticamente.  
[Fuente: [69]]

Otro aspecto importante en los comandos y el ingreso de datos es el de la retroalimentación, lo que implica que el usuario debe recibir una respuesta del sistema ya sea positiva o negativa, inmediatamente después de que ha efectuado una acción. En cuanto a su ubicación, pueden ser localizados junto con los sinópticos siempre y cuando su uso sea muy frecuente, de tal manera que el usuario no tenga que estar cambiando de pantalla a cada momento [22].

Para el ingreso de datos se deben tener en cuenta las siguientes directrices:

- El ingreso de datos debe ser visible, identificable y de tamaño adecuado.
- El operador debe confirmar con el botón de entrada o con un botón de pantalla antes de aceptar el ingreso del o los datos.
- Se debe confirmar al operador que el dato ha sido aceptado por la interfaz.

En [18] se hace referencia a 5 requisitos que deben tener los elementos de dialogo y mando en las HMIs.

**Finalidad:** Principio que dicta que los elementos indicadores o de mando, deben ser claramente visibles e identificables, con etiquetas si es necesario.

El uso de animaciones también puede ser de utilidad con esta clase de mandos, por ejemplo, en el momento de realizar una acción de prendido o apagado, se prendera un led informativo, o también el mismo mando puede dar la sensación de parpadeo para que el usuario sepa en qué estado se encuentra el equipo o proceso. En Android estas animaciones se pueden desarrollar con la clase DrawableAnimation.

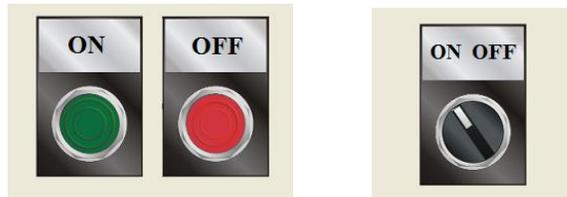


Figura 2. 52. Indicadores de mando convencionales.  
[Fuente: propia]

**Funcionalidad:** Los elementos estarán ubicados de tal manera que se pueda maniobrar con seguridad, sin vacilación, ni pérdida de tiempo y de forma inequívoca.

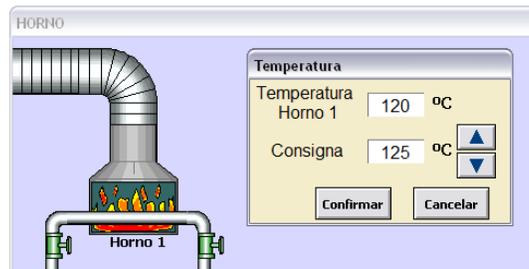


Figura 2. 53. Funcionalidad en indicadores de mando.  
[Editado de: [4]].

**Coherencia:** Cualquier acción de movimiento en pantalla relacionada con indicadores o mando, debería ser fiel a la realidad, siguiendo las ideas intuitivas de escala y dirección. Resultado de esto se tienen dos premisas:

- El arco generado por un elemento rotativo se acopla de manera tangencial, y en la misma dirección, que el elemento de visualización, ver Figura 2. 55.
- Los movimientos lineales de los mandos de control deberían ser siempre en la misma dirección del eje de visualización de la magnitud.

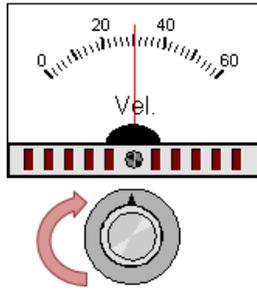


Figura 2. 55. . Movimiento coherente.  
[Fuente: [4]]

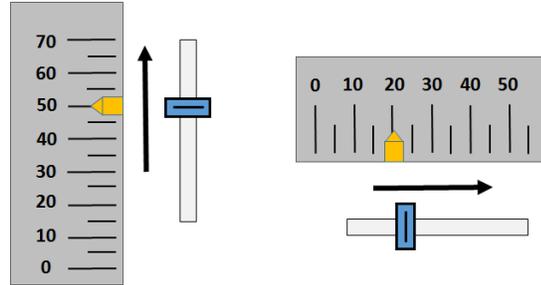


Figura 2. 54. Mandos deslizantes.  
[Fuente: propia].

**Seguridad:** Los elementos de mando nunca se deberán accionar de forma accidental. Por ejemplo si se quiere prender un equipo o una planta del proceso X, se debe mandar un mensaje de confirmación por parte del sistema preguntándole al usuario si efectivamente desea realizar esa acción. En Android esto se puede llevar a cabo mediante el uso de un dialogo como el de la Figura 2. 57 seguido de un Toast mostrando claramente la acción que se realizó. Ver Figura 2. 10.

Otro aspecto importante es el de implementar funciones de supervisión en los mandos de la aplicación, de manera que, ni por descuido, ni voluntariamente, se puedan causar daños o mal funcionamiento al sistema.

Las contraseñas son también una herramienta importante en el tema de seguridad, estas crean niveles de acceso para que sólo el personal autorizado pueda acceder a ciertas funcionalidades del sistema, con el fin de que no se presenten accionamientos imprevistos o accidentales.

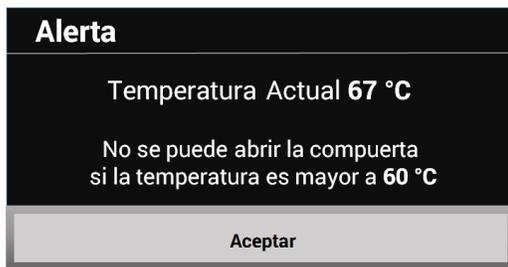


Figura 2. 57. Seguridad en el mando.  
[Fuente: propia]



Figura 2. 56. Protección del mando.  
[Fuente: propia]

En la Figura 2. 56 se observa un dialogo común en Android, cuyo fin es el de pedir una contraseña al usuario a través de un EditText.

**Autodescriptividad:** Cuando se diseñe un elemento de accionamiento para ejecutar varias acciones distintas, (por ejemplo, utilización de teclados, etc.), la acción ordenada debe visualizarse de forma clara y si es necesario, pedir una confirmación [4].

### 2.3.1.4.3.1 Posición de indicadores y mandos.

Los indicadores y los elementos de control permiten establecer una comunicación entre el operador y el sistema: pulsadores, pilotos y visualizadores.

Dentro de un panel o un área de mando se tendrán en cuenta las siguientes directrices para el posicionamiento de los elementos de mando.

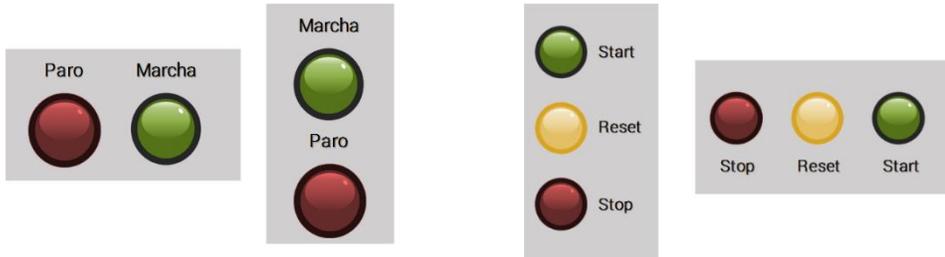


Figura 2. 58. Posición por secuencia.  
[Fuente: propia]

Otra alternativa es la posición por sentido único, que es utilizada cuando varios mandos corresponden a una secuencia de operaciones (Figura 2. 59) Estos deben agruparse y ordenarse siguiendo la secuencia real, de izquierda a derecha, o de abajo hacia arriba (avance, progreso, aumento).



Figura 2. 59. Posición por sentido único.  
[Fuente: propia]

Si los cambios de magnitud son analógicos (progresivos), se definen indicadores de accionamiento siguiendo las reglas establecidas. Ver Figura 2. 60.

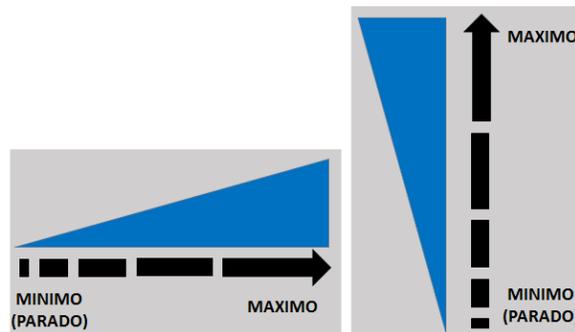


Figura 2. 60. Mandos deslizantes, dos sentidos.  
[Fuente: propia]

Por último, en cuanto a la forma de los indicadores y mandos, se tienen las siguientes convenciones de diseño.

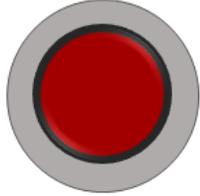
ARRANQUE	PARADA	Pulsador Biestable	Pulsador Mantenido
Puesta en Tensión /ON	Corte de Tensión /OFF	ON/OFF MARCHA /Paro	ON (pulsado) / OFF(libre)
			

Tabla 2. 9. Complementos para elementos de mando.  
[Editado de: [18]]

#### 2.3.1.4.4 Manejo de Alarmas

Otro detalle importante a tener en cuenta es el relacionado con la organización de las alarmas (tema que se trató en 2.3.1.2.1). En la Figura 2. 39 se aprecia que los mensajes de las alarmas se ubican en la parte superior de la pantalla de la interfaz gráfica; siguiendo los patrones de diseño de Android es recomendable que el diseño de las alarmas se lleve a cabo ya sea en una ventana o pestaña diferente, y se realice una lógica en la cual se cuente con un botón de navegación en todas las pestañas, el cual siempre se estará actualizando para mostrar las alarmas que se presenten. Es una buena alternativa para no sobrecargar el diseño de la pantalla y ahorrar el mayor espacio posible en la pantalla principal del proceso en el dispositivo.

En la Figura 2. 61 y Figura 2. 62 se ilustran las tablas de alarmas utilizadas por InTouch e ignition Mobile by inductive Automation, para sus soluciones software.

Alarmas y Eventos					
Date	Time	Type	Priority	Name	Value
10 Noviembre	13:25	HH	1	Temp. Máx.	OFF
10 Noviembre	15:42	LL	750	Vel. Máx.	OFF
12 Noviembre	08:31	H	250	Vel. Máx.	OFF
12 Noviembre	12:16	L	500	Temp. Máx.	OFF

Update Successful      Default Query

Figura 2. 61. Tabla de alarmas y eventos de InTouch.  
[Fuente: propia]

Event Time	Display Path	Event State	Priority	Ack'ed By	Event Value
4/14/14 4:12 PM	Tank 2	Active	Medium		6.8642
4/14/14 4:12 PM	Tank 2	Ack	Medium		
4/14/14 4:12 PM	Day Tank	Clear	Critical		89.9872
4/14/14 4:11 PM	Day Tank	Active	Critical		91.0079
4/14/14 4:11 PM	Day Tank	Ack	Critical		
4/14/14 4:11 PM	Day Tank	Clear	Critical		89.8696
4/14/14 4:11 PM	Day Tank	Active	Critical		90.6295
4/14/14 4:11 PM	Day Tank	Ack	Critical		
4/14/14 4:11 PM	Day Tank	Clear	Critical		89.3132
4/14/14 4:11 PM	Day Tank	Active	Critical		90.0519

Figura 2. 62. Tabla de alarmas y eventos Ignition Mobile.  
[Fuente: propia]

A nivel de interfaz en los dispositivos móviles, para implementar este tipo de distribución (asemejar una tabla) es común utilizar un `TableLayout`, el cual se puede definir como un contenedor de `TableRows` que a su vez contiene `Views` (o vistas), estos `Views` pueden ser imágenes, texto o incluso otra tabla, lo normal en estos casos es insertar `TextViews`. De esta manera el número de filas de la tabla vendrá determinado por el número de `TableRows` y el número de Columnas por el número de `Views` que haya dentro de cada fila, ver Figura 2. 63.

En [13] se habla de que el `TableLayout` no contempla bordes para sus filas, columnas o celdas, lo cual conlleva a un problema si se quiere implementar una tabla con sus características más comunes. Para solucionar este inconveniente la solución más sencilla consiste en dar un color de fondo al `TableLayout` y otro diferente a cada celda, de esta manera se crea la falsa impresión de que la tabla tiene un borde, ver Figura 2. 64. Se debe tener en cuenta que en cada celda hay que dejar un margen, esto se puede hacer con la propiedad `margin` en las propiedades de cada `View`.

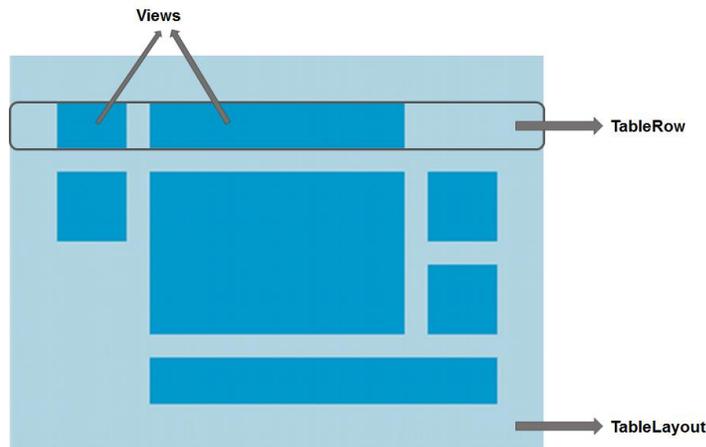


Figura 2. 63. Distribución de `Views` dentro de `TableRows` que a su vez están contenidos en un `TableLayout`.  
[Editado de: [13]]

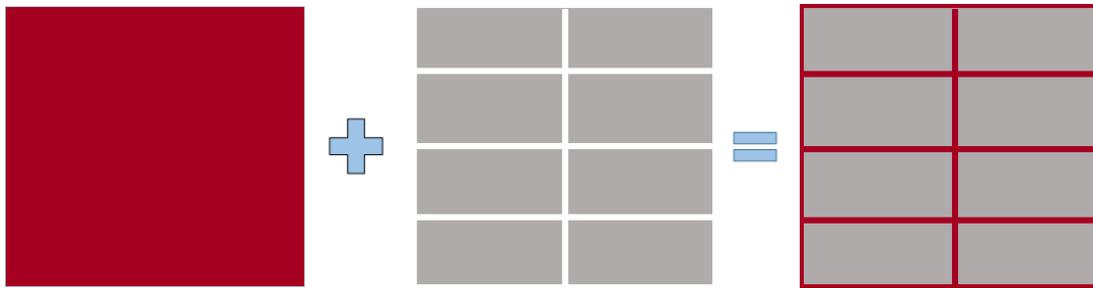


Figura 2. 64. Superposición de varios Views (grises) en un TableLayout (fondo rojo).  
[Fuente: propia]

Para llevar a cabo una buena gestión de las alarmas en la aplicación a desarrollar, la opción más recomendable es implementar una base de datos. Android utiliza el sistema SQLite para gestionar bases de datos, este utiliza el lenguaje SQL (Structure Query Language); un lenguaje de consulta y acceso a bases de datos ampliamente utilizado en muchos sistemas. El paquete que utiliza Android para manipular las bases de datos es **android.database.sqlite** [30].

Para crear la base de datos se pueden utilizar tan solo las clases, que serán suficientes para el diseño de la interfaz. Ya que se van a manejar muchas celdas en una misma pantalla lo recomendable es utilizar un Scroll View horizontal para poder visualizar todos los datos. En la Figura 2. 65, se puede visualizar una tabla de alarmas desarrollada en Android teniendo en cuenta todos los aspectos de desarrollo anteriormente mencionados.

Fecha	Hora	Tipo	Prioridad	Nombre	Valor
3/mar/2012	13:25	HH	1	temperatura maxima	123°C
6/jun/2012	15:27	RR	250	temperatura alta	80°C
4/feb/2013	17:05	TT	500	temperatura normal	25°C
7/mar/2013	8:30	MM	232	temperatura alta	87°C
14/oct/2013	10:30	WW	428	temperatura normal	28°C

Figura 2. 65. Tabla de Alarmas en Android.  
[Fuente: propia]

#### 2.3.1.4.5 Gráficos de Tendencias y Tablas.

Los gráficos de tendencias y las tablas son los principales medios de agrupamiento de las variables para crear esquemas informativos para el usuario. Estos son una buena herramienta para presentar de una forma cómoda la evolución de las variables del sistema. Una de las ventajas de elaborar gráficos de este tipo es que se proveerá al operador de un panorama más completo de la situación de la planta, mostrándole además de los equipos y valores puntuales, los datos que requieren una interpretación a través del tiempo.

Para especificar los gráficos de tendencias se es recomendable:

- No poner más de 9 variables en una gráfica.

- Diferenciar los datos con diferentes colores y tipos de línea.
- Asegurarse que los rangos del grafico sean adecuados para la operación.
- Mostrar el mínimo y máximo para cada variable.
- Usar una rejilla tenue (grid) para ayudar al operador.
- Las divisiones de la rejilla deben ser comunes tales como 1, 2, 5 o 10 unidades.
- Permitir al operador visualizar los valores numéricos de los datos en el tiempo.
- Etiquetar los ejes y puntos graficados.
- Permitir al operador quitar temporalmente algunas de las variables de la tendencia. [22].



Figura 2. 66. Gráfica de tendencias del software Ignition de Inductive Automation [Fuente: <http://iademo.inductiveautomation.com/main/system/mobile>]

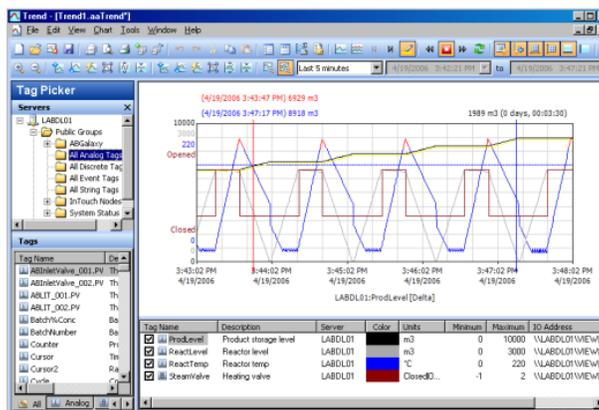


Figura 2. 67. Tabla de tendencias del software Wonderware [Fuente: [http://trainweb.wonderware.com/getstartaf/AFPages/aM01\\_Overview.htm](http://trainweb.wonderware.com/getstartaf/AFPages/aM01_Overview.htm)]

En cuanto al diseño de gráficas en Android la opción más recomendable es utilizar AndroidPlot, el cual es un API para la creación de gráficos dinámicos y estáticos dentro de cualquier aplicación Android. Una de sus principales ventajas es que es compatible para todas las versiones disponibles.

AndroidPlot soporta los siguientes tipos de gráficos:

- Gráficos de líneas.
- Gráficos de barras.
- Gráficos de pastel.
- Gráficos de dispersión.

En la Figura 2. 68 se ilustran algunos de los gráficos que se pueden implementar con AndroidPlot.

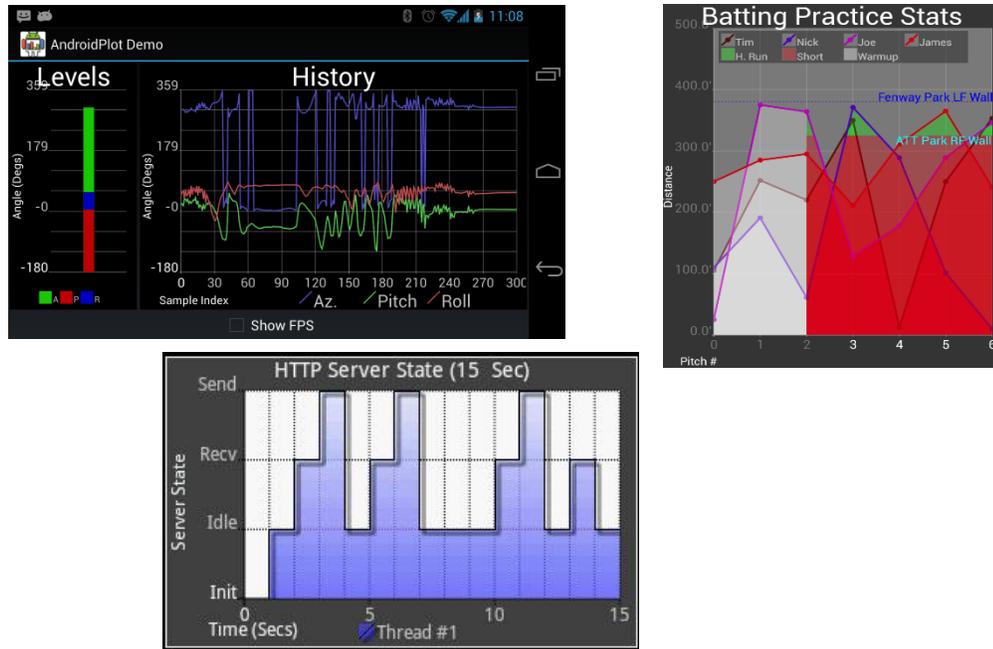


Figura 2. 68. Ejemplos de graficas desarrolladas en AndroidPlot.  
 [Fuente: <http://androidplot.com/> ]

Por otra parte las tablas de datos se utilizan para agrupar información relacionada, en los sinópticos se utilizan principalmente para mostrar resúmenes de diversos tópicos referentes al proceso y su finalidad es la de mostrar un esquema informativo completo al usuario.

Entre las principales recomendaciones para el diseño de tablas se tiene que:

- Es recomendable distinguir las tablas utilizando un color de fondo diferente al de la pantalla.
- Para separar las celdas se debe usar una rejilla (grid) tenue pero distinguible para el usuario.
- Titular claramente la tabla.
- La ubicación de la tabla debe seguir los criterios de posicionamiento según la importancia de la información.
- Si la tabla debe contener una gran cantidad de renglones y columnas, es mejor dedicarle una pantalla de detalle en lugar de ponerla junto a los sinópticos de proceso. [22].

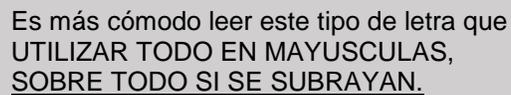
Para el diseño de las tablas en Android, en este caso se siguen los mismos parámetros utilizados para el manejo de tablas de alarmas visto en 2.3.1.4.4.

### 2.3.1.4.6 Información Textual.

La información textual que el usuario recibe de una interfaz es uno de los factores más importantes para una buena comunicación entre este y el sistema, pero esto no quiere decir que se tenga que saturar al usuario con mucha información. Los textos que se presenten no deben ser demasiado grandes ni demasiado pequeños; deben tener una dimensión adecuada al igual que una fuente bien definida y legible [18].

Algunas de las directrices más importantes para la definición de las fuentes son las siguientes:

- No se deben utilizar más de tres fuentes en la interfaz.
- El tamaño de la letra debe ser aquel que permita una lectura cómoda, desde la posición que tenga el usuario [18].
- El tipo de fuente debe facilitar la lectura, por tal es recomendable utilizar fuentes tipo sans – serif que son aquellas sin pequeños remates en sus extremos, esto facilita su lectura en pantallas, algunos ejemplos de este tipo de letra son: Arial, Arial Narrow, Tahoma y Verdana. Si se habla estrictamente de Android también se podría utilizar el tipo de letra Roboto, que es muy común en muchas aplicaciones.
- No utilizar énfasis en el texto (subrayado, itálico, sombreado salvo en casos muy especiales).
- El color del texto debe proporcionar un contraste adecuado con el texto. La combinación de colores con bajo contraste dificulta la lectura de lo escrito, entre la mejor relación de colores se encuentran: negro sobre blanco, negro sobre un tono de gris claro, negro sobre amarillo. En general es aconsejable usar en el fondo colores claros y suaves y siempre un color de texto oscuro [4].
- Evitar al máximo el uso de letras mayúsculas en todas las letras del texto, ya que su uso exagerado causa estrés visual, ver Figura 2. 69.
- Cuando se presenten grandes cantidades de información, se debe aumentar la separación entre líneas o insertar líneas en blanco para mejorar la legibilidad.
- Cuando se muestra información crítica esta debe ser espaciada con suficiencia.



Es más cómodo leer este tipo de letra que  
UTILIZAR TODO EN MAYUSCULAS,  
SOBRE TODO SI SE SUBRAYAN.

*Figura 2. 69. Abuso de mayúsculas y subrayados.  
[Fuente: [18]]*

En la Figura 2. 70 se observan dos pantallas de un proceso de envasado, en una se tienen en cuenta las directrices antes mencionadas (arriba) y en la otra simplemente se diseña sin ninguna recomendación (abajo). Nótese que los textos de los botones son en minúscula y con un buen contraste, que además cumple con los parámetros de colores establecidos. El verde significa marcha y el rojo paro. Además el uso de mayúsculas esta exclusivamente para el título de la planta del proceso, por otro lado, en la imagen inferior no se ha tenido en cuenta ninguna recomendación; el tipo de fuente es inadecuado al igual que el texto de los botones (ilegibles) y su contraste, además, el título de la planta cuenta con un subrayado innecesario.

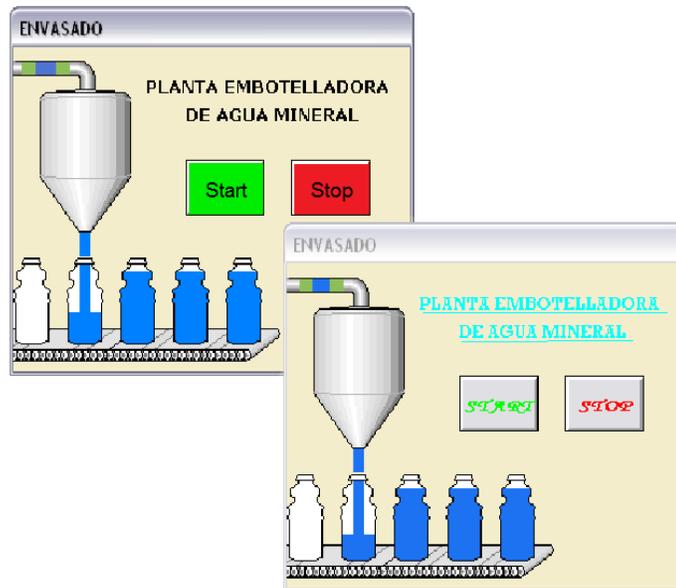


Figura 2. 70. Tipo de fuente y color  
[Editado de: [4]]

El sistema operativo Android utiliza herramientas tipográficas tradicionales como la escala, el espacio, el ritmo y la alineación, para llevar a cabo una implementación exitosa, con el fin de que el usuario pueda comprender rápidamente una pantalla de información textual. Ice Cream Sandwich (API 4.0) introduce un nuevo tipo de familia tipográfica del tipo Sans – serif llamada Roboto, que fue creada específicamente para los requisitos de la interfaz de usuario y las pantallas de alta resolución. Este tipo de familia incluye variantes: delgada, liviana, regular, media, negrita, “black” y condensada [13]. Lo cual hace que se cuente con una poderosa herramienta para cumplir con las directrices anteriormente expuestas.

Se debe recordar que el framework de Android utiliza las siguientes escalas para el tamaño de fuente.

Text Size Micro	12sp
Text Size Small	14sp
Text Size Medium	18sp
Text Size Large	22sp

Figura 2. 71. Escalas para el tamaño de fuente en Android. Se debe recordar que el tamaño de fuente se especifica en sp<sup>14</sup>.

<sup>14</sup> **Sp**: Es similar a la unidad dp, pero en este caso también se escala en función de la preferencia de tamaño de fuente especificada por el usuario. Se recomienda usar esta unidad cuando se especifican tamaños de tipo de letra, de manera que se ajusten tanto a la densidad de la pantalla como a las preferencias del usuario [67].

## 2.4 Prototipado.

Un prototipo está definido como un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas [31].

El prototipo es una herramienta con una gran utilidad ya que conlleva a que el usuario participe activamente en el desarrollo de la interfaz, haciendo que se amplíe y se tenga una visión más concreta de los requisitos de éste para llevar a cabo un desarrollo más confiable; además de esto, se crea un canal de comunicación estable entre el equipo de desarrollo y los usuarios.

El MPIu+a es muy flexible en cuanto a la utilización de una u otra técnica de prototipado. El objetivo del modelo es garantizar el cumplimiento de todos los pasos para disponer de un sistema altamente usable y accesible, por esta razón concede un alto grado de libertad al equipo desarrollador para que decida en que momento debe aplicar las diferentes técnicas y tome sus decisiones. Esto es una gran ventaja ya que se puede utilizar esta técnica en cualquier fase del desarrollo.

El modelo clasifica los prototipos o las técnicas de prototipado, en función del coste y el esfuerzo de realización y de la fidelidad de dichos prototipos respecto al sistema final:

<b>Prototipo de Baja Fidelidad</b>	<b>Prototipo de Alta Fidelidad</b>
Implementaciones genéricas. Permiten abarcar un espectro mayor de la interacción a realizar.	Implementaciones específicas y de mayor detalle de una o varias tareas del sistema.
Caracterizado por ser económico, rápido de construir, rápido de arreglar y no precisa de técnicos expertos. Además que es útil para el diseño general de las interfaces.	Caracterizado por el uso de herramientas especializadas de prototipado, requiere de expertos que conozcan dichas herramientas, por tanto más costoso y requiere de más tiempo.

*Tabla 2. 10. Clasificación de prototipos o técnicas de prototipado.  
[Fuente: [14]]*

Es importante tener en cuenta que un prototipo no será mejor que otro, por el número de funcionalidades o características que se vean representadas en él. Cada tipología resultara más o menos útil en una u otra circunstancia. Siendo el diseñador o el equipo de desarrollo el encargado de determinar qué será mejor para cada situación y evaluación a realizar [14].

En el anexo A se presenta una tabla con las diferentes técnicas de prototipado, relacionando su coste y tiempo de desarrollo, factores influyentes en la elección de una u otra técnica, según las necesidades establecidas. Para efectos del presente documento se explicarán de una manera general los utilizados para la guía propuesta.

Un aspecto muy importante en el diseño de aplicaciones móviles es que los diseñadores necesitan métodos rápidos con los cuales se pueda colaborar, prototipar, evaluar e iterar sus diseños [32]. Dentro de estos métodos el más utilizado es el prototipado en papel (boceto) que puede derivar en un Story Board Navegacional o modelo de navegabilidad. Estas son técnicas de baja fidelidad basadas en la utilización de materiales tan básicos como lápiz, colores, papel y las tijeras. Su finalidad es la creación de prototipos minimalistas, flexibles y enormemente versátiles. Además, permiten probar y refinar un

diseño en repetidas ocasiones y realizar pruebas de usabilidad con el usuario antes de que se pase a realizar cualquier línea de código.

El prototipado en papel (boceto) y story board navegacional (modelo de navegabilidad), son técnicas que consisten en dibujar en un papel, sin entrar en grandes detalles estéticos, las interfaces que se van a probar y valorar. Los diferentes estados de la interfaz se van dibujando en hojas separadas y mediante un proceso de ordenación que el diseñador determina permite al usuario final interactuar con este material simulando el funcionamiento del sistema. El objetivo principal de estos prototipos no es probar lo bonito que es el diseño, sino que trata de verificar si los usuarios son capaces de realizar sus tareas con la interfaz propuesta [15].

Algunas de las principales razones por las que se utiliza estas técnicas son:

- Permiten mapear todas las interfaces de la aplicación en un tiempo muy corto, esto hace que el ahorro para cualquier empresa sea significativo ya que el tiempo de diseñadores y desarrolladores tiene un alto costo.
- Debido a su rápida construcción permite rápidas iteraciones en el diseño antes de que se lleve a cabo el lento trabajo de crear diseños visuales o escribir código.
- Son técnicas de un muy bajo costo, lo cual es beneficioso para el usuario por que no se cohibe en proponer cualquier arreglo en el diseño.
- Permite que todos los miembros del equipo de desarrollo y el usuario final se involucren en el proceso, para alcanzar un óptimo diseño.
- Son prototipos centrados en el usuario. [32].
- No están sujetos a restricciones impuestas por la tecnología -arquitectura del sistema, la base de datos, el ancho de banda, el sistema operativo-, y a pesar de ello ayuda al equipo a anticipar problemas y decisiones derivadas de la tecnología.

En la Figura 2. 72 y Figura 2. 73. se ilustran algunos ejemplos de prototipos en papel para aplicaciones móviles.

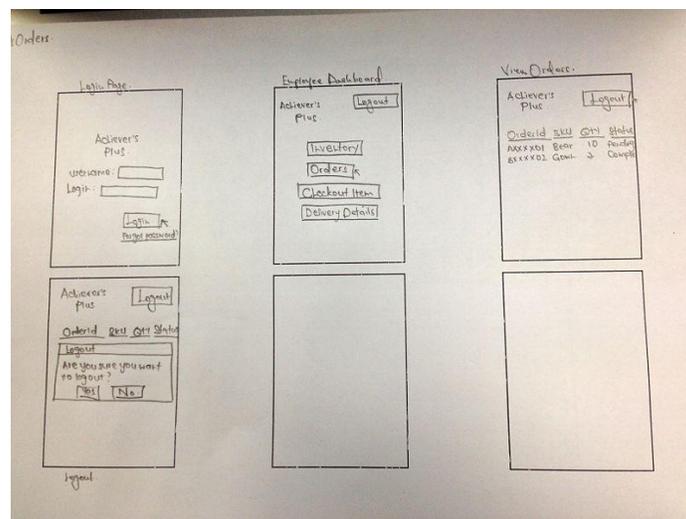


Figura 2. 72. Prototipo de papel (boceto)

[Fuente:

[https://wiki.smu.edu.sg/is480/IS480\\_Team\\_wiki%3A\\_2012T2\\_DaDa\\_Achievers\\_Project\\_Documentation\\_Paper\\_Prototype](https://wiki.smu.edu.sg/is480/IS480_Team_wiki%3A_2012T2_DaDa_Achievers_Project_Documentation_Paper_Prototype) ]

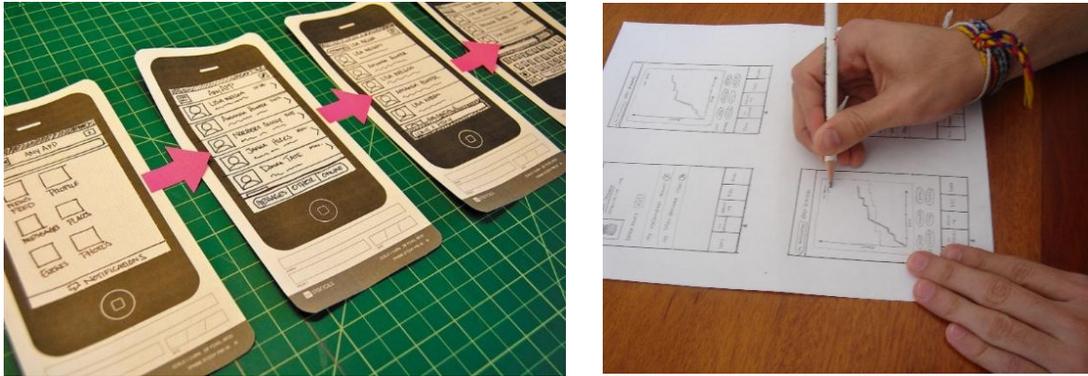


Figura 2. 73. Ejemplos Story board de papel, además de las diferentes pestañas se muestra flechas que significan navegabilidad en la aplicación.  
 [Fuente: <http://www.lucidity.ie/blog/155-idea-to-app-how-to-make-a-mobile-app> ]

Dos aspectos muy importantes que debe mostrar un prototipo de papel son: la **presentación**, que hace referencia a qué elementos proporciona dicha interfaz para la interacción y la **navegación** como información complementaria con diseño claramente perceptible que facilite la movilidad durante la fase de evaluación del prototipo.

Aparte de los prototipos mencionados anteriormente, la presente guía de diseño contempla las **maquetas digitales**, que es una técnica para representar interfaces de la aplicación, principalmente para evaluar el diseño de la información y el **prototipado software** para evaluar la etapa final de la aplicación. Este es un prototipo que reproduce el funcionamiento de una parte importante de las funcionalidades con el objetivo de probar determinados aspectos del sistema final [15]

En cuanto a la navegación en una HMI industrial se debe tener en cuenta la jerarquía de ventanas que se maneja según las directrices de diseño establecidas. Los tipos de pantalla que normalmente son incluidas en una aplicación HMI son las siguientes:

- **Pantallas de proceso:** muestran el estado de los equipos y del proceso en general. Se pueden dividir en general de planta, general de área, de detalle y de equipo. También son conocidas como mímicos o sinópticos de proceso.
- **Pantallas de comandos:** permiten al operador realizar acciones generales tales como arranque/paro de equipos y selecciones diversas.
- **Pantallas de configuración:** Permiten al operador y al ingeniero de proceso establecer los parámetros de configuración del sistema tales como límites de alarmas, sintonización de PID's, calibración, recetas de producción, etc.
- **Pantallas de tendencias:** Se muestran los valores de las variables más importantes del proceso en el tiempo.
- Pantallas de alarmas. [22].

Es importante que el número de pantallas sea el adecuado. Las pantallas de comandos y de configuración se pueden establecer dentro de la ventana del proceso si estas no contienen muchos elementos. En la Figura 2. 74 se ilustra una jerarquía de ventanas básicas para una aplicación HMI.

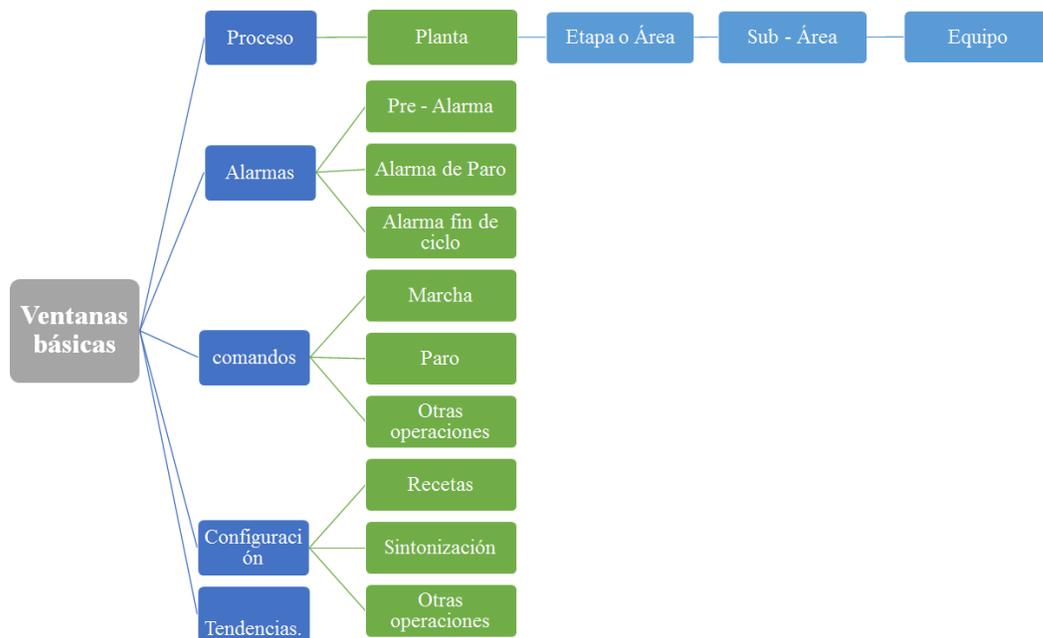


Figura 2. 74. Jerarquía de ventanas básicas en una aplicación HMI industrial  
[Editado de: [4]]

Teniendo presente las anteriores recomendaciones se puede desarrollar la fase de prototipado.

## 2.5 Evaluación.

El modelo MPLu+a propone que la evaluación de la aplicación se lleve a cabo en todo el ciclo de vida del proceso de desarrollo [15], siempre teniendo en cuenta los objetivos de usabilidad, accesibilidad y el usuario final del sistema, para que haya una retroalimentación constante con los evaluadores para la mejora del sistema. Es conveniente que esta evaluación se realice en cada aspecto que sea considerado relevante para el desarrollo del sistema, para que no se cometa el error de evaluar al finalizar todo el proceso o en un solo punto de este. Por otra parte, un aspecto importante dentro del modelo, es que incluye la evaluación de la accesibilidad como un proceso integrado en el ciclo de vida del desarrollo de sistemas software, aspecto que se ha visto separado de dicho ciclo de vida.

En [33] se definen tres objetivos principales en cuanto a la evaluación de sistemas:

- Comprobar la extensión de la funcionalidad del sistema.
- Comprobar el efecto de la interfaz en el usuario.
- Identificar cualquier problema específico con el sistema.

El diseño del sistema debe permitir al usuario realizar sus tareas de una forma fácil. Además es importante comprobar el impacto de dicho diseño sobre el usuario, esto incluye aspectos como la facilidad de su aprendizaje, su manejabilidad, etc. Por último se deben identificar los problemas específicos del diseño [15].

Adicional a estos aspectos de evaluación se deben tener en cuenta varios tópicos relacionados con la usabilidad en dispositivos móviles, en [34] se hace referencia a tres aspectos claves:

- El limitado tamaño de la pantalla de los dispositivos móviles.
- La falta de herramientas de software específicas.
- Las dificultades adicionales derivadas de un contexto móvil.

El tamaño de la pantalla, debe ser analizado de una manera apropiada, ya que en los dispositivos móviles la interfaces deben tener un tamaño adecuado y la información debe presentarse correctamente.

A pesar del constante desarrollo en los dispositivos, estos cuentan con ciertas desventajas tecnológicas en cuanto a software y hardware, si se comparan con los ordenadores de mesa. Por esta razón las aplicaciones se deben desarrollar teniendo presente este aspecto que no es menos importante que el anterior.

Por ultimo cuando se habla de un contexto móvil se debe tratar el tema de que el usuario y el teléfono están en constante movimiento, y por tanto, el entorno es cambiante. Por otra parte, las tareas del usuario están sujetas a las interrupciones (ejemplo de esto es la perdida de cobertura, llamadas entrantes o distracciones) [35].

Ya que la guía de diseño está enfocada hacia los dispositivos móviles Touchscreen, en [36] se tienen en cuenta algunos otros aspectos en la evaluación de la usabilidad, entre los que se encuentran:

- Resolución de la pantalla: Diferentes resoluciones de pantalla pueden conducir a diferentes aspectos de usabilidad.
- Procesamiento limitado, capacidades de memoria y energía: algunas aplicaciones necesitan altos niveles de memoria y procesamiento ya sea para dar soporte a graficas o llevar a cabo tratamiento de datos.
- Métodos de entrada de datos: Botones y etiquetas pequeñas pueden reducir la eficiencia y efectividad en la entrada de datos al sistema. Reduciendo así la velocidad de interacción del usuario con este.

En [15] los métodos de evaluación están clasificados de la siguiente manera:

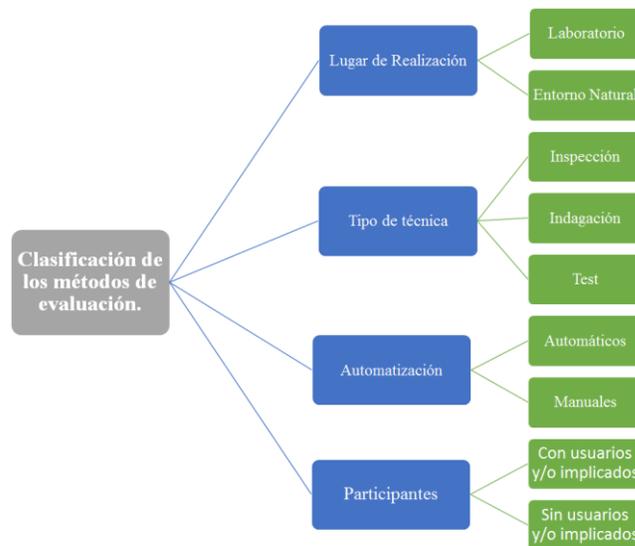


Figura 2. 75. Clasificación de los métodos de evaluación.  
[Fuente: propia].

Son muchas las técnicas existentes para evaluar la usabilidad de aplicaciones informáticas, en la Figura 2. 75 se observa que estas se clasifican como:

- Inspección.
- Indagación.
- Test.

La primera clasificación está enfocada hacia un conjunto de métodos, donde la usabilidad es evaluada por expertos llamados “evaluadores” los cuales explican el grado de usabilidad el sistema basándose en la inspección. En la Tabla 2. 11 se observan los métodos más importantes de esta clasificación:

Método	Característica	Clasificación
Heurística	Proceso en el que se resuelven problemas a partir de una serie de reglas (heurísticas) previamente determinadas.	
Recorridos	Aproximación alternativa a la evaluación heurística, con los que se intentan predecir problemas de usabilidad sin realizar test con usuarios	<ul style="list-style-type: none"> <li>• Recorrido cognitivo.</li> <li>• Recorrido de usabilidad plural.</li> <li>• Recorrido cognitivo con usuarios. (Propuesto por MPLu+a).</li> </ul>
Inspección de estándares	Se precisa de un evaluador experto en el o los estándares a evaluar. Realiza una inspección completa a la interfaz para comprobar que cumple en todo momento los puntos definidos en el estándar establecido.	<ul style="list-style-type: none"> <li>• Estándares de irue</li> <li>• Estándares de Facto</li> </ul>

Tabla 2. 11. Técnicas de evaluación por inspección  
[Fuente: propia]

Las técnicas de evaluación por indagación tienen como objetivo llegar al conocimiento de un sistema discurrendo o por conjeturas y señales. Por medio de estas técnicas se habla con los usuarios y se observan detenidamente usando el sistema en tiempo real y obteniendo respuestas a preguntas formuladas verbalmente o por escrito.

En la Tabla 2. 12 se muestra una clasificación de los métodos por indagación más importantes:

<b>Método</b>	<b>Característica</b>
Observación de Campo	Su principal objetivo es entender cómo los usuarios de los sistemas interactivos realizan sus tareas y conocer todas las acciones que estos llevan a cabo durante la realización de las mismas.
Grupo de discusión dirigido (Focus Group)	Técnica de recogida de datos donde se reúnen de 6 a 9 personas (usuarios e implicados) para discutir aspectos relacionados con el sistema. Un experto en usabilidad y/o accesibilidad, hace el papel de moderador, prepara los temas a discutir y recoge la información necesaria.
Entrevistas	Conversación donde uno o más usuarios reales del sistema que se desarrollara o rediseñara responden a una serie de preguntas relacionadas con el sistema que el entrevistador va formulando.
Cuestionarios	Listas de preguntas que el evaluador distribuye entre usuarios para que estos devuelvan respuestas que luego serán debatidas, con la finalidad de sacar conclusiones.
Grabación del uso (análisis de logs)	Conocida también como logging, se basa en grabar o recoger todas las actividades realizadas por el usuario con el sistema para su posterior análisis.

*Tabla 2. 12. Métodos de evaluación por indagación.  
[Fuente: propia]*

Por último los métodos de evaluación de la usabilidad de un sistema por medio de Test son caracterizados por que usuarios representativos trabajan en tareas utilizando el sistema –o el prototipo- y los evaluadores utilizan los resultados para ver como la interfaz de usuario soporta a los usuarios con sus tareas.

En la Tabla 2. 13 se ilustran los principales métodos de evaluación por Test.

<b>Método</b>	<b>Característica</b>
Medida de las prestaciones	Basado en la toma de medidas acerca del rendimiento u otro tipo de aspecto subjetivo que afecte la usabilidad del sistema, para esto se ha de disponer del sistema ya implementado o un prototipo que permita evaluar estos aspectos.
Pensando en voz alta (Thinking Alaud)	Método descrito por Nielsen, en el cual se pide a los usuarios de una manera individual que expresen en voz alta y libremente sus pensamientos, sentimientos y opiniones sobre cualquier aspecto (diseño, funcionalidad, accesibilidad, etc.) mientras interactúan con el sistema o un prototipo del mismo.
Interacción Constructiva	Conocido como aprendizaje por codescubrimiento, es una derivación del Thinking aloud e implica tener, en vez de uno, a dos usuarios realizando conjuntamente cada test del sistema.
Test retrospectivo	Para llevar a cabo este método es necesario tener una grabación de la sesión de test, para que luego el usuario pueda revisarlo y genere más comentarios que cuando está realizando el test en tiempo real.
Método del conductor	En este método se conduce al usuario en la dirección correcta mientras se usa el sistema, durante este test el usuario puede preguntar cualquier aspecto relacionado con el sistema y el evaluador le responderá. Es centrado en el usuario inexperto.
Ordenación de tarjetas (Card Sorting)	Utilizada para conocer como los usuarios visualizan la organización de la información. El diseñador utiliza las aportaciones de los usuarios para decidir cómo deberá estructurarse la información en la interfaz.

*Tabla 2. 13 Métodos de evaluación por test  
[Fuente: propia]*

En [15] y [33] se explican de una manera detallada todos los métodos descritos anteriormente.

Todos los métodos existentes para evaluar la usabilidad, son herramientas que el equipo desarrollador puede escoger libremente según las necesidades y los aspectos que se vayan a evaluar. En [15] se muestra una tabla (Tabla 2. 14) comparativa de todos los métodos, con sus principales características, los momentos más adecuados para aplicar en el ciclo de vida, y si los resultados obtenidos son cuantitativos (lista de errores o mejoras detectadas o de acciones concretas a realizar) o no lo son (conjunto de datos que se deben analizar posteriormente).

Como se planteó anteriormente, la usabilidad en dispositivos móviles difiere un poco a la web, por aspectos como la pantalla del dispositivo en el que se está trabajando, la movilidad y las carencias tecnológicas de hardware y software [35], razones por las cuales se han realizado numerosos estudios acerca de la usabilidad en dispositivos móviles que han arrojado resultados favorables para dar soporte a muchos desarrolladores. Es así que por ejemplo en [36] y [37] se han descrito una serie de modificaciones a las heurísticas de Nielsen [38] para evaluar la usabilidad en el contexto de las aplicaciones móviles.

Ya que son muchos los métodos existentes para llevar a buen término la evaluación de la usabilidad, se deben tener en cuenta todos los aspectos adicionales que conlleva la usabilidad en aplicaciones móviles, los cuales se describieron anteriormente. En la presente guía se ha optado por escoger el método heurístico, ya que es un método que está bien documentado, ha sido probado satisfactoriamente y cuenta con una serie de publicaciones

relacionadas con su aplicación en el contexto móvil, concretamente Smartphones Touchscreen . Por lo que se hará una descripción más detallada del método.

	Tipo	Fase	Lugar	Usuarios	Autom.	Remot o	Usabilidad			Datos cuant .
							Efect.	Efic.	Satisf	
Heurística	Inspección	D//L	L	No	No	Si	Si	Si	No	Si
Rec. Usab. Plural		AR/D	L	Si	No	No	Si	No	Si	No
Recorrido Cognitivo		AR/D/I	L	No	No	No	Si	No	No	Si
Rec. Cog. Con usuarios		AR/D/I	L/E	Si	No	No	Si	No	Si	Si
Estándares		D//L	L	No	No	Si	No	Si	Si	Si
Observ. de campo	Indagación	AR	E	Si	No	No	Si	No	Si	No
Focus Group		AR/D//L	L/E	Si	No	No	Si	No	Si	Si
Entrevistas		D//L	L/E	Si	No	No	Si	No	Si	No
Cuestionarios		L	L/E	Si	Ambos	Si	No	No	Si	Si
Logging		L	L/E	No	Si	Si	Si	Si	No	Si
Medida Prestaciones	Test	I/L	L/E	Si	Ambos	No	Si	Si	No	Si
Thinking Aloud		D//L	L/E	Si	No	No	Si	No	Si	No
Interacción constructiva		D//L	L/E	Si	No	No	Si	No	Si	Si
Test Retrospectivo		D//L	L/E	Si	No	No	Si	Si	Si	Si
Método Conductor		D//L	L/E	Si	No	No	Si	No	Si	No
Ordenación tarjetas		AR/D	L/E	Si	Ambos	Si	Si	Si	No	Si

Tabla 2. 14. Métodos de evaluación con sus principales características.  
[Fuente: [15]]

**Fase:** AR (Análisis de requisitos), D (Diseño), I (Implementación), L (Lanzamiento).

- La fase en negrilla indica que es la fase más adecuada para la aplicación del método.
- **Lugar:** L (Laboratorio), E (Entorno).

### 2.5.1 Evaluación Heurística.

Puede ser descrito como un método de valuación de la usabilidad por inspección, que se caracteriza por que es llevado a cabo por evaluadores expertos a partir de unos principios o reglas (heurísticas) previamente establecidos. Tiene como objetivo medir la calidad de la interfaz de cualquier sistema interactivo en relación a su facilidad para ser aprendido y usado por un determinado grupo de usuarios en un determinado contexto de uso [39].

La metodología de la evaluación heurística fue desarrollada por Nielsen y Molich, y esta conlleva a que un grupo de expertos inspeccionen la interfaz y evalúen el grado de cumplimiento de cada uno de las 10 reglas heurísticas (principios o reglas). Cada evaluador

inspecciona la interfaz de una manera individual. Una vez concluidas estas evaluaciones, se lleva a cabo una reunión entre los evaluadores y se elabora un informe con los resultados de esta, este informe puede ser realizado por los mismos evaluadores en consenso o por el responsable de la evaluación, que es llamado también “observador” [40].

Las 10 reglas heurísticas se presentan a continuación:

1. **Visibilidad del estado del sistema:** El sitio web o aplicación debe mantener siempre informado al usuario de lo que está ocurriendo y brindarle una respuesta en el menor tiempo posible.
2. **Relación entre el sistema y el mundo real.** El sitio web o aplicación debe utilizar el lenguaje del usuario, con expresiones y palabras que le resulten familiares. La información debe aparecer en un orden lógico y natural.
3. **Libertad y control por parte del usuario.** En caso de elegir alguna opción del sitio web o aplicación por error, el usuario debe disponer de una “salida de emergencia” para abandonar el estado no deseado en que se halla. Debe poder deshacer o repetir una acción realizada.
4. **Consistencia y estándares.** Los usuarios no tienen por qué saber que diferentes palabras, situaciones o acciones significan lo mismo. Es conveniente seguir convenciones.
5. **Prevención de errores.** Es importante ayudarle al usuario a que no caiga en un error. La funcionalidad de autocomplete de los buscadores ayuda a que una persona no tenga que escribir toda la palabra y no se equivoque.
6. **Reconocer antes que recordar.** Hacer visibles acciones y opciones para que el usuario no tenga que recordar información entre distintas secciones o partes del sitio web o aplicación. Es importante mantener a nivel de diseño visual un estándar para que los elementos de interface sean consistentes en diferentes pantallas.
7. **Flexibilidad y eficiencia en el uso.** Los aceleradores o atajos de teclado pueden hacer más rápida la interacción para usuarios expertos, de tal forma que el sitio web o aplicación sea útil tanto para usuarios básicos como avanzados.
8. **Diseño estético y minimalista.** Las páginas no deben contener información innecesaria. Cada información extra compite con la información relevante y disminuye su visibilidad.
9. **Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores.** Los mensajes de error deben estar redactados con un lenguaje simple, lo peor son errores como “Error 34-x1” que no le dicen nada al usuario de cómo puede recuperarse, deben ofrecerse alternativas para que el usuario pueda continuar realizando la tarea o recuperando lo último que hizo (autosave).
10. **Ayuda y documentación.** Aunque es mejor que el sitio web o aplicación pueda ser usado sin ayuda, puede ser necesario proveer cierto tipo de ayuda. En este caso, la ayuda debe ser fácil de localizar, especificar los pasos necesarios y no ser muy extensa. [41].

Como se comentó anteriormente se han realizado varios estudios en donde se han aplicado y analizado estas 10 reglas heurísticas en el contexto móvil. Es así que, por ejemplo en [37] y [42] se ha optado por eliminar algunas heurísticas, para contextualizar la evaluación hacia dispositivos móviles. En [43] en cambio, se proponen nuevas heurísticas que ayudan a realizar la evaluación, y se crea una lista de evaluación (checklist). En tanto que en [36] se ha propuesto la adición de algunas heurísticas para la evaluación de la usabilidad en

dispositivos móviles táctiles. Este último estudio es el que se tendrá en cuenta en la presente guía, ya que la HMI industrial está desarrollada en un principio para tablets o Smartphone táctiles.

En [36], luego de una serie de estudios se propusieron 12 heurísticas de usabilidad. Las primeras 11 se hicieron en base a las reglas heurísticas de Nielsen y se complementaron con aspectos relevantes de los dispositivos móviles con pantalla táctil. La heurística 12 se obtuvo teniendo en cuenta varios conceptos móviles y aspectos de usabilidad.

Las 11 heurísticas propuestas en [36] son:

1. **Visibilidad del estado del sistema:** El dispositivo debe mantener siempre informado al usuario y los cambios de estado del proceso, a través de una retroalimentación específica en un tiempo razonable.
2. **Correspondencia entre el sistema y el mundo real:** La aplicación deberá hablar el mismo lenguaje que el usuario, seguir convenciones del mundo real mostrando la información de una manera lógica y un orden natural.
3. **Control y libertad del usuario:** La aplicación deberá permitir al usuario deshacer y rehacer acciones, deberá tener salidas de emergencia. Las opciones se deben marcar claramente, y el usuario no se deberá forzar al usuario a pasar a través de diálogos extensos.
4. **Consistencia y estándares:** La aplicación deberá seguir las convenciones establecidas, el usuario deberá hacer las cosas de una manera familiar, estándar y consistente.
5. **Prevención de errores:** La aplicación deberá tener un diseño de interfaz que prevenga errores, que oculte funcionalidades no disponibles. Se debe permitir acceso a información adicional y se debe advertir a los usuarios cuando los errores son probables de que ocurran.
6. **Minimizar la carga de memoria del usuario:** Objetos, acciones y opciones visibles, que el usuario no tenga que recordar la información de una parte del dialogo a otra.
7. **Personalización y atajos:** El sistema debe proporcionar opciones de configuración básica y avanzadas. Debe permitir personalizar y definir, o proporcionar atajos.
8. **Eficiencia de uso y rendimiento:** La aplicación debe realizar sus tareas en el mínimo tiempo posible. También debe minimizar los pasos para llevar a cabo una tarea determinada. Las animaciones y las transiciones se deben mostrar sin ningún problema.
9. **Estética y diseño minimalista:** Se debe evitar mostrar información irrelevante o rara vez necesaria.
10. **Ayuda al usuario para reconocer, diagnosticar y recuperarse de errores:** El dispositivo debe mostrar mensajes de error en un lenguaje natural y familiar para el usuario, indicando el error de una manera precisa.
11. **Ayuda y documentación:** El sistema debe proporcionar ayuda y documentación fácil de encontrar centrada en el usuario, y sin ser demasiado extensa.
12. **Interacción física y ergonómica:** Proporcionar botones físicos o elementos de interfaz similares y adecuados para sus principales funciones, deben estar ubicados en posiciones reconocibles.

Estas heurísticas se complementaron con una propuesta de Checklist para realizar la evaluación de una manera eficiente. Aunque son muchos los Checklist que se encuentran

para realizar esta evaluación, la guía propone que se utilice el propuesto en [36], ya que contempla todos los aspectos que se deben evaluar en un dispositivo móvil, tales como el contexto en el que el usuario utiliza el dispositivo, el diseño de la interacción, la eficacia y eficiencia de la aplicación, etc.

## 2.6 Implementación.

El modelo MPlu+a no enfatiza mucho en esta fase debido a que ésta corresponde más a un proceso de ingeniería software. Gracias a que se han tratado todos los aspectos de usabilidad y accesibilidad en las fases anteriores, el trabajo para estas últimas dos, se minimiza considerablemente, esto es una gran ventaja, teniendo en cuenta que esta es la fase donde hay una mayor demanda de recursos humanos. Teniendo claras todas las especificaciones que se establecieron en el análisis de requisitos y en el diseño, se lleva a cabo esta fase que es donde se realiza toda la codificación por parte de los desarrolladores encargados para tal fin.

El modelo si hace hincapié en que desde el principio de esta fase se tiene que realizar varios prototipos teniendo siempre en cuenta al usuario final, para poder garantizar la usabilidad y accesibilidad del sistema final. El equipo debe tener siempre presente que cualquier cambio que se realice debe ser consultado con el/los usuarios finales para que se siga la línea del modelo de proceso. En [15] se especifica que las aportaciones de evaluar en esta fase son altamente valiosas para no malgastar tiempo en desarrollar software que después deberá, sin ninguna duda, ser cambiado.

## 2.7 Lanzamiento

En esta fase es donde se muestran los resultados del producto final y si efectivamente se cumplieron los objetivos trazados antes de empezar el desarrollo, por esta razón es la fase con un alto grado de criticidad [14].

El éxito del producto dependerá de dos factores importantes:

- La satisfacción y comodidad del usuario con el sistema. Que el sistema no le genere errores, que no le sea complicado utilizarlo, que recuerde donde están las diferentes opciones y sus funcionalidades, etc.
- Que los responsables del proyecto obtengan los resultados esperados, esto está relacionado con la funcionalidad ofrecida por el sistema [15].

Debido a la aplicación satisfactoria del modelo, estas dos premisas se cumplen a cabalidad, ya que el usuario ha estado involucrado en todo el desarrollo y por tanto todo el sistema se habrá hecho para él.

El modelo enfatiza en un concepto muy importante: el **feedback** (realimentación con el usuario), el cual está relacionado con el tiempo de prueba del sistema por parte de los usuarios y las quejas, comentarios, dudas u opiniones que estos generan al momento de probarlo. Una vez ha transcurrido este tiempo de prueba (test), todos los datos y aspectos recogidos son tenidos en cuenta para realizar las mejoras respectivas al sistema con el único fin de generar al usuario una satisfacción total que es el objetivo primordial por parte del grupo desarrollador.

## 2.8 Resumen de la guía propuesta.

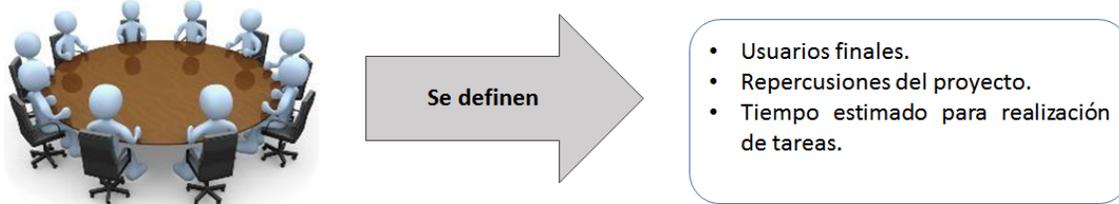
En el siguiente apartado se pretende mostrar al lector a manera de resumen los aspectos más relevantes que se deben tener en cuenta cuando se va a llevar a cabo el desarrollo de una HMI industrial bajo el sistema operativo Android, teniendo en cuenta la guía de diseño propuesta.

### 2.8.1 Análisis de requisitos.

En primera instancia el modelo MPIu+a propone llevar a cabo un análisis de requisitos. Para esto se deben tener en cuenta a todos los implicados (directos e indirectos) involucrados en el desarrollo de la HMI.

#### 2.8.1.1 Reunión de implicados.

Para poder definir los alcances del proyecto se recomienda que primero se realice una “reunión de implicados” para definir aspectos importantes como: cuáles serán los usuarios finales de la aplicación, cuál será la repercusión del proyecto a desarrollar, el tiempo de realización y objetivos primarios.



*Figura 2. 76. Reunión de implicados.  
[Fuente: propia]*

Una vez se han definido los usuarios finales de la aplicación se debe realizar una pequeña reunión conjunta o reuniones con cada uno de ellos expresándoles los objetivos de la aplicación y se debe pedir su consentimiento para realizar el análisis etnográfico y demás actividades propias del desarrollo del proyecto.

#### 2.8.1.2 Análisis Etnográfico.

El análisis etnográfico se debe realizar de una manera adecuada ya que es la principal actividad de la fase de análisis de requisitos, y repercutirá en las demás actividades del desarrollo. La actividad idónea en este punto es la “observación”. Se deben anotar de una manera detallada todas las tareas que realiza el usuario final para la supervisión del proceso que se está llevando a cabo, así como los requisitos que estos vayan generando ya sea teniendo una charla informal con cada uno de ellos o con una pequeña entrevista. Al término del análisis etnográfico se tendrán de una manera más clara los requisitos de los usuarios así como su papel dentro de la organización donde se lleva a cabo el proceso industrial.

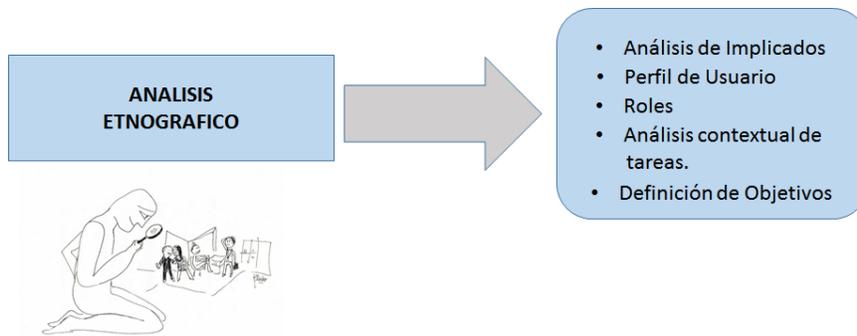


Figura 2. 77. Actividades principales en el análisis de requisitos. El análisis etnográfico repercute sobre las otras actividades.

[Fuente: Propia]

### 2.8.1.3 Clasificación de usuarios finales.

Una vez consolidada la información acerca de cada uno de los usuarios se puede consignar en una tabla como la que se muestra a continuación:

Nro. Usuario	Nombre	Edad	Perfil de usuario	Rol en la organización
1	Pedro Pérez	30	operario planta	Monitorización de procesos
.	.	.	.	.
.	.	.	.	.
n	n	n	n	n

Tabla 2. 15. Información de los usuarios finales del sistema a desarrollar.

[Fuente: Propia]

Es importante tener en cuenta la experiencia que tienen los usuarios respecto al uso de teléfonos móviles, esto con el fin de establecer condiciones respecto al desarrollo del proyecto siempre pensando en que los usuarios llevarán a cabo un buen uso del dispositivo en donde se albergará la HMI.

### 2.8.1.4 Análisis contextual de las tareas.

Luego de definir los usuarios con sus respectivos perfiles y roles dentro de la organización donde se encuentra el proceso productivo es necesario realizar un “análisis contextual de las tareas”, enfocado hacia determinar cuáles son las herramientas que utilizan actualmente los usuarios finales de la aplicación respecto a la supervisión del proceso. Esto con el fin de determinar de una manera clara cuales son y como realizan cada una de las tareas enfocadas hacia la monitorización y supervisión del proceso industrial.

### 2.8.1.5 Primer prototipo de baja fidelidad y definición de objetivos del sistema a desarrollar.

Una vez obtenida toda la información relacionada con los requisitos y necesidades de los usuarios, es aconsejable que el grupo desarrollador realice un primer prototipo de baja fidelidad de la aplicación a desarrollar, con el fin de realizar un análisis de la información suministrada por los usuarios, teniendo en cuenta el análisis contextual de las tareas y el

análisis etnográfico. La guía de diseño propone que la técnica para realizar este primer prototipo sea la de boceto, ya que es una técnica que no requiere de mucho tiempo y es muy económica.

El modelo MPlu+a recomienda que para llevar a cabo evaluaciones en las primeras fases del desarrollo, se utilice el método denominado “focus Group” (grupo de discusión dirigido). En este se les presenta a los usuarios finales el primer prototipo de baja fidelidad con el fin de que estos expresen cuales son las funcionalidades que debe cumplir el sistema a desarrollar. Al final de esta reunión se obtendrán los objetivos finales de la aplicación: objetivos funcionales, objetivos de usabilidad y accesibilidad. En la figura 2.79 se observa de una manera general lo expuesto.

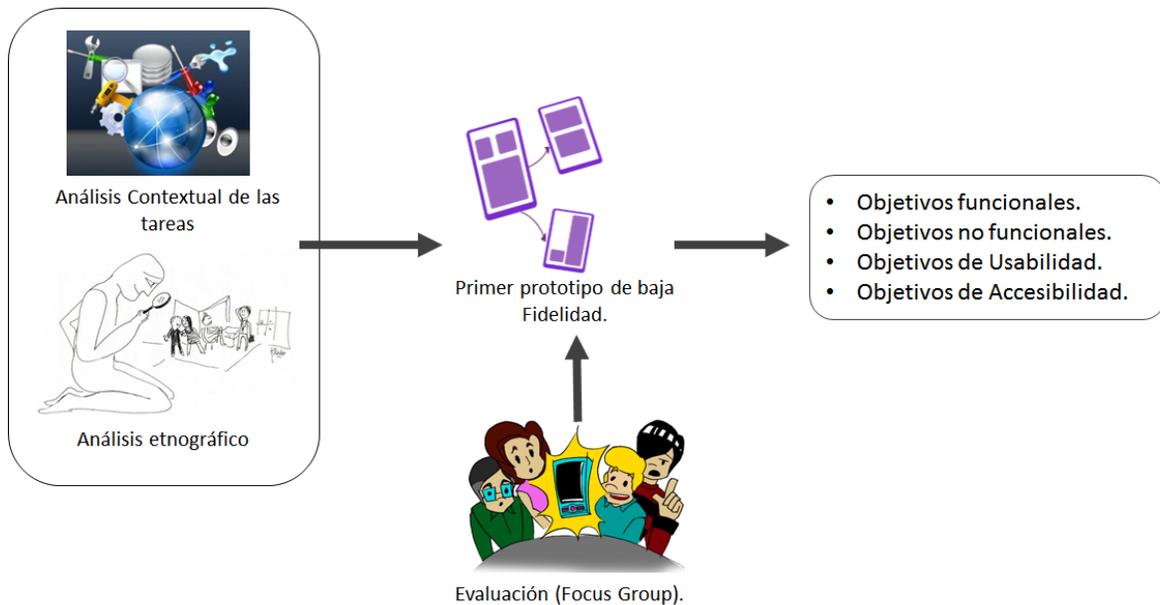


Figura 2. 78 Esquema general para la obtención de los objetivos de la aplicación a desarrollar. [Fuente: propia]

## 2.8.2 Diseño de la interacción.

El diseño de la información está dividido en dos actividades principales:

- Diseño de la actividad.
- Diseño de información.

### 2.8.2.1 Diseño de la actividad.

La principal actividad a desarrollar en esta fase es el análisis de las tareas, el cual se puede ver como una descripción sistemática de las interacciones usuario – sistema. El objetivo principal en este punto es realizar una descripción formal de todas las actividades que realiza el usuario para la consecución de un objetivo trazado.

### 2.8.2.2 Análisis de las tareas.

Para realizar este análisis de tareas existen muchos métodos, aunque el más usado y consolidado es el **HTA** (análisis jerárquico de las tareas), esto debido a que es un método muy fácil de usar; razón por la cual se recomienda en la guía de diseño propuesta.

### 2.8.2.3 Diseño de la información.

Luego de establecer formalmente las tareas que realizan los usuarios para el proceso de supervisión gracias al HTA. El siguiente paso es realizar toda la parte física de la interacción. Se deben tener en cuenta las recomendaciones enfocadas hacia el uso de los colores, los sinópticos a utilizar, la distribución de los elementos dentro de la interfaz, el tamaño de las imágenes, los diálogos a utilizar en el sistema, etc.

Para desarrollar la interfaz gráfica de la aplicación bajo el sistema operativo Android se deben tener en cuenta estas recomendaciones.

- Recuerde desarrollar la aplicación bajo las dos orientaciones del dispositivo. Portrait (horizontal) y landscape (vertical).
- Los colores en Android se representan de una manera Hexadecimal, lo cual se debe tener en cuenta al momento de diseñar los layouts de la aplicación.
- El tratamiento de las alarmas se debe realizar teniendo en cuenta un `TableLayout` para poder asemejar una tabla relativa a los eventos del proceso.
- Recuerde que para la gestión de los datos Android utiliza el sistema SQLite, que utiliza el lenguaje de consulta y acceso a base de datos SQL. El paquete que se utiliza para manipular las bases de datos es **android.database.sqlite**.
- Recuerde utilizar un degrade de tonalidades claras de colores como el gris, azules, beige y arena, para dar el fondo a la pantalla.
- Tener presente la regla de Gutenberg para la distribución de los elementos.
- El sinóptico debe representar la planta de una manera minimalista, el realismo es aceptable en pocas proporciones.
- En Android la clase para realizar animaciones y tratamiento de imágenes en `AnimationDrawable`.
- Recuerde que para desarrollar las gráficas en Android se recomienda utilizar la librería `Androidplot`.
- A nivel industrial para el manejo del texto se utiliza la fuente san – serif (Narrow, Tahoma, Verdana, Arial Narrow), recuerde que Android utiliza el tipo de fuente `Roboto` que también es un tipo de fuente san – serif.

### 2.8.2.4 Segundo prototipo: Maqueta digital.

Teniendo estas premisas presentes es recomendable que el equipo desarrollador realice un primer prototipo digital, en donde se plasmen las ventanas de la aplicación; la guía de diseño recomienda que para este primer prototipo se utilice el software libre “Pencil Project”, que es muy fácil y rápido de utilizar. Aunque se puede escoger la opción que el equipo considere más adecuada. Una vez desarrollada esta primera maqueta de la aplicación es conveniente que sea evaluada; en este sentido el MPLu+a es muy flexible en cuanto que se puede utilizar cualquier método de evaluación. La guía de diseño recomienda que esta primera evaluación la realice un experto basándose en el método de “inspección de

estándares” ya que se cuenta con los parámetros de diseño de Android y las recomendaciones propuestas en la guía de diseño. Otra buena opción es realizar un focus group con los usuarios finales para recoger las impresiones de estos, frente a la maqueta presentada.

### 2.8.2.5 Tercer prototipo: Maqueta digital.

Ya que por lo general cada vez que se realizan evaluaciones de los prototipos siempre se generan nuevos requisitos o necesidades, es conveniente que luego de esta primera maqueta digital se realicen las correcciones necesarias y se realice otro prototipo más elaborado. Este se puede desarrollar bajo el software Inkscape, el cual es libre. La evaluación para este nuevo prototipo la pueden realizar los usuarios bajo métodos como “Thinking Alaud”, mediante un Focus Group o una entrevista, ya que son métodos en donde se recogen las impresiones de los usuarios de una manera directa. Una vez más el equipo desarrollador puede escoger el método que crea más adecuado. En la figura 2. 80 se ilustra de una manera general el proceso descrito anteriormente.

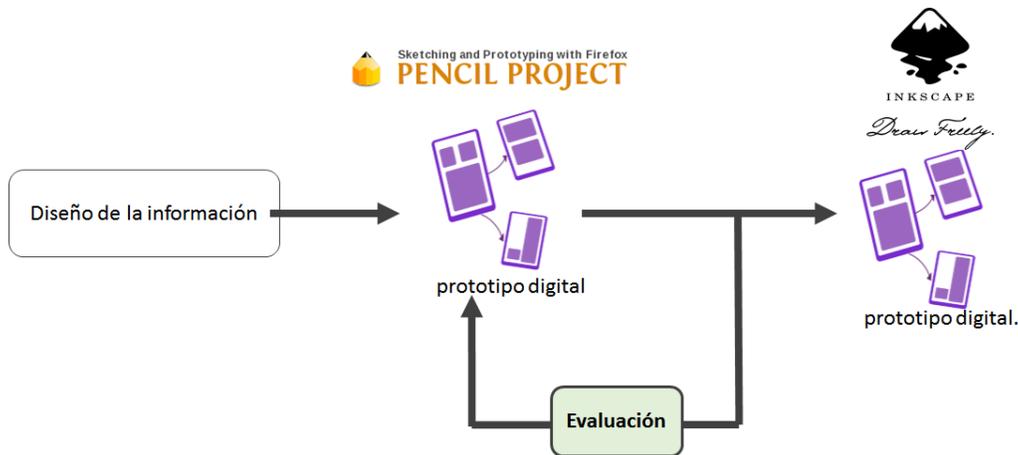


Figura 2. 79. Proceso para el desarrollo de la maqueta digital  
[Fuente: Propia]

Es importante aclarar que aunque en la guía se recomienda llevar a cabo el desarrollo de dos prototipos digitales para evaluar el diseño de la interfaz, el número de prototipos a desarrollar será proporcional a la satisfacción de los usuarios finales respecto a los requisitos y necesidades generados por estos.

### 2.8.2.6 Story board Navegacional.

Una vez que el diseño de la interacción está definido y antes de pasar a la implementación del proyecto es aconsejable realizar un prototipo para poder mostrar la navegabilidad del sistema, esto con el fin de que los usuarios finales puedan llevar a cabo la consecución de las tareas definidas. Una técnica de prototipado ideal para esta etapa es el Story board Navegacional, ya que se asemeja al modelo de navegabilidad que se utiliza normalmente en el desarrollo de aplicaciones móviles y tiene como objetivo evaluar como los usuarios realizan la interacción con el sistema para la consecución de las tareas trazadas. Este prototipo es aconsejable que se construya en papel ya que se ahorra tiempo y recursos. En la figura 2. 81 se ilustra un prototipo de Story board en donde se asemejan las ventanas

finales de una aplicación para la supervisión de un proceso y la navegación que se realiza por medio de flechas.

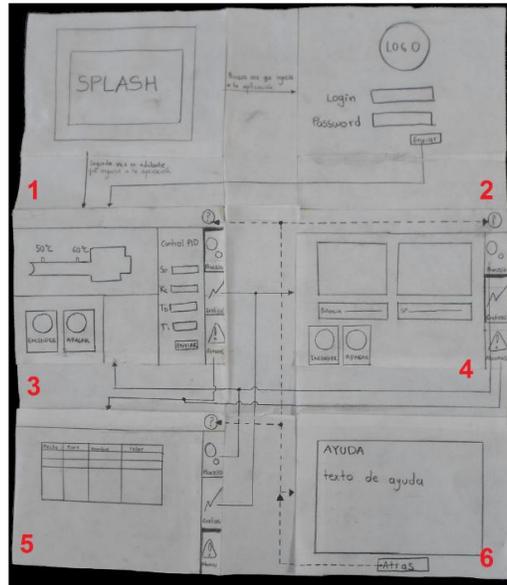


Figura 2. 80. Prototipo Story board para la supervisión de un proceso.  
[Fuente: Propia]

Para evaluar este prototipo el equipo desarrollador puede utilizar cualquier técnica de evaluación, siempre y cuando se tenga al usuario presente para que pueda realizar las tareas definidas con el prototipo. La guía propone que se realice esta evaluación mediante un recorrido cognitivo con usuarios, esto debido a que con este método se les pueden explicar las tareas a los usuarios y pueden llevar a cabo la interacción

### 2.8.3 Implementación.

Una vez terminado y evaluado el story board navegacional y las maquetas digitales se puede pasar a la fase de implementación, esta es la fase que requiere de más recursos humanos, pero debido a que el usuario ha estado presente en todo el desarrollo del proyecto y se han realizado una serie de evaluaciones para los prototipos realizados, factores como el tiempo y recursos se ven minimizados gracias a que se ha llevado a cabo un buen trabajo previamente gracias al MPIu+a.

El entorno de desarrollo ideal para implementar aplicaciones nativas bajo el sistema operativo Android es Eclipse ya que es el que utilizan los desarrolladores de Google. Además se basa en el lenguaje JAVA para la escritura del código fuente.

Para implementar de una manera adecuada el proyecto planteado, se deben llevar a cabo los siguientes pasos:

1. En primera instancia se deben definir todos los recursos propios de la aplicación a desarrollar.
2. Se desarrollan todos los layouts de la aplicación de supervisión del proceso, recuerde que el lenguaje utilizado para este fin es XML.
3. Se desarrolla la lógica para cumplir con los objetivos funcionales del sistema.

En la figura 2. 82 se ilustra un esquema general para realizar la implementación del sistema interactivo.

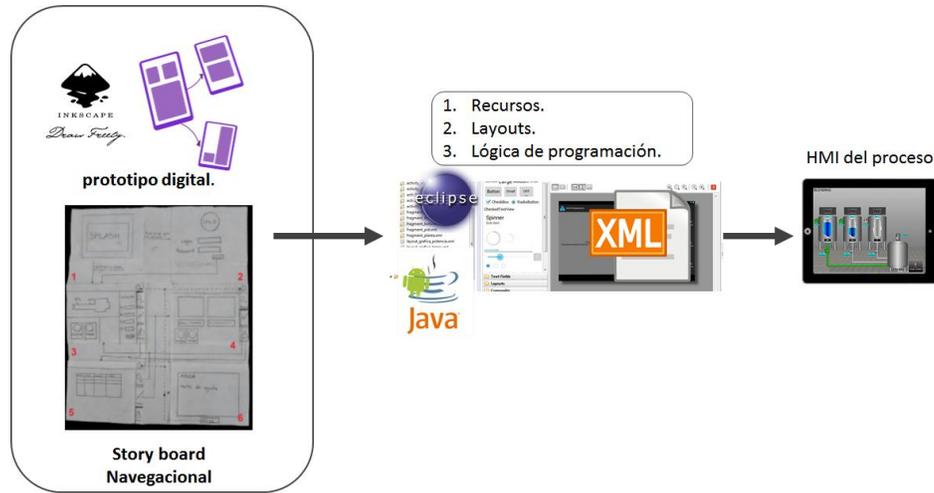


Figura 2. 81. Implementación de la HMI a desarrollar.  
[Fuente: Propia]

### 2.8.3.1 Evaluación del prototipo Software desarrollado.

Aunque son muchas las técnicas que se pueden utilizar para realizar la evaluación en las últimas fases del desarrollo de un sistema interactivo, se debe tener muy en cuenta que la evaluación que se tiene que llevar a cabo es en un dispositivo móvil, razón por la cual se deben considerar aspectos importantes como la pantalla del dispositivo en el que se está trabajando, las carencias hardware y software y el contexto móvil del usuario. En este sentido la guía propone que se utilice el método heurístico para realizar la evaluación en estas últimas fases, ya que se cuenta con algunos trabajos investigativos relacionados con su aplicación en el contexto móvil, concretamente en dispositivos Smartphones Touchscreen. Una vez realizada esta evaluación se puede tener en cuenta a los usuarios finales de la aplicación para que evalúen el sistema interactivo completo. La técnica a utilizar puede ser la que el grupo desarrollador crea más conveniente. En la guía de diseño se propone que se utilice la técnica “Cuestionarios tipo indagación” ya que se cuenta con una herramienta versátil denominada MUMMS, desarrollada por el grupo de investigación irlandés HFRG (Human factor Research group) de la universidad College Cork de Irlanda. En esta se pueden evaluar aspectos de usabilidad por parte de los usuarios finales del sistema. Una vez realizado este procedimiento se pueden obtener datos concretos acerca de cuál es el índice de satisfacción por parte del usuario, así como el cumplimiento o no de los objetivos trazados por el grupo desarrollador.

### 2.8.4 Lanzamiento.

Una vez realizadas todas las etapas previstas, se puede llevar a cabo el lanzamiento del producto con la plena seguridad que se han cumplido los objetivos trazados, ya que el usuario ha estado en todas las fases del desarrollo. Es conveniente que se evalúe periódicamente el sistema desarrollado por si se necesitan realizar actualizaciones.

### **3. Aplicación de la guía propuesta a un caso de estudio.**

En el presente capítulo se mostraran todas las actividades llevadas a cabo por el equipo desarrollador para poder aplicar la guía de diseño propuesta, teniendo siempre como base el MPIu+a, con esto se quiere evidenciar que la guía propuesta es un método confiable para el desarrollo de las HMIs en Android, y se quiere destacar la importancia de la inclusión del usuario en el desarrollo de un sistema interactivo.

El Laboratorio de control de procesos (LCP) del departamento de instrumentación y control de la universidad del cauca cuenta con varias plantas de control de procesos. Para la aplicación de la guía se optó por utilizar la planta de temperatura, ya que es una planta que funciona en perfectas condiciones y reúne una serie de condiciones ideales (control PID, supervisión del proceso, Graficas, etc.).

#### **3.1 Análisis de Requisitos**

El primer paso que se llevó a cabo fue el de una “reunión de implicados” para definir el alcance del proyecto a realizar en el LCP: el objetivo del desarrollo del sistema, los usuarios objetivos de la aplicación y el tiempo estimado para la realización de las tareas correspondientes al desarrollo.

##### **3.1.1 Clasificación de los usuarios.**

En una primera instancia se pensó desarrollar el sistema teniendo como usuarios finales a solo ingenieros del programa de ingeniería en automática industrial, luego de realizar la reunión de implicados, se decidió que la aplicación móvil debía ser desarrollada teniendo en cuenta a los estudiantes del programa así como a la laboratorista del LCP, teniendo esto presente, se concluyó que se tendría en cuenta a estudiantes que hubieran interactuado con la planta de temperatura por lo menos una vez, sin importar su edad, ni condiciones físicas.

La aplicación móvil desarrollada se realizó teniendo como potenciales usuarios a 3 ingenieros pertenecientes al departamento de instrumentación y control de la facultad de ingeniería Electrónica y telecomunicaciones, 4 estudiantes del programa de ingeniería en automática industrial y la encargada del LCP del programa.

Una vez identificados los usuarios finales del sistema se llevó a cabo una reunión conjunta, en donde se plasmó el proyecto a realizar y se pidió la colaboración de cada uno de ellos, refiriéndose especialmente a que la aplicación se iba a desarrollar teniéndolos a ellos siempre presentes. Una vez finalizada la reunión, se concluyó que el equipo desarrollador acompañaría a cada uno de los usuarios en la realización de una sesión de supervisión del proceso para realizar la siguiente fase: el análisis etnográfico.

##### **3.1.2 Análisis Etnográfico**

Se aplicó teniendo en cuenta las recomendaciones del modelo MPIu+a. El objetivo de este, es describir el contexto en el que se sitúa el usuario respecto a la consecución de las tareas que lleva a cabo para la supervisión del proceso de temperatura de la planta del LCP, para

ello se tuvo un acercamiento con el usuario y se realizó una entrevista abierta con cada uno de ellos para que expusieran libremente sus requisitos respecto al sistema a desarrollar.

Debido al alto grado de experiencia que tienen los ingenieros acerca de la planta del LCP las sesiones de supervisión solo se llevaron a cabo con uno de ellos. Por otra parte se realizaron sesiones con la laboratorista y los 4 estudiantes del programa. En cada una de ellas se tuvo en cuenta el método de “observación de campo” (Tabla 2. 12) para evaluar como llevaban a cabo las tareas de supervisión del proceso y a su vez, se iba teniendo una charla informal con cada uno de ellos para que plasmaran sus requisitos y complementar así, la información obtenida.

### 3.1.2.1 Sesiones de supervisión

Las sesiones de supervisión tuvieron un tiempo promedio de 30 minutos, en los cuales se tuvo una pequeña charla introductoria, donde se le pedía a cada usuario que realizara el proceso de supervisión de una manera natural como siempre lo llevaba a cabo. Mientras iba transcurriendo la sesión, el encargado de esta, tomaba notas de todas las tareas que el usuario realizaba, al tiempo que iba escuchando los comentarios que éste hacía acerca del sistema, para poder conocer a grandes rasgos cuál era el modelo mental que tenía de la planta del proceso de temperatura

Teniendo en cuenta la “observación” como principal herramienta se realizó un modelo a grandes rasgos del proceso de supervisión (Ver Figura 3. 1). Este se realiza las veces que el usuario lo crea necesario.

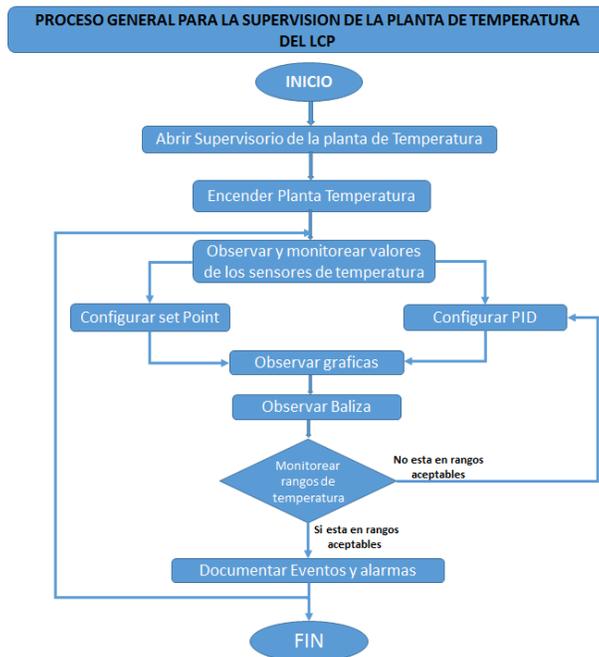


Figura 3. 1. Proceso general para la supervisión del proceso de temperatura. [Fuente: propia]

Esta serie de pasos lo realizaban de una manera similar todos los usuarios. Teniendo en cuenta sus perfiles, se pudo observar que los requisitos de uno u otro eran diferentes, por ejemplo, cuando se realizaba la sesión de supervisión con los estudiantes, su principal enfoque era el monitoreo de los sensores con los que cuenta la planta, al igual que las

gráficas de las variables del proceso, por otra parte, el objetivo principal de la laboratorista era el estado de la planta, que tuviera un funcionamiento normal, llevaba a cabo una observación general del buen funcionamiento de la planta. Por último los ingenieros tenían una visión más completa del proceso, además del monitoreo de los sensores y el estado de las variables por medio de las gráficas, estos se enfocaban en los eventos que se llevaban a cabo, el estado de la baliza con la que cuenta el proceso y las alarmas que se podían generar en éste. Cabe aclarar que en el momento en el que se realizaron las sesiones de supervisión, no se contaba con un sistema de alarmas, por lo cual se convirtió en un requisito por parte de todos los usuarios.

Una vez concluida cada una de las sesiones el procedimiento a seguir fue realizar una pequeña reunión con cada uno de los usuarios, cuyo fin fue evaluar el proceso general descrito en la Figura 3. 1, para que se generara una retroalimentación con cada uno de ellos. Se les preguntó acerca de qué aspectos relacionados con la supervisión faltaban en el proceso, así como qué tareas de funcionalidad móvil les gustaría ver reflejadas en la aplicación. Todo esto en pro de realizar un primer prototipo del sistema a desarrollar.

Por último se les realizó unas breves preguntas acerca de su familiarización con smartphones y tablets.

1. Cuál es su nombre.
2. Edad
3. Cargo que desempeña en la universidad.
4. Cuenta con Smartphone y/o Tablet?
5. Está familiarizado con aplicaciones móviles?

Con las respuestas obtenidas se realizó un compendio de la información que se ilustra en la tabla.

Nro. Usuario	Nombre	Edad	Perfil de usuario	Rol en el LCP
1	Ermilso Díaz Imbachi	34	Profesor Estudiante maestría en Automática	Usuario de las plantas de procesos Enseñanza en el LCP
2	Vladimir Trujillo Arias	40	Profesor Estudiante maestría en Automática Jefe del Departamento de instrumentación y control	Usuario de las plantas de procesos
3	Francisco Franco	35	Profesor	Enseñanza en el LCP
4	Yuly Bolaños	27	Laboratorista	Encargada de Mantenimiento Usuaría plantas de procesos
5	Marco Piamba	26	Estudiante	Usuario plantas LCP
6	José Ortega	30	Estudiante	Usuario plantas LCP
7	José López	22	Estudiante	Usuario plantas LCP
8	Hernesto Ruiz	26	Estudiante	Usuario plantas LCP

*Tabla. Información de los usuarios finales del sistema a desarrollar.*

Una vez definidos los potenciales usuarios de la aplicación, sus perfiles y roles dentro del LCP, se procedió a realizar un análisis contextual de las tareas.

### 3.1.3 Análisis contextual de las tareas.

El análisis contextual de las tareas está encaminado a determinar, a partir del análisis etnográfico, todas las tareas que el “sistema”<sup>15</sup> es capaz de realizar actualmente, relacionadas en el contexto específico en el cual se desarrollan [15].

El primer análisis que se realizó fue que, aunque la planta cuenta con dos transmisores de temperatura, los usuarios por lo general observan los valores de temperatura en un comienzo, cuando se pone en funcionamiento la planta, luego de esto, la supervisión del proceso se lleva a cabo desde el ordenador con el que cuenta la planta. En este sentido es importante decir que todo el proceso de supervisión se realiza desde una HMI industrial desarrollada con el software RSView32 de la empresa Rockwell Automation, esta es la principal herramienta que utilizan los usuarios de la planta para llevar a cabo todas las tareas de supervisión del proceso.

#### 3.1.3.1 RSView 32

Es un software basado en Windows de la empresa Rockwell Automation, para la creación de interfaces Hombre – máquina, cuyo objetivo es el monitoreo, la adquisición de datos y aplicaciones de control. Su uso debe llevar a cabo bajo las plataformas Windows 2000, Windows NT, Windows 9x o Windows XP.

Este software proporciona todas las herramientas necesarias para la creación de interfaces industriales: mímicos industriales en tiempo real, tendencias, graficas sobre el estado de variables, alarmas, etc. Utilizando esta herramienta se desarrolló la interfaz gráfica con la que el proceso cuenta actualmente. (Figura 3. 2). En esta se puede observar que el desarrollo se llevó a cabo sin tener en cuenta recomendación alguna de diseño, lo que hace que la interfaz sea poco usable para el usuario.

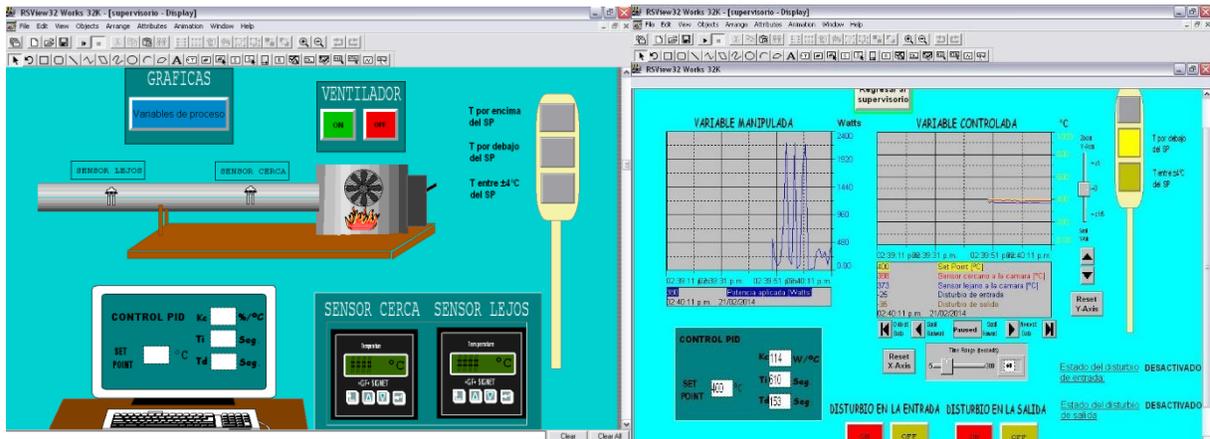


Figura 3. 2. Interfaces para la supervisión del proceso de temperatura del LCP. Ventana principal del proceso (izquierda) y ventana de graficas (derecha).  
[Fuente: Propia]

En la ventana principal del proceso (Figura 3. 2, izquierda) el usuario observa los valores actuales de temperatura, así como un mímico de la planta de temperatura, la baliza de eventos, la configuración del Set point (consigna) y el control PID para el proceso. A su vez

<sup>15</sup> El sistema es visto como el conjunto de personas, métodos, objetos y herramientas que intervienen en la consecución de las tareas de supervisión.

cuenta con un botón navegacional para trasladarse a las gráficas, cuyo objetivo es mostrar al usuario el estado de las variables del proceso (Figura 3. 2, derecha). Igual que en la anterior ventana, en el diseño de las gráficas no se ha tenido en cuenta patrones de diseño como los que se tratan en la guía propuesta. Además, se muestran los campos para configurar el set point y los valores de PID si se desea, así como el comportamiento de la variable manipulada y controlada. En este contexto el usuario se limita a monitorear el estado del proceso para corroborar su estado normal, para ello también cuenta con la baliza que indica cual es el estado del proceso cambiando el color de los indicadores si se presenta alguna anomalía con los valores del set point establecidos.

Analizando esto se pudo verificar que el usuario tiene que estar todo el tiempo frente al proceso, ya que éste no cuenta con ningún mensaje de alarma, por lo que esto se convirtió en otro de los requisitos funcionales del sistema.

El proceso de supervisión que realizaron todos los usuarios fue similar, con la salvedad de que el sistema no cuenta con una gestión de las alarmas, por lo que se convirtió en un requisito para todos, ya que si se quería realizar una gestión adecuada de estas, era necesario utilizar el software para tal fin, o simplemente utilizar herramientas manuales como lápiz y papel (objetos) para consignar los cambios importantes.

### **3.1.3.2 Experiencia en el uso de teléfonos móviles y sus aplicaciones.**

Por último, teniendo en cuenta las respuestas de las preguntas 4 y 5 que realizaron los potenciales usuarios de la aplicación, se determinó que todos tenían una afinidad con dispositivos móviles, lo cual es un aspecto muy importante debido a que el sistema a desarrollar utiliza varias de las funcionalidades que ofrece un dispositivo móvil de alta gama. A continuación se hace una pequeña descripción de cada usuario respecto a la utilización de dispositivos móviles.

**Usuario Nro. 1:** cuenta con un celular de alta gama, además de una Tablet. En su condición de ingeniero, está muy interesado con muchas novedades tecnológicas, entre ellas las relacionadas con dispositivos móviles. Utiliza su celular a menudo, en una mayor proporción que la Tablet.

**Usuario Nro. 2:** cuenta con un celular de alta gama, está muy bien familiarizado con aplicaciones móviles, utiliza algunas relacionadas con GPS y juegos. Utiliza muy a menudo su teléfono móvil.

**Usuario Nro. 3:** cuenta con un celular de gama media, el uso general que le da a su teléfono móvil es el de realizar y recibir llamadas, aunque cuenta con algunas aplicaciones en su celular que utiliza a menudo.

**Usuario Nro. 4:** cuenta con un celular de gama media, así como de una Tablet. La utilización que hace de su Smartphone es constante, sobre todo por que utiliza aplicaciones móviles relacionadas con redes sociales (WhatsApp, Facebook, Gmail, etc.), aunque la Tablet la utiliza un poco menos, cuando hace uso de ella utiliza las mismas aplicaciones, y algunas relacionadas con lectura de documentos digitales.

**Usuario Nro. 5:** cuenta con un celular de gama media, el uso que hace de su teléfono móvil es constante, ya que utiliza bastante las redes sociales, y algunos juegos en su tiempo libre, tiene un gran interés en aplicaciones móviles.

**Usuario Nro. 6:** cuenta con un celular de gama baja, solo utiliza su dispositivo móvil para realizar y recibir llamadas.

**Usuario Nro. 7:** cuenta con un celular de alta gama, utiliza a menudo las redes sociales y la navegación por internet.

**Usuario Nro. 8:** cuenta con un celular de gama media, utiliza las funcionalidades propias del celular, como la cámara, el reproductor de música, etc. No utiliza aplicaciones móviles a menudo, ni navega por internet.

Una vez se ha realizado el análisis contextual de las tareas, y antes de definir los objetivos del sistema, la guía recomienda que se realice un primer prototipo de baja fidelidad que refleje lo obtenido hasta este momento con toda la información recolectada.

### 3.1.3.3 Primer prototipo de baja fidelidad.

El objetivo de realizar este primer prototipo, es llevar a cabo un análisis de la información suministrada hasta el momento. Realizar un análisis de las primeras ideas respecto a la aplicación a desarrollar, siempre teniendo en cuenta la información recogida en el análisis etnográfico y el análisis contextual de las tareas. La técnica de prototipado ideal para esta fase es la del boceto, ya que su costo es muy bajo, y se realiza de una manera muy rápida.

En la Figura 3. 3 se ilustra el boceto desarrollado, en éste se sintetiza de una manera muy general la información recolectada en el análisis etnográfico. Para su elaboración se utilizaron 6 hojas de papel que asemejan 6 ventanas para llevar a cabo el proceso de supervisión<sup>16</sup>, además, se tuvo en cuenta el modelo de distribución de pantalla propuesto por la guía de diseño de HMIs bajo el sistema operativo Android. De manera muy general lo que se muestra en cada una de las ventanas es: un menú para el proceso de temperatura del LCP, una sección de ayuda, la ventana principal del proceso, una sección de gráficas, y por ultimo las alarmas.

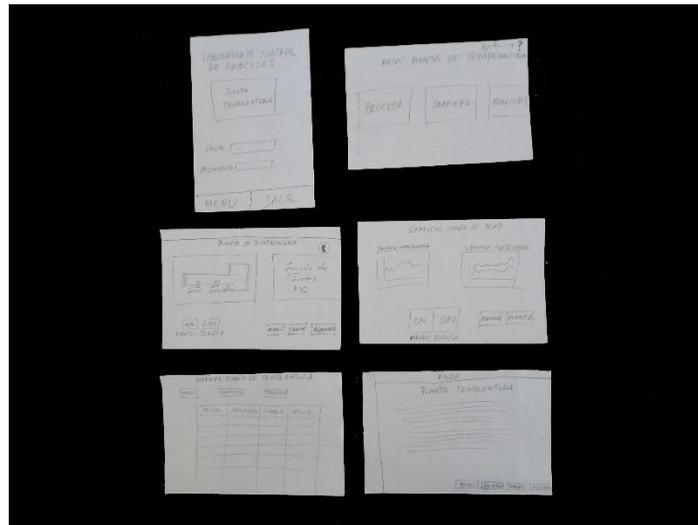


Figura 3. 3. Primer prototipo de baja fidelidad: Boceto.  
[Fuente: propia]

<sup>16</sup> La guía de diseño de HMIs bajo el sistema operativo Android recomienda que el sistema se realice en una Tablet o Smartphone en orientación landscape, de ahí que se elabore el primer boceto de esta manera.

El paso siguiente fue realizar una reunión con los usuarios finales de la aplicación, con el fin de presentar este primer prototipo y evaluarlo. El método de evaluación escogido para este fin fue el de Focus Group (grupo de discusión dirigido), ya que es ideal para aplicar en la fase de Análisis de requisitos tal como lo describe el modelo MPLu+a. Con esta técnica lo que se pretende es que los usuarios puedan dar sus opiniones acerca del prototipo y no se limiten al momento de agregar o quitar funcionalidades que deba cumplir el sistema, ya que por ser un prototipo de baja fidelidad provoca confianza en el usuario y no lo cohibe de hacer cualquier comentario. En la Figura 3. 4 se puede observar la reunión que se llevó a cabo, se tuvo una buena acogida del primer prototipo, y se discutieron algunos requisitos adicionales que estos tenían acerca del sistema a desarrollar.



*Figura 3. 4. Focus Group para la evaluación del primer prototipo.  
[Fuente: propia]*

La reunión arrojó como primer resultado una serie de requisitos que se analizaron teniendo en cuenta las recomendaciones industriales que se mencionan en la guía de diseño propuesta.

Las principales funcionalidades del sistema una vez terminada la reunión fueron:

- Control para el ingreso de usuarios (Login, password, etc).
- Menú del proceso
- Monitorización del sistema. (mímico del proceso, estado de las variables)
- Control del proceso (ingreso de valores PID, Set Point)
- Tratamiento de alarmas (Consigna de los eventos del sistema)
- Conexión a redes sociales para comunicación.
- Descripción del proceso de temperatura (ayuda).
- Consignar valores del material calentado. (industrial)
- Mensajes de notificación de alarmas.

Una vez especificado estas funcionalidades se debatieron en la reunión, sacando como conclusión, que dos de ellas no era necesario realizar en este momento. La conexión a redes sociales, aunque puede ser una buena herramienta, distraerá mucho al operario o usuario que esté utilizando el sistema, por otra parte, en estos momentos no es necesario observar cual es la cantidad de material calentado, debido a que en el LCP no se utiliza material para calentar en la planta de temperatura. Por último se concluyó que, debido a

que la aplicación tiene pocas ventanas, el menú principal no es necesario, la mejor opción es que se muestre la ventana principal del proceso una vez se haya accedido a la aplicación.

### **3.1.4 Objetivos**

Una vez culminada la reunión tras la evaluación del primer prototipo, el equipo desarrollador se encarga de definir los objetivos funcionales y de usabilidad que se trazaran para llevar a buen término la realización de la aplicación.

Es importante tener en cuenta que aunque los objetivos se determinan en esta fase, estos pueden ir variando según los nuevos prototipos que se realicen y las evaluaciones que se lleven a cabo.

#### **3.1.4.1 Objetivos funcionales**

Estos objetivos están enmarcados dentro de las funcionalidades del sistema (las tareas que debe cumplir), están estrechamente ligados con los requisitos que se definieron con los usuarios.

- Contar con un Login y password para ingresar al sistema, en pro de la seguridad de la planta de temperatura del LCP.
- Mostrar el mímico del proceso en tiempo real, al igual que la lectura de los sensores de temperatura, y el estado de las variables por medio de gráficas.
- Ofrecer al usuario la facilidad de ingresar los datos de set point y la configuración del PID para realizar el control del proceso.
- Suministrar información acerca de los eventos del proceso de temperatura (alarmas).
- Suministrar mensajes de alerta al momento de realizar cambios importantes en el proceso.
- Brindar información acerca del proceso de temperatura, para contextualizar al usuario.

#### **3.1.4.2 Objetivos no funcionales.**

- El sistema se desarrollará de acuerdo a los parámetros establecidos en la guía propuesta.
- Definir un modelo de distribución de pantalla final de acuerdo a las evaluaciones y prototipos realizados (si se requiere).
- La aplicación móvil estará soportada para todas las versiones de Android.

#### **3.1.4.3 Objetivos de usabilidad.**

- **Facilidad de aprendizaje:** El sistema debe mostrar una interfaz minimalista y agradable, elementos que sean fácilmente comprendidos y funcionalidades accesibles, siempre teniendo en cuenta las recomendaciones de diseño planteadas.
- **Consistencia:** El sistema se realizará teniendo en cuenta todas las recomendaciones y parámetros en cuanto al diseño industrial, esto con el fin de que el usuario se pueda identificar con la aplicación y pueda interactuar fácilmente con ella. Además, el usuario siempre sabrá en que parte de la aplicación se encuentra.

- **Flexibilidad:** Se evitará al máximo que el usuario tenga que escribir. Lo único que tendrá que hacer es ingresar los parámetros de set point y pid, el sistema le brindará la información requerida por medio de sus funcionalidades.
- **Recuperabilidad:** Se tendrá en cuenta los errores que el usuario pueda cometer, por ende, se mostrarán diálogos en los que se advierte al usuario de las acciones críticas del sistema (prender y apagar planta, configurar PID, set point, etc).
- **Tiempo de respuesta:** El sistema deberá de responder lo más rápido posible a las demandas de los usuarios, sin que la aplicación presente errores.
- **Disminución de la carga cognitiva:** El sistema no se sobrecargará con demasiada información en cada una de sus ventanas, para no forzar la memoria de trabajo del usuario.
- **Estética:** El sistema presentará una interfaz agradable, que será desarrollada teniendo en cuenta todas las directrices que se tienen a nivel industrial (colores, tamaños, letra, organización del contenido, etc.). Gracias a los procesos de evaluación que proporciona el modelo, la evaluación final por parte de usuarios y expertos será alta.

#### 3.1.4.4 Objetivos de accesibilidad.

En la realización de esta clase de objetivos es muy importante tener en cuenta que, ya que se está llevando a cabo un proceso industrial, las personas que lo supervisan deben ser calificadas, y por tanto no deben presentar discapacidades graves, como por ejemplo ceguera, o discapacidades mentales. Teniendo esto presente se pueden plantear algunos objetivos de accesibilidad.

- Para atender pequeños problemas visuales que no son tan notorios, se debe presentar un tipo de fuente muy legible y alto contraste en la interfaz.
- El sistema debe presentar un cambio de sonido cuando se presente alguna alarma en el sistema (situaciones de alerta).

Teniendo claros los objetivos de la aplicación a desarrollar se puede llevar a cabo la fase de diseño.

## 3.2 Diseño de la interacción.

El diseño de la interacción tiene dos componentes principales: el diseño de la actividad y el diseño de la información. El primero relacionado con la especificación funcional del sistema y el segundo con la organización del contenido, el estilo de la interfaz, y las reglas de diseño. A nivel industrial este último aspecto es muy importante, ya que es aquí, donde se tienen en cuenta todas las directrices y recomendaciones realizadas en la guía propuesta.

### 3.2.1 Diseño de la actividad.

Teniendo en cuenta las funcionalidades descritas en la anterior fase del proceso, el equipo desarrollador definió una serie de tareas para cumplir, teniendo como base las funciones definidas. Cada tarea está dividida en actividades que se mostraran de una manera detallada gracias al análisis jerárquico de las tareas descrito en el modelo MPlu+a.

En un paso posterior se tendrá en cuenta este análisis jerárquico, el diseño de la información y el modelo de navegabilidad (prototipo de story board navegacional) para

realizar una serie de evaluaciones con el fin de tener una retroalimentación con los usuarios de la aplicación.

Las funciones con sus respectivas tareas y actividades se definen a continuación:

### 3.2.1.1 Función: Control Ingreso de Usuarios.

Esta función cuenta con una tarea principal que es ingresar a la aplicación de una forma segura.

#### Tarea: Ingresar a la aplicación de forma segura:

El primer paso que debe realizar el usuario es ingresar a la aplicación, que se presenta en el menú de aplicaciones del dispositivo.

- En este momento la aplicación muestra una interfaz con dos componentes: un nombre de usuario (user name) y una contraseña (password), para que el usuario pueda registrarse y acceda a la aplicación.
- En segunda instancia el usuario ingresa su user name. Para este fin se presiona el Edit Text correspondiente y el sistema desplaza el teclado en pantalla y el usuario digita su nombre. Para el ingreso del password el proceso a seguir es el mismo.
- Por último se presiona el botón “enviar” para que los datos ingresados sean enviados y si estos son correctos, el usuario podrá acceder a la pantalla principal del supervisorio.

Cabe aclarar que por motivos de tiempo, esta tarea se realizó simulando que existe una base de datos, que corrobore tanto el usuario como la contraseña ingresada.

En la Figura 3. 5 se puede observar el análisis correspondiente a esta tarea.

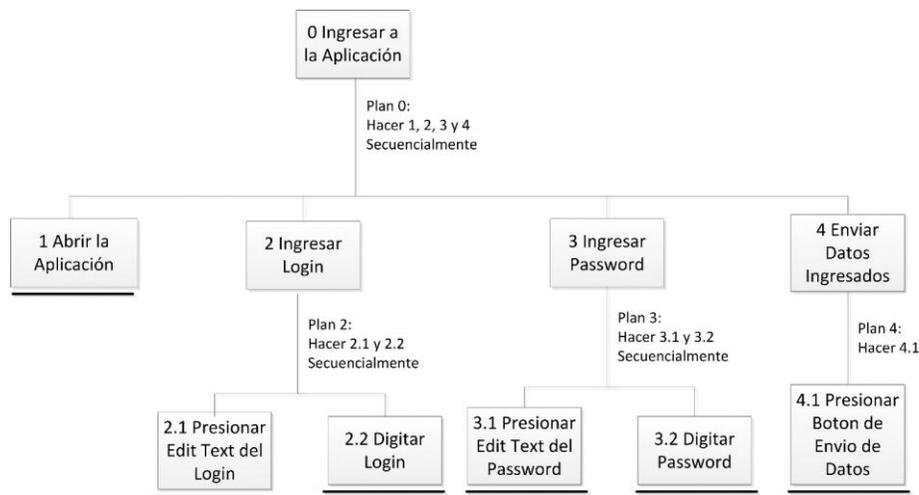


Figura 3. 5. Modelo HTA correspondiente al ingreso de a la aplicación [Fuente: propia].

Es importante tener en cuenta que realizar la tarea correspondiente a la función de control de ingreso de usuarios es obligatoria para poder realizar las demás funciones.

### 3.2.1.2 Función: Monitorización del sistema.

Para esta función se definieron las siguientes tareas:

## Tarea 1: Monitorizar la planta del proceso de temperatura.

Para esta primera tarea se debe tener en cuenta que el usuario presiona el botón de “proceso” que se encuentra en el menú de navegación de la aplicación.

- Una vez que ha ingresado a esta pantalla, el sistema muestra el mímico del proceso (animación de la planta de temperatura) con sus respectivos valores para los sensores. Con esto, el usuario puede monitorear el estado del proceso de una manera gráfica y sencilla.
- Por otra parte el sistema le muestra al usuario una representación gráfica de la baliza del proceso, para que este pueda observar si el control que está realizando la planta se encuentra dentro de los rangos normales o si por el contrario está muy por encima o por debajo de los valores deseados.

## Tarea 2: Monitorear gráficas del proceso.

Para realizar las actividades correspondientes a esta tarea el usuario debe presionar el botón “graficas” que se encuentra en el menú navegacional de la aplicación.

Una vez se ha presionado el botón de “graficas” el sistema despliega dos graficas correspondientes al seguimiento de las variables del proceso que son los valores de temperatura del proceso, la potencia entregada a las resistencias y el set point. Esto con el fin de que el usuario supervise el estado en el que se encuentran estas variables.

En la Figura 3. 6 y Figura 3. 7 se observa el análisis de tareas correspondiente.

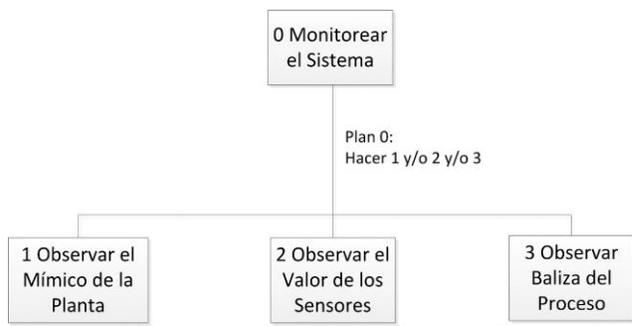


Figura 3. 7. HTA para monitorización del proceso  
[Fuente: propia]

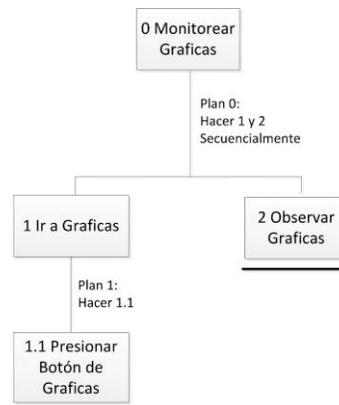


Figura 3. 6. HTA para monitorización de las gráficas del proceso.  
[Fuente: propia]

### 3.2.1.3 Función: Control del proceso

Para esta función se debe tener en cuenta un aspecto importante: Las tareas que se lleven a cabo dependerán de la orientación del dispositivo por parte del usuario.

Si la orientación es “portrait” (vertical) el sistema desplegará una ventana exclusiva para configurar los parámetros de set point y PID. Por el contrario, si la orientación es “landscape” (horizontal) el sistema desplegará la configuración de los parámetros en la misma pantalla del proceso.

### Tarea: Configuración de parámetros PID y Set point.

Para realizar estas actividades el usuario deberá presionar el botón de “Configuración” en el menú navegacional, si se encuentra en la orientación portrait, o “Proceso” si se encuentra en la orientación landscape.

- El sistema despliega una ventana donde se muestran cuatro cajas de texto para que el usuario interactúe con ellas ingresando los valores de Set point y PID deseados.
- Una vez ingresados estos parámetros el usuario presiona el botón “Enviar” para que los datos sean recibidos por el sistema, y a continuación se le notifica al usuario que la configuración se ha llevado a cabo con éxito.

En la Figura 3. 8 se observa el análisis correspondiente a esta tarea:

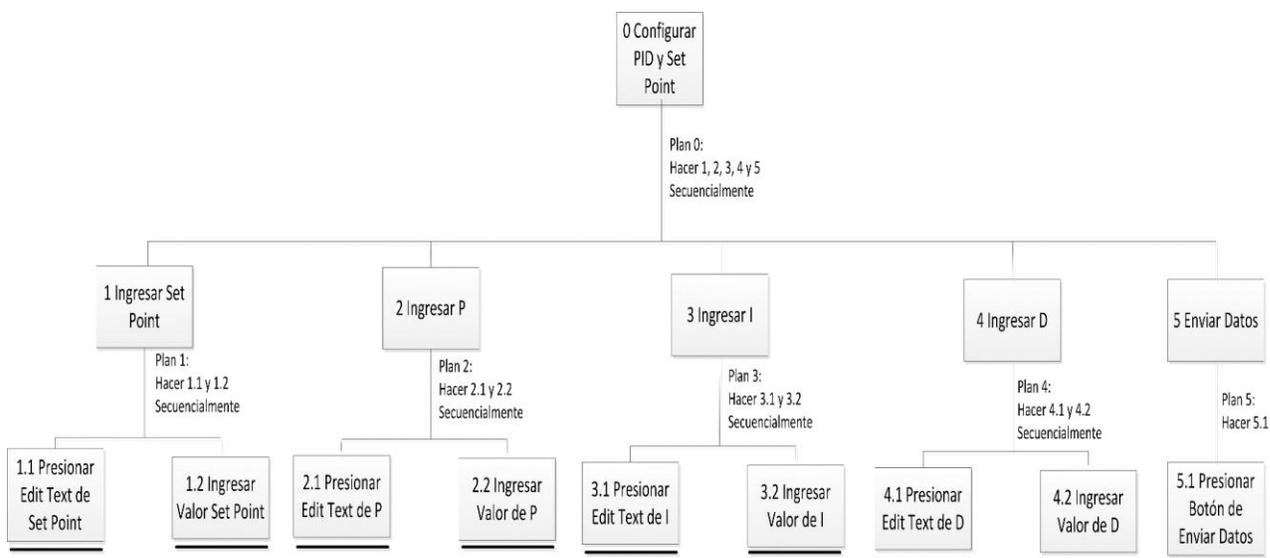


Figura 3. 8. HTA para configuración de parámetros PID y set point.  
[Fuente: propia]

#### 3.2.1.4 Función: Tratamiento de Alarmas

##### Tarea: Supervisar Alarmas.

Para llevar a cabo las actividades correspondientes a esta tarea el usuario debe presionar el botón de “alarmas” ubicado en el menú navegacional.

Una vez se ha presionado el botón el sistema despliega una ventana donde se muestra todos los eventos del proceso (estados normales del proceso, pre alarmas y alarmas) con el fin de que el usuario tome acciones correctivas si es que se necesitan, para esto el usuario puede llevar a cabo cualquier tarea que le ayude a superar estas situaciones anómalas. En la Figura 3. 9 se observa el análisis realizado para esta tarea:

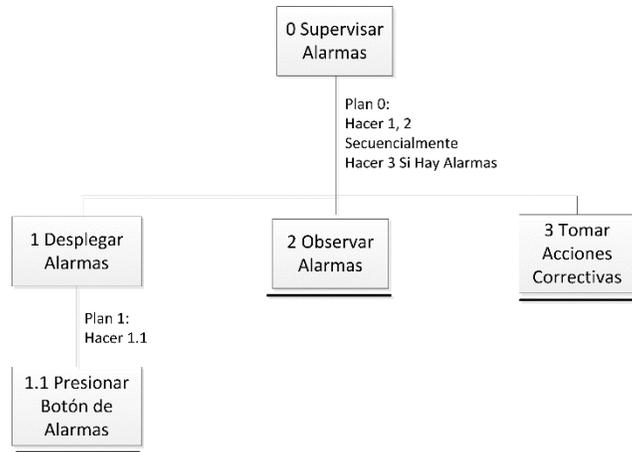


Figura 3. 9. HTA para monitorear las alarmas.  
[Fuente: propia]

### 3.2.1.5 Función: Ayuda acerca del proceso de temperatura.

#### Tarea 1: Observar la descripción del proceso de temperatura.

Para realizar esta tarea el usuario debe presionar el botón de “información” que se encuentra en el Action Bar (barra de tareas) de la aplicación.

Una vez que se presiona el botón, el sistema despliega una pantalla cuyo contenido es una breve descripción de la planta de temperatura para que el usuario pueda contextualizarse.

El análisis para esta tarea se observa en la Figura 3. 10.

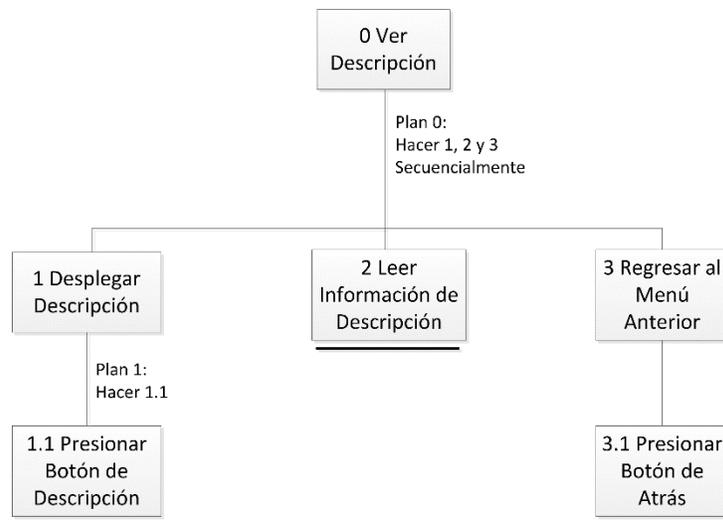


Figura 3. 10. HTA para observar descripción del proceso de temperatura.  
[Fuente: propia]

#### Tarea: Observar la ayuda acerca de la base de datos relacionada a los eventos y alarmas del proceso.

Para realizar esta tarea el usuario debe presionar el botón de “ayuda” ubicado en el Action Bar correspondiente a la ventana de las alarmas del proceso.

Una vez se presiona el botón el sistema despliega una ventana que muestra la nomenclatura utilizada en la base de datos de la aplicación, para que el usuario se contextualice de una manera más sencilla.

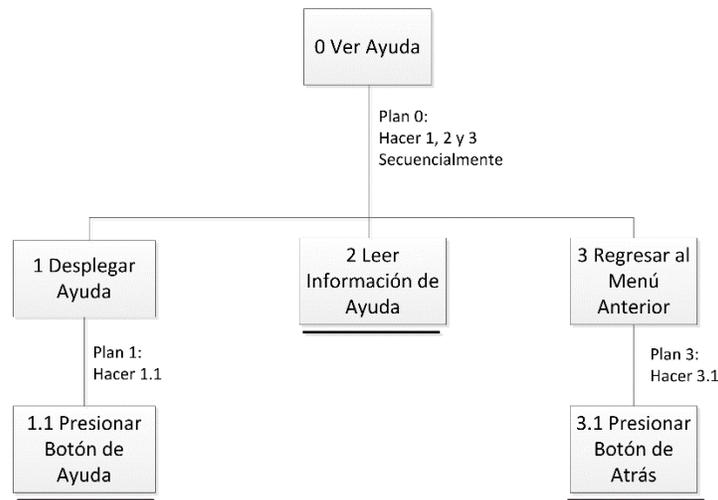


Figura 3. 11. HTA para observar la ayuda referente a los eventos y alarmas del proceso [Fuente: propia]

### 3.3 Diseño de la información.

La elaboración de la interfaz gráfica industrial, es sin duda alguna, uno de los aspectos más importantes dentro del proceso de desarrollo del sistema interactivo. Es aquí donde se debe tener en cuenta todas las recomendaciones que se plasmaron en la guía propuesta, la cual reúne tanto aspectos de diseño a nivel industrial, como los parámetros de diseño en el sistema operativo y las recomendaciones del MPUi+a.

Para la sección del acceso al sistema se tuvo en cuenta los principios de Gestalt de proximidad similitud y simetría, en donde se agruparon por un lado, dos Text Fields (campos de texto), con la propiedad Hint para mostrar un texto alternativo para guiar al usuario, y por otro lado se agruparon dos Buttons (botones) para poder ingresar al sistema o salir de este.

Para la sección del menú se tuvieron los mismos principios Gestalt de proximidad similitud y simetría con el fin de mostrar botones cuya función es la de dirigir al usuario a la ventana principal del proceso, a las gráficas, o las alarmas.

Para el diseño de la ventana principal del proceso se tuvo en cuenta la distribución de pantalla propuesta en la guía de diseño de Hmis (Figura 2. 40). Se debe tener en cuenta que en la orientación landscape, se puede aprovechar el espacio de la pantalla con la que cuenta el dispositivo.

Para representar el sinóptico del proceso, se utilizó una representación de la planta que se acomoda a las recomendaciones de diseño de la guía. Ver Figura 3. 13. Este cuenta con una animación que asemeja el proceso de temperatura que se lleva a cabo al interior de la planta. El aire a temperatura ambiente ingresa por la cámara por medio de un ventilador y es calentado por 3 resistencias de calefacción, luego el aire es despedido por el tubo cilíndrico, es aquí donde se encuentran dos sensores de temperatura que hacen la lectura de los valores de la variable controlada [44].

En cuanto a los mandos de uso ocasional para la planta de temperatura, se utilizaron dos botones de mando relacionados con el encendido, estos tienen que cumplir con el principio de finalidad descrito en la guía propuesta. Para ahorrar espacio en la pantalla del dispositivo decidió que los elementos de mando, fueran también indicadores (pilotos), esto con el fin de que cuando el usuario realizara alguna acción de encendido o apagado, se produciría un efecto de luz intermitente, para saber el estado en el que se encuentra la planta.

Para el ingreso de los datos de set Point y PID con los que cuenta el proceso de temperatura, se tuvieron en cuenta de nuevo los principios de Gestalt para ubicar cuatro textview con sus respectivos EditText, acompañados de un botón para realizar la acción de enviar. Ver Figura 3. 14.

Debido al espacio con el que se cuenta en la orientación landscape, se decidió que la baliza del proceso se ubicará al lado del sinóptico de la planta.

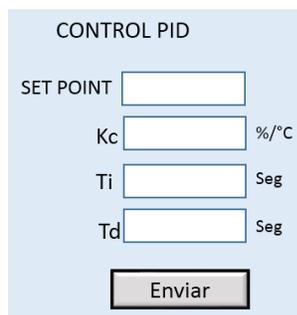


Figura 3. 14. Ingreso de datos para el control de proceso de temperatura. [Fuente propia]



Figura 3. 13. Representación de la planta de temperatura. [Fuente: propia]



Figura 3. 12. Representación de la baliza del proceso de temperatura. [Fuente: propia]

Para realizar el menú de navegación principal del proceso se utilizaron tres botones cuya principal función es brindar la navegación en la aplicación móvil.

En esta primera ventana se utilizaron 3 botones:

- Inicio: que dirige al usuario al menú principal de la aplicación.
- Graficas: dirige al usuario a las gráficas del estado de las variables del proceso.
- Alarmas: dirige al usuario a la base de datos donde se encuentra el histórico de los eventos del proceso.

Para la interfaz que corresponde a las gráficas donde se muestra el estado de las variables del proceso, en este caso temperatura (variable controlada) y potencia en las resistencias (variable manipulada), se optó por utilizar gráficos de líneas.

Al igual que en la ventana de proceso principal, en esta ventana de graficas se ubicaron los dos botones de mando de uso ocasional (encendido y apagado), y tres botones de navegación por la aplicación.

Por ultimo para el diseño de la tabla de eventos y alarmas de la aplicación se tuvieron en cuenta las reglas de diseño presentadas en la guía propuesta. Debido a que se utiliza un ScrollView para la tabla, se optó por ubicar los botones de navegación en la parte superior de esta para no entorpecer el manejo de la aplicación por parte del usuario.

Teniendo en cuenta los anteriores aspectos se realizó el **primer prototipo digital** de las interfaces (Figura 3. 15), para ello se utilizó la herramienta de diseño **Pencil**. Esta es una herramienta OpenSource gratuita para crear diagramas y bocetos GUI, es fácil de utilizar y ofrece una serie de características para realizar prototipos rápidos para ser evaluados. Es importante tener en cuenta que en primera instancia se desarrollaron las 4 interfaces más importantes del proceso: la ventana para el ingreso de los usuarios, la ventana principal del proceso, la de gráficas para el estado de las variables y por último la de alarmas y eventos.

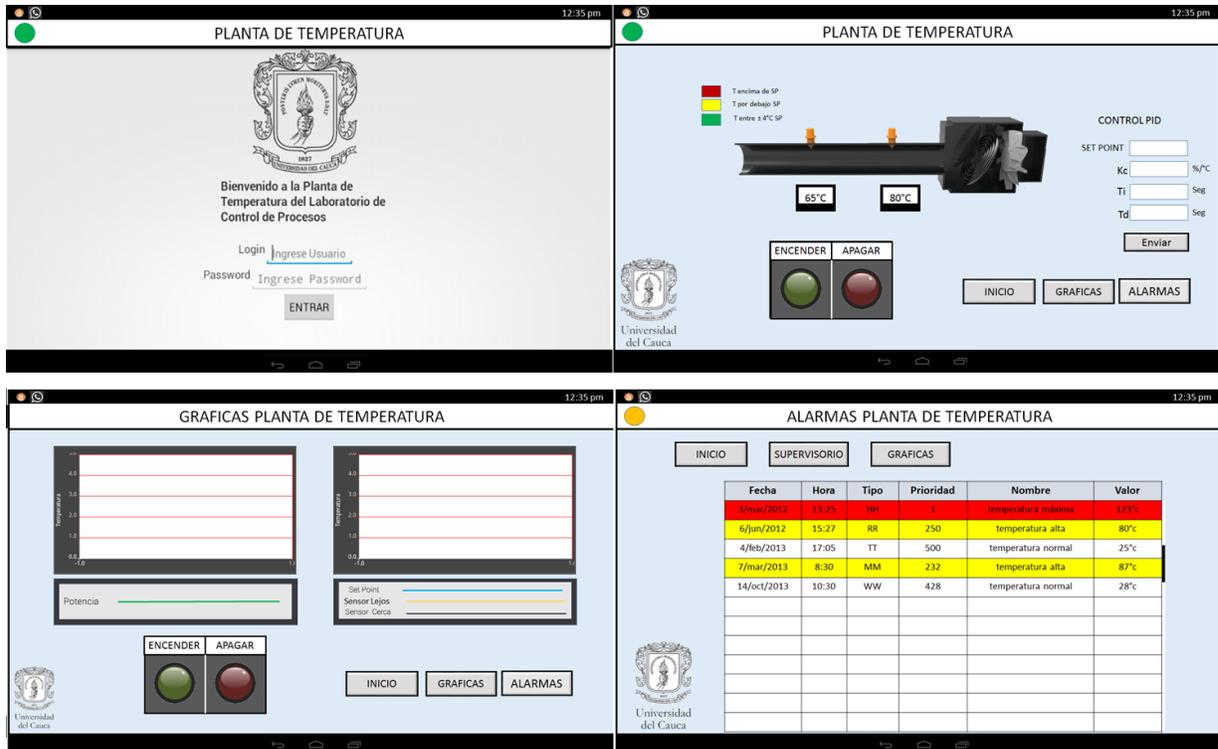


Figura 3. 15. Primera maqueta digital del sistema. Desarrollada con la herramienta software Pencil Project. Utilizada para evaluar el diseño de la información. [Fuente: propia]

### 3.3.1.1 Evaluación de las interfaces graficas

Debido a que el diseño de la interfaz de usuario para un proceso industrial es muy importante, y a que se está realizando el sistema interactivo en un dispositivo móvil, se llevó a cabo una evaluación de esta primera maqueta por parte del desarrollador en aplicaciones móviles, el ingeniero Darío Chamorro<sup>17</sup>. Teniendo en cuenta que el único objetivo de esta evaluación, sería el de valorar el diseño y la distribución de las 4 ventanas sin olvidar las recomendaciones de diseño propuestas en la guía.

El método de evaluación utilizado fue el de “Inspección de Estándares” (Tabla 2. 11), ya que es un buen método utilizado principalmente para evaluar las interfaces gráficas,

<sup>17</sup> Debido a la experiencia en el desarrollo de aplicaciones móviles por parte del ingeniero, se acudió a él para realizar la evaluación de esta primera maqueta digital.

basándose en estándares. Para esto, el ingeniero desarrollador Android, cuenta con todos los parámetros establecidos en la página de desarrolladores del sistema operativo referentes al diseño de las interfaces y los parámetros de diseño propuestos en la guía.

Los principales aspectos que se recogieron una vez realizada la evaluación fueron:

- En primera instancia las interfaces no cuentan con un buen look & feel<sup>18</sup>, lo que hace que, por ejemplo los botones no se vean como elementos interactivos para el usuario (clicqueables). Otro aspecto es el color de fondo, Android cuenta con muchas herramientas en este sentido, como aplicar un gradiente, que hará que su estilo mejore enormemente.
- El diseño de los botones de mando no es coherente con el fondo de la aplicación.
- La organización de los EditText y los textView no es la adecuada, deben tener un mejor diseño para que parezcan elementos con los cuales el usuario pueda interactuar.
- El menú de navegación principal debería ir resaltado en la interfaz ya que es una parte importante dentro del sistema. Es recomendable utilizar metáforas para este fin.
- Las cajas de los valores de los sensores al encontrarse en la parte inferior de la planta, hacen que dé lugar a una lectura no tan legible ya que las palabras “encender y apagar” tienen un diseño muy parecido.
- Se debe evitar al máximo el uso de letras MAYUSCULAS.

Todos los datos recogidos en esta evaluación se tuvieron en cuenta por parte del equipo desarrollador para realizar las correcciones necesarias. De esta manera, se procedió a realizar una segunda maqueta digital para evaluar de nuevo el diseño de la información, pero esta vez con los usuarios finales del sistema. Para esta segunda maqueta se utilizó la herramienta de diseño INKSCAPE, ya que es una herramienta software ideal para realizar diseños con más detalle, ya que es un editor de gráficos vectoriales. En la Figura 3. 16 se ilustra las interfaces obtenidas con esta herramienta software.

---

<sup>18</sup> El término “look & feel” hace referencia a la “presentación e interacción” de la interfaz de usuario. La presentación comprende la disposición de los elementos de la interfaz (gráficos, texto, colores, etc.) y la interacción hace referencia al procesamiento y rendimiento.



### 3.3.2 Story board navegacional.

Una vez realizada la pequeña reunión (focus group), y teniendo la seguridad de que la mayoría de aspectos acerca del diseño de la información de la aplicación habían sido abordados, el paso a seguir por parte del equipo desarrollador fue realizar un prototipo con la técnica de story board navegacional, donde se muestra la navegabilidad del sistema para llevar a cabo la consecución de las tareas por parte de los usuarios. Teniendo en cuenta la información adquirida en el diseño de la actividad (análisis jerárquico de las tareas). Este prototipo se hizo en papel debido a que los aspectos de diseño de la información (el estilo de la aplicación) ya habían sido evaluados anteriormente, y con este prototipo se ahorra tiempo y recursos. En la figura se ilustra el story board (modelo de navegabilidad), gracias a este prototipo se pueden simular las actividades que el usuario lleva a cabo dentro de la aplicación.

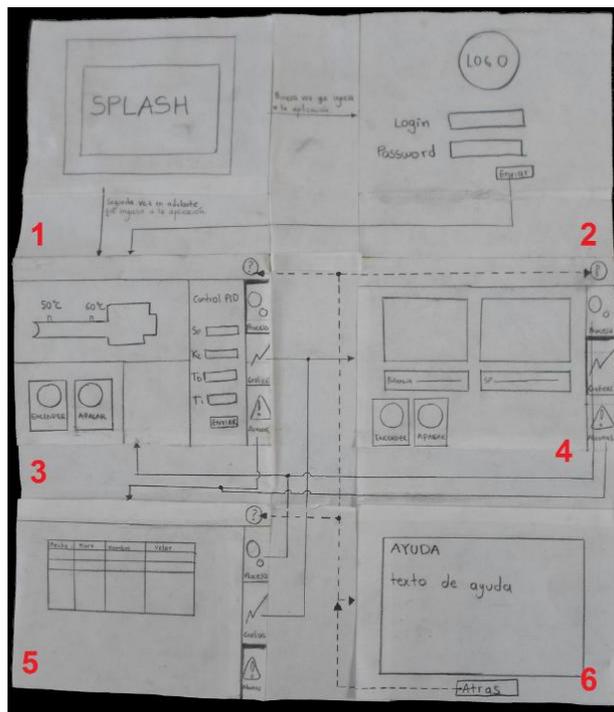


Figura 3. 17. Prototipo de Story Board (modelo de navegabilidad.)  
[Fuente: Propia]

El modelo presentado consta de 6 recortes que simulan las pantallas con las que el usuario se encontrara en la aplicación final. En la primera (1) de ellas se encuentra el splash de la aplicación, que es la pantalla de presentación, de ahí el usuario es desplazado a la pantalla para acceder a la aplicación por medio de un password (2). Una vez que ha accedido, se muestra la pantalla principal del proceso (3), en esta se muestran los botones de mando del proceso (encendido y apagado de la planta), el espacio para la entrada de datos del proceso: configuración de los parámetros de PID y el setpoint. Por último se encuentra el menú de navegación para que el usuario pueda trasladarse a través de la aplicación.

La pantalla de graficas (4) hace referencia al estado de las variables (temperatura y potencia) a través del tiempo, también se puede observar el menú de navegación descrito anteriormente. Una vez el usuario decida trasladarse a la pantalla relacionada con los

eventos del proceso (alarmas), se encontrara con la base de datos desarrollada, aquí se muestra los eventos a través del tiempo (normal, pre-alarmas y alarmas). En la última pantalla (6) se observa la ventana de ayuda del proceso, esta es una pequeña descripción de la planta de temperatura para contextualizar al usuario, o la nomenclatura utilizada para la base de datos de los eventos del proceso.

### **3.3.2.1 Evaluación del story board navegacional.**

La evaluación para este segundo prototipo fue realizada por los usuarios finales de la aplicación y el método utilizado para este fin fue el de recorrido “cognitivo con usuarios”. En primera instancia se explicó a cada usuario (en sesiones separadas) de que se trataba la evaluación, pidiéndole que observara detenidamente el story board. Luego, se le solicitaba que llevara a cabo una serie de tareas con la herramienta suministrada, estas tareas fueron fundamentadas en el análisis jerárquico que se realizó en el diseño de la actividad. Las tareas que el equipo desarrollador planteó fueron las siguientes:

#### **Ingresar a la aplicación:**

- Ingresar login.
- Ingresar Password.
- Enviar datos ingresados.

#### **Empezar proceso de supervisión:**

- Pulsar botón de encendido de la planta.
- Observar mímico del proceso y valor de los sensores que arroja el sistema.
- Observar el funcionamiento de la representación de la baliza con la que cuenta el sistema.
- Pulsar cualquier botón en el menú de navegación que lo desplace a cualquier parte de la aplicación.
- Escoger la opción de ayuda dispuesta en el ActionBar de la aplicación.

#### **Configurar parámetros de Set point y PID:**

- Ingresar valor de Set point para la temperatura en la planta.
- Ingresar valores de Configuración de los parámetros PID para controlar la planta.
- Enviar los datos introducidos.
- Observar la respuesta del sistema al recibir los datos introducidos.
- Pulsar cualquier botón en el menú de navegación que lo desplace a cualquier parte de la aplicación.
- Escoger la opción de ayuda dispuesta en el ActionBar de la aplicación.

#### **Monitorear las variables del proceso mediante gráficas:**

- Desplegar. la ventana de graficas mediante el menú de navegación.
- Observar parámetros de potencia, set point y temperatura mediante graficas de líneas.
- Pulsar cualquier botón en el menú de navegación que lo desplace a cualquier parte de la aplicación.
- Escoger la opción de ayuda dispuesta en el ActionBar de la aplicación.

### Acción de ayuda:

- Pulsar el botón de ayuda en el action bar
- Desplazarse a la ventana de ayuda.
- Pulsar el botón de atrás para regresar a las ventanas principales de la aplicación.

### Monitorear eventos del proceso (Valores de temperatura, pre-alarmas y alarmas):

- Desplegar la ventana de alarmas desde el menú de navegación.
- Revisar eventos del proceso (Valores de temperatura, pre-alarmas y alarmas).
- Pulsar cualquier botón en el menú de navegación que lo desplace a cualquier parte de la aplicación.
- Escoger la opción de ayuda dispuesta en el ActionBar de la aplicación.

Con esta serie de tareas el usuario interactuaba con el story board navegacional con el objetivo de cumplirlas (Figura 3. 18).



*Figura 3. 18. Recorrido cognitivo por parte de los usuarios.  
[Fuente: Propia]*

El resultado de esta evaluación fue positivo en el sentido en que todos los usuarios realizaron todas las tareas satisfactoriamente. Esto es debido a que en el diseño del story board se tuvieron en cuenta todas las sugerencias generadas en el diseño de la información (interfaces del proceso), además del análisis de las tareas realizado en el diseño de la actividad. Las metáforas utilizadas en el prototipo fueron adecuadas y no generaron confusión. La entrada de los datos están ubicados de una manera tal que, al usuario no le genera problemas de interacción. Y la ayuda es mostrada de una manera eficiente.

Un aspecto que se debe tener en cuenta que resaltaron los usuarios 1, 2, 4 y 5, es la navegación en la orientación portrait (vertical). Tema que se tuvo en cuenta en la fase de implementación de la aplicación. Esto debido a que el diseño no cambia en gran medida teniendo en cuenta las recomendaciones mostradas en la guía de diseño.

### **3.4 Implementación.**

Teniendo en cuenta el story board navegacional y el diseño de las interfaces de la aplicación (diseño de la información), el paso a seguir es llevar a cabo la implementación del proyecto.

La implementación se realizó bajo el IDE (entorno de desarrollo integrado) eclipse ya que es la herramienta más completa para el desarrollo de aplicaciones en Android.

El primer paso que se llevó a cabo teniendo el diseño de las interfaces, fue desarrollar propiamente todas las interfaces de la aplicación (layouts). Teniendo en cuenta que la aplicación sería instalada en Tablets y Smartphones con el sistema operativo Android, los cuales son dispositivos que cuentan con diferentes tamaños de pantalla (10", 7", 4.7", 3.7", etc.) se optó por utilizar fragments<sup>19</sup> para que la aplicación se acomodara a todos estos tamaños.

El resultado del desarrollo de los layouts se muestra en la Figura 3. 20 (orientación Landscape) y Figura 3. 21 (orientación Portrait). En Landscape se observan las 6 pantallas correspondientes al sistema para la supervisión del proceso de temperatura y en portrait se ilustran 8 pantallas. La primera hace referencia al Splash (presentación) de la aplicación, que se ha diseñado de una manera sencilla, muy minimalista. La segunda corresponde a la pantalla para el ingreso a la aplicación, el diseño es el mismo que el de la primera maqueta digital.

La ventana principal del proceso se encuentra en la tercera pantalla de la Figura 3. 20, en esta se observa que los elementos están distribuidos de una uniforme, el mímico de la planta y los botones de mando se diferencian de una manera clara, la utilización de metáforas ayuda al usuario a comprender de una mejor manera la navegación por el proceso. Es importante anotar que se tuvieron en cuenta los dos aspectos definidos en la segunda evaluación de las interfaces (sección 3.3.1.2) enfocados hacia la mejora del diseño de las interfaces de la aplicación. Teniendo en cuenta esto, es que el menú de navegación se encuentra ubicado de manera vertical en la pantalla, con el fin de que haya un acoplamiento cuando se cambie de orientación landscape a portrait, y para que la navegación sea uniforme entre todas las ventanas. Este aspecto se considera relevante para realizar un rediseño de la distribución de pantalla propuesto en la Figura 2. 40.

Para la orientación portrait (Figura 3. 21), se realizó otra distribución de los elementos con el fin de brindar una mejor percepción del proceso al usuario, el menú de navegación se encuentra ubicado en la parte inferior de la ventana ya que en esta orientación la manipulación del dispositivo por parte del usuario se realiza desde esta posición. En la figura se observa también que debido al menor espacio con el que se cuenta en esta orientación, la configuración de los parámetros PID y Set Point desaparecen y se da paso

---

<sup>19</sup> Los fragments son un fragmento de una actividad que cuenta con su propio layout y su propio ciclo de vida, por tanto es una poderosa herramienta cuyo objetivo es que el desarrollador pueda adaptar la aplicación a una gran variedad de dispositivos sin necesidad de grandes líneas de código. Logrando así un ahorro de tiempo y recursos.

a un nuevo botón de navegación (Control PID), el cual conduce al usuario a una nueva ventana (cuarta pantalla de la figura). Esto se hace siguiendo las recomendaciones de la guía de diseño propuesta.

En las figuras Figura 3. 20 y Figura 3. 21 se observan también las interfaces relacionadas con las gráficas para la supervisión del proceso, se diferencia de una manera clara los colores para las variables del proceso, así como el menú de navegación y la ayuda. Cada una de las gráficas cuenta con una leyenda que hace referencia a la variable manipulada (potencia) y la variable controlada (temperatura).

La ventana relativa a los eventos del proceso (alarmas) se presenta de una manera similar en las dos orientaciones, con la salvedad de que en la orientación portrait, se hace necesario ocultar dos columnas de la base de datos: tipo y prioridad, esto debido a que el espacio disponible es muy poco, lo que conlleva a que la lectura de 6 columnas sea engorrosa en esta orientación.

Con la base de datos implementada se realizó una adquisición de los valores de temperatura del proceso, clasificándolos en 5 categorías: temperatura normal, temperatura alta, temperatura máxima, temperatura baja y temperatura mínima. Ver Figura 3. 19. Para el caso de estudio se establecieron estos valores con el fin de representar situaciones de alarmas dentro del proceso.

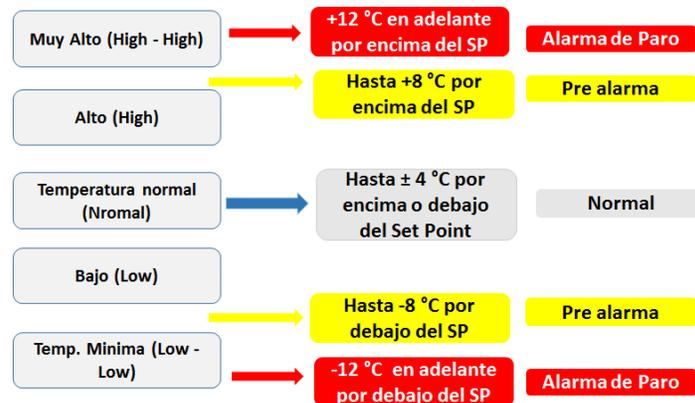


Figura 3. 19. Valores correspondientes a las alarmas del proceso.  
[Fuente: propia]

En la ventana de alarmas también se debe tener en cuenta que se cuenta con otro botón en el Action Bar, el objetivo de este, es brindar información al usuario acerca de la nomenclatura utilizada en la base de datos. Así por ejemplo las siglas HH (High - High) significan temperatura muy alta y LL (Low – Low) temperatura muy baja. Esta ventana se observa en las últimas ventanas de las Figura 3. 20 y Figura 3. 21.

Con el diseño de todas las interfaces del sistema el siguiente paso fue implementar todo el código de la aplicación teniendo en cuenta todas las funcionalidades obtenidas en las fases anteriores, para esto se tuvo en cuenta el análisis jerárquico de las tareas, los resultados de la evaluación del Story Board navegacional, así como el diagrama de casos de uso que se describe en el Anexo B del documento.

El resultado de esta fase de implementación es un prototipo Software, que será evaluado en la siguiente fase del proceso.

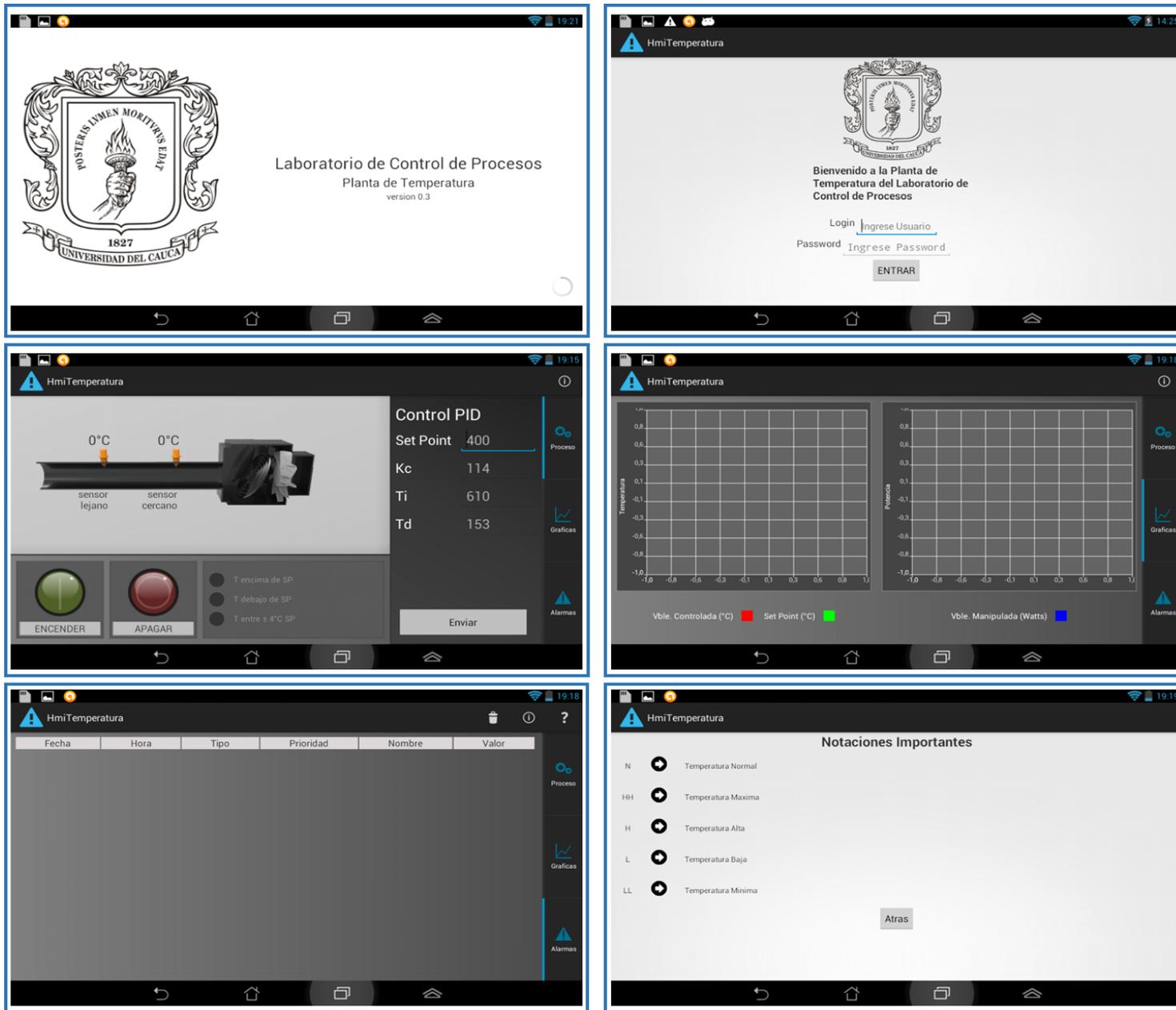


Figura 3. 20. Interfaces en la orientación Landscape implementadas en el IDE Eclipse, para la aplicación móvil de supervisión del proceso de temperatura.

[Fuente: propia]

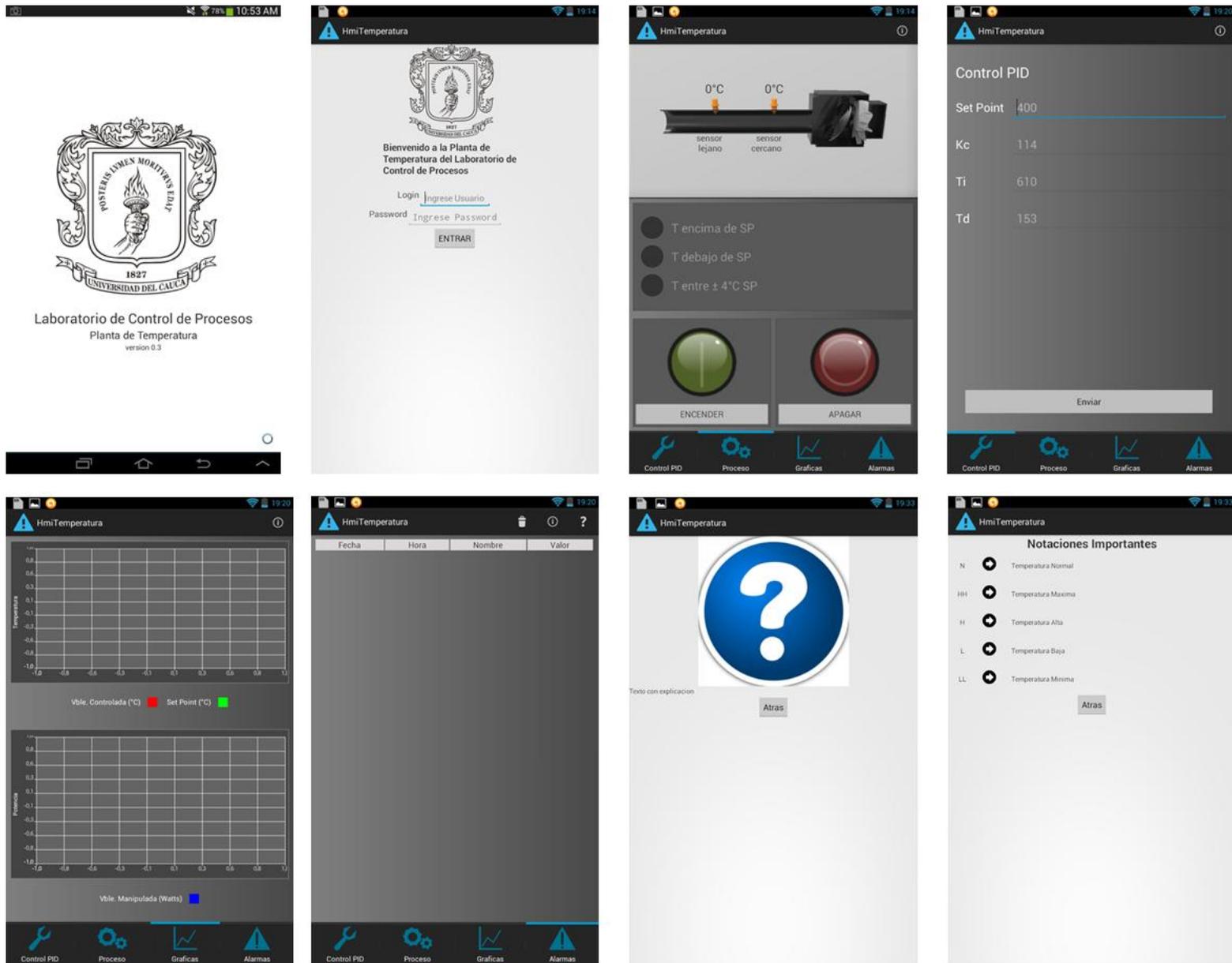


Figura 3. 21. Interfaces en la orientación Portrait implementadas en el IDE Eclipse, para la aplicación móvil de supervisión del proceso de temperatura.  
[Fuente: propia]

### 3.5 Evaluación del prototipo Software.

La evaluación del prototipo software finalizado, se llevó a cabo en primera instancia por el experto en usabilidad: el Magister Andrés Solano. Teniendo como principal técnica la evaluación heurística para el contexto de las aplicaciones móviles, descritas en la guía de diseño propuesta. Para llevar a cabo esta evaluación se contó con un Checklist<sup>20</sup> desarrollado en [36], el cual está enfocado hacia la evaluación de la usabilidad en aplicaciones para Smartphones Touchscreen.



Para realizar esta evaluación se le hizo entrega del prototipo final al ingeniero experto, describiéndole de una manera general todas las funcionalidades de la aplicación, para que se pudiera contextualizar con el proceso. Teniendo como herramienta el checklist descrito anteriormente, llevó a cabo la evaluación de usabilidad por espacio de 1 hora, realizando las observaciones pertinentes respecto al cumplimiento de cada una de las heurísticas de evaluación. El checklist con las observaciones realizadas se consignó en el anexo C del documento.

Todas las observaciones generadas se tendrán en cuenta para un posterior desarrollo del prototipo.

Por otra parte, también se llevó a cabo la evaluación del prototipo software por parte de los usuarios finales del sistema.

Se realizó con cada uno de ellos una sesión de supervisión con el prototipo finalizado. Teniendo en cuenta que fue desarrollado según sus necesidades y modelo mental, y que ya se habían realizado evaluaciones anteriormente, la realización de las tareas se llevaron a cabo de una manera eficiente; los usuarios no tuvieron problemas para llevar a cabo las tareas de supervisión. Sin embargo se pudo notar que dos de los usuarios tuvieron problemas en la navegación debido a que cuando navegaban entre pestañas, ocasionalmente presionaban el botón de atrás para retornar a su posición anterior, esto

---

<sup>20</sup> En el Anexo C del documento se detalla el Checklist utilizado para la evaluación Heurística, con las observaciones realizadas por el experto en usabilidad.

ocasionaba que se salieran de la aplicación. Aun así los usuarios se mostraron complacidos con el prototipo software. Además destacaron la importancia y utilidad que podría tener este prototipo en industrias reales.

Para evaluar a los usuarios sobre su experiencia con el prototipo software, se utilizó la técnica de “cuestionarios tipo indagación” en la cual se le hizo entrega a cada uno de ellos de un cuestionario que tenía preguntas acerca de aspectos de usabilidad. Cada uno respondió teniendo como consigna que 1 significa: “Estoy totalmente de acuerdo” y 5 significa: “Estoy en total desacuerdo” y su calificación podía estar en esa escala. Los resultados del cuestionario tienen un puntaje máximo de 86.46 y un puntaje mínimo de 17.01 y para su cálculo se utilizó una herramienta denominada MUMMS (Measuring the usability of multi – media Systems) facilitada por el grupo de investigación irlandés HFRG (Human Factor research group). En la que se mide la usabilidad de un producto multimedia a través de 6 aspectos de medición: Efectividad, control, Eficiencia, utilidad, facilidad, emoción (Figura 3. 22). En esta herramienta se ingresaron los datos obtenidos por cada cuestionario y se llevaron a cabo los cálculos automáticamente. Los resultados obtenidos se ilustran en la Tabla 3. 1. En el anexo digital se adjunta la herramienta utilizada, con los cálculos obtenidos.



Figura 3. 22. Estructura de usabilidad de MUMMS] [Fuente: [45] ]

No. De Usuario	Atracción	Control	Eficiencia	Utilidad	Aprendibilidad	Emoción	Puntaje Global
1	86,46	86,46	86,46	86,46	86,46	86,46	86,46
8	86,46	86,46	86,46	86,46	86,46	86,46	86,46
5	75,61	84,29	82,12	82,12	82,12	84,29	81,76
4	86,46	77,78	77,78	82,12	82,12	79,95	81,03
7	82,12	79,95	79,95	79,95	82,12	79,95	80,67
2	86,46	86,46	84,29	84,29	82,12	77,78	83,56
3	84,29	75,61	86,46	75,61	86,46	75,61	80,67
6	86,46	86,46	86,46	86,46	86,46	86,46	86,46
						<b>Promedio</b>	<b>83,38</b>

Tabla 3. 1. Resultados obtenidos con la herramienta MUMMS para evaluar usabilidad con los usuarios finales. [Fuente: propia]

En la Tabla 3. 1 se observa que se obtuvo un buen promedio en el puntaje global de la evaluación: 83.38, lo que indica que el prototipo software tuvo una buena aceptación entre los usuarios y muestra una buena medida de la usabilidad. Sin embargo, para una posterior

versión de la aplicación se tendrán en cuenta algunas observaciones y recomendaciones para aumentar los niveles de emoción por parte de los usuarios 2 y 3, entre las que se encuentran: utilizar variables de ingeniería para representar los datos mostrados, incluir más información acerca del proceso, y mostrar el estado de la conexión con el proceso.

Una conclusión muy importante que se debe tener en cuenta en este punto es que, debido a todas las evaluaciones realizadas, a las observaciones, recomendaciones y análisis que llevaron a cabo tanto expertos como usuarios finales, es que se planteó una redistribución de la pantalla en la orientación Landscape tal y como se presentó originalmente en la guía propuesta, lo que hace inferir que el MPlu+a es una herramienta potente en el diseño de sistemas interactivos. En la Figura 3. 23 se observa la nueva distribución propuesta gracias a la aplicación de la guía. La principal diferencia está en la ubicación del menú navegacional, se comprobó que con esta distribución el usuario se siente mucho más cómodo que con la anterior propuesta. Por otra parte, la distribución para el diseño de pantalla en la orientación portrait fue aceptada de una muy buena manera por parte de los usuarios, por lo que no se hace necesario realizarle ajustes.

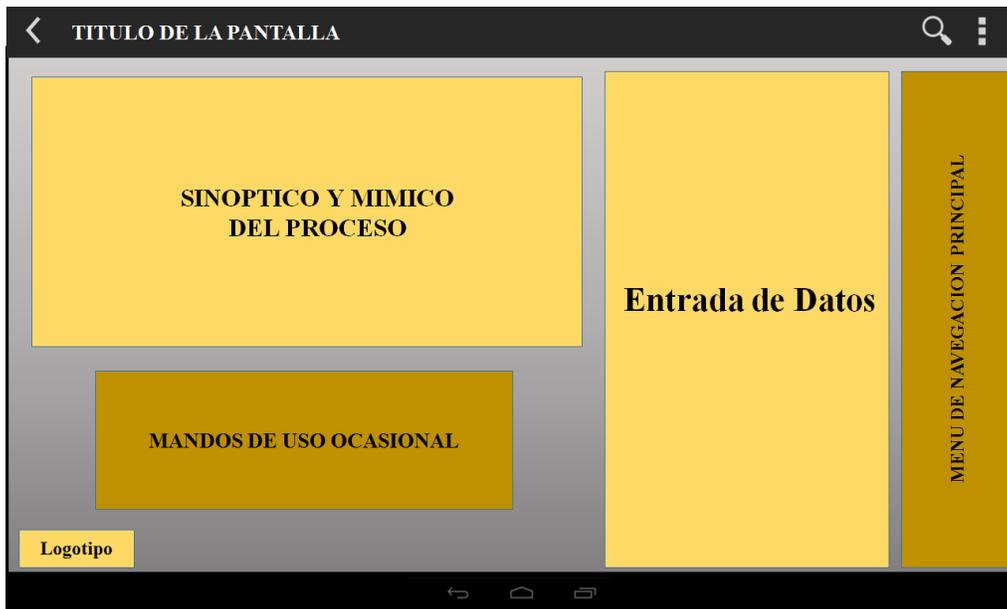


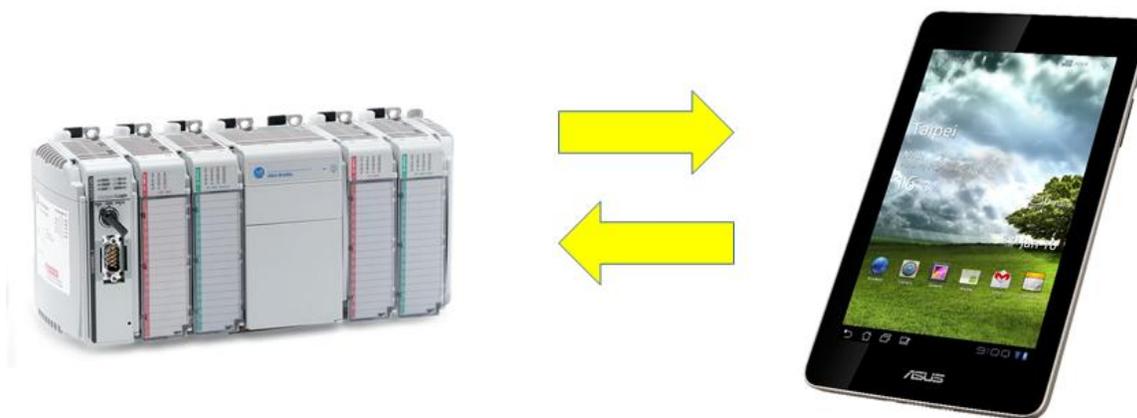
Figura 3. 23. Distribución de pantalla final propuesta luego de aplicar la guía de diseño [Fuente: propia]

### 3.6 Lanzamiento.

Para el lanzamiento de la aplicación se tuvo en cuenta a la laboratorista del LCP, a la cual se le hizo entrega del .apk desarrollado, para que los interesados en el proceso de supervisión de la planta puedan llevarlo a cabo desde su teléfono móvil. Por otra parte se espera que la aplicación pueda estar alojada en algún servidor del departamento de instrumentación y control para que pueda ser descargada e instalada en cualquier dispositivo móvil Smartphone o Tablet.

## 4. Mecanismo de comunicación entre una HMI Android y un proceso gobernado por un PLC.

Un objetivo importante en el desarrollo del proyecto es la comunicación bidireccional que se debe implementar entre el PLC o dispositivo de control y el dispositivo móvil utilizado para la supervisión remota del proceso. En este sentido se ha planteado una solución de comunicación teniendo en cuenta herramientas como el estándar OPC, el protocolo de comunicación TCP/IP y el software LabVIEW. Es importante aclarar que la solución presentada se realizó con fines académicos ya que para un nivel alto de optimización a nivel industrial se debe recurrir a una herramienta software diferente a LabVIEW.



*Figura 4. 1. Objetivo principal de comunicación. Envío y recepción de datos de una manera bidireccional entre el dispositivo de control y el dispositivo móvil.*

Uno de los principales retos en la automatización a nivel industrial es la comunicación de dispositivos, controladores y/o aplicaciones de diferentes fabricantes para lograr una interoperabilidad ilimitada sin tener el problema de los protocolos propietarios. En este contexto una solución establecida y aceptada es el estándar OPC que facilita la integración de aplicaciones, desde la adquisición de datos hasta la generación de informes a medida de cada departamento [46].

### 4.1 Estándar De Comunicación OPC.

OPC (OLE<sup>21</sup> for Process Control) es el estándar de interoperabilidad para el seguro y confiable intercambio de datos, en la automatización industrial y en otros campos. Es independiente de la plataforma y asegura el flujo continuo de información entre dispositivos de múltiples fabricantes. Las especificaciones del estándar fueron desarrolladas por la industria de proveedores, usuarios finales y desarrolladores de software, estas definen la interfaz entre clientes y servidores, incluyendo acceso a datos en tiempo real, monitoreo de alarmas y eventos, acceso a datos históricos y otras aplicaciones. En la actualidad su mantenimiento y desarrollo es responsabilidad de la fundación OPC.

---

<sup>21</sup> Enlace e incrustación de objetos, protocolo desarrollado por Microsoft.

El estándar fue realizado en 1996, con el propósito de abstraer protocolos específicos de PLC (como Modbus, Profibus, etc.) a una interfaz estandarizada permitiendo que los sistemas HMI/SCADA interactúen con un “medio – hombre”, en donde se pueden convertir peticiones de lectura/escritura genéricas de OPC en peticiones específicas para dispositivos y viceversa [58]. En un principio fue restringido a ser utilizado únicamente en el sistema operativo Windows.

#### 4.1.1 Arquitectura OPC.

La arquitectura OPC puede ser representada por un modelo cliente – servidor, en éste, se ofrece una plataforma para extraer datos de una fuente y de esa manera a través de un servidor comunicarlos a cualquier aplicación cliente de manera estándar. Los componentes OPC del modelo son:

- Cliente OPC: puede comunicarse con cualquier servidor OPC
- Servidor OPC: es una aplicación que permite el acceso a objetos ítems de un sistema automatizado, desde una aplicación cliente OPC.



Figura 4. 2. Esquema de una Arquitectura OPC  
[Fuente: [59]]

En la Figura 4. 2 se ilustra un esquema de la arquitectura OPC, los elementos son:

- Comunicaciones OPC cliente – OPC servidor.
- Servidor OPC: tiene como función traducir los datos provenientes de la fuente de datos a un protocolo propietario en formato OPC, para que estos sean compatibles con una o varias especificaciones OPC.
- Comunicaciones servidor OPC – fuente de datos: Pueden ser realizadas de múltiples maneras debido a que cada controlador o aplicación utiliza un protocolo distinto.

#### 4.1.2 Estándar OPC en LabVIEW

En el mecanismo de comunicación propuesto se utiliza la herramienta software LabVIEW para establecer un envío y recepción de datos del proceso entre el PLC y el dispositivo móvil, en este sentido es importante tener en cuenta que la solución propuesta no es la más óptima debido a la utilización de esta herramienta software, esto debido a que su aplicabilidad en el entorno industrial es muy baja.

LabVIEW puede ser utilizado como un servidor OPC, debido a que se pueden comunicar cualquier tipo de datos con dispositivos que soporten este estándar, esto se realiza

mediante una variable compartida, que puede ser utilizada para comunicar datos con otras aplicaciones utilizando tanto OPC como otras funcionalidades del software LabVIEW [60].

Para crear un servidor/cliente OPC y realizar la conexión con cualquier dispositivo en el mecanismo de comunicación propuesto se utiliza la herramienta NI OPC Servers. En la Figura 4. 3 se ilustra la interfaz del NI OPC Servers, en la cual se realiza la conexión con el servidor OPC, aquí se especifican todas las configuraciones necesarias para realizar la conexión de LabVIEW con el PLC, se crea un canal, se agrega un dispositivo y un grupo de tags que se importarán después según sea necesario. El servidor NI OPC cuenta a su vez con una interfaz de un cliente rápido (Figura 4. 4) en donde se puede observar el estado de la conexión entre el servidor y el dispositivo enlazado (PLC).

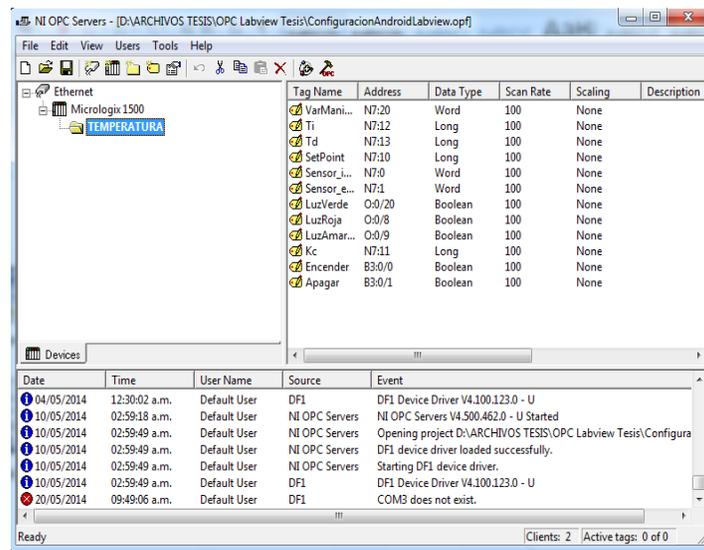


Figura 4. 3. Servidor NI OPC Servers  
[Fuente: Propia]

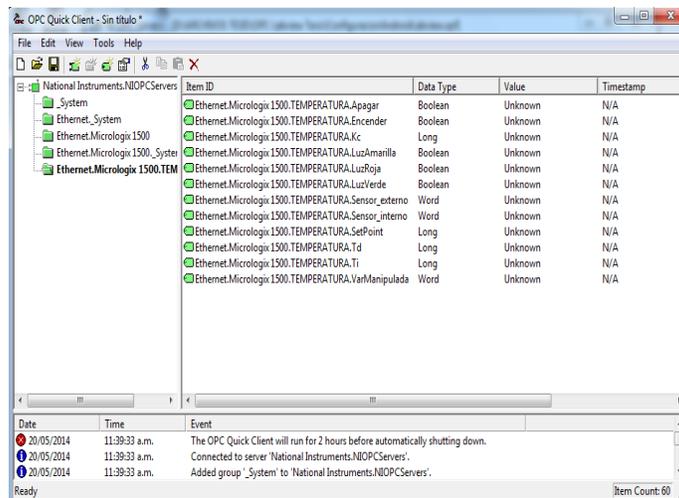


Figura 4. 4. Cliente Rápido de NI OPC Servers  
[Fuente: propia]

Luego de establecer la conexión correspondiente entre la fuente de datos (PLC o dispositivo de control) y el servidor OPC. Se crean todas las variables compartidas mencionadas anteriormente para poder ser manipuladas posteriormente mediante Instrumentos virtuales (.vi) desarrollados en el software LabVIEW. El objetivo en este punto es capturar las variables compartidas para que puedan ser enviadas hacia el dispositivo móvil mediante el protocolo TCP/IP. En la Figura 4. 5 se observa una tag importada (“Encender”) por medio de OPC la cual hace parte de una comunicación TCP desarrollada en LabVIEW.

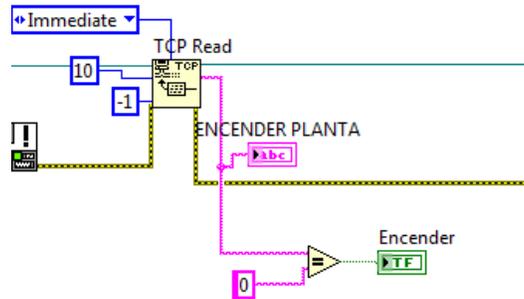


Figura 4. 5. Ejemplo de Uso de una Tag OPC en Código de LabVIEW  
[Fuente: Propia]

## 4.2 Comunicación TCP/IP Bidireccional entre LabVIEW y el dispositivo móvil.

En el desarrollo del presente proyecto se optó por utilizar el protocolo TCP/IP debido a que es muy común en la industria, cuenta con un generoso ancho de banda para su uso en HMIs Industriales y se integra de manera muy simple a las empresas. Otro aspecto importante es que es el mismo protocolo utilizado por la World Wide Web (www) [47]; Con el uso de este protocolo se tiene la opción de acceder a la Web, lo cual permite una mayor variedad de funciones y mejora de gran manera la conectividad de las aplicaciones que se implementen con el uso de la red de internet.

### 4.2.1 Arquitectura TCP/IP

Las siglas TCP/IP se refieren a un conjunto de protocolos para comunicaciones de datos. Este conjunto toma su nombre de dos de sus protocolos más importantes, el protocolo TCP (Transmission control protocol) y el protocolo IP (internet protocol) [48]. En la Figura 4. 6 se puede observar las capas de la arquitectura TCP/IP con su familia de protocolos.

**TCP:** es un protocolo de transporte orientado a conexión, esto hace que los mensajes se entreguen sin errores, sin omisión y en secuencia. TCP actualmente se encuentra ampliamente extendido en internet siendo el protocolo utilizado por las aplicaciones de red más populares, como Telnet, ftp y acceso a web [49].

El protocolo TCP tiene las siguientes características:

- Es orientado a conexión, debido a que las aplicaciones solicitan la conexión al destino y luego usan esta conexión para entregar los datos, y garantizan que estos datos lleguen de manera exitosa a su destino.
- Punto a punto, es decir, una conexión TCP tiene dos extremos, emisor y receptor.
- Confiabilidad. TCP garantiza que los datos transmitidos sean entregados sin pérdidas, duplicación o errores de transmisión.

- Full Dúplex. Los extremos que participan en una comunicación TCP pueden intercambiar datos en ambas direcciones de manera simultánea.
- Conexión de inicio confiable. TCP permite una conexión de inicio confiable y sincronizada entre los dos extremos de la conexión.
- Conexión de finalización aceptable. TCP garantiza la entrega de todos los datos antes de que se cierre la conexión [50].

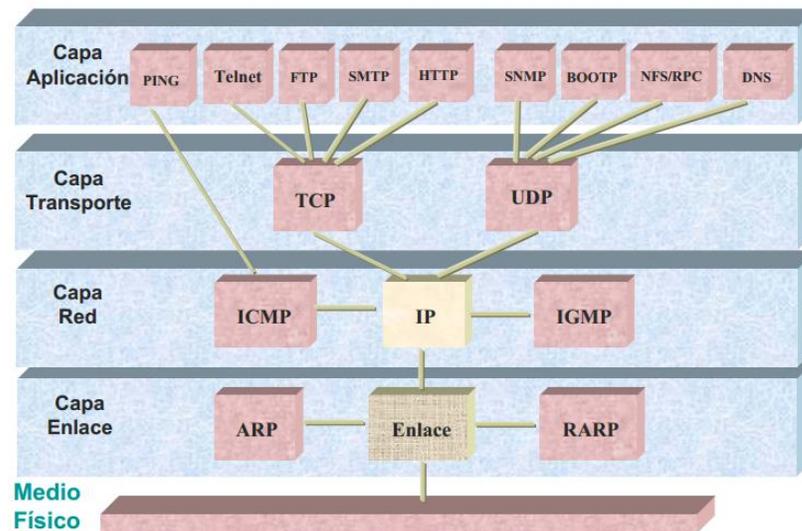


Figura 4. 6. Arquitectura TCP/IP y su familia de protocolos.  
 [Fuente: [http://isa.umh.es/asignaturas/sitr/TraspSITR\\_TCP-IP.pdf](http://isa.umh.es/asignaturas/sitr/TraspSITR_TCP-IP.pdf)]

**IP:** Protocolo perteneciente a la capa de red, por tanto, es utilizado por los protocolos del nivel de transporte como TCP para llevar los mensajes hacia su destino. El trabajo de IP es únicamente encaminar al datagrama, sin comprobar la información que contiene, para ello adiciona una cabecera que se ubica al datagrama que está usando.

Este protocolo identifica los ordenadores que se encuentran conectados a la red mediante las direcciones. Esta dirección dependiendo de la versión del protocolo IP pueden ser dos: la primera una dirección IP versión 4 que es un número de 32 bits el cual es único para cada host. La segunda es la dirección IP versión 6, la cual es un número de 128 bits también único para cada host, es importante mencionar que el protocolo IP versión 6 es un protocolo mejorado, el cual tiene más capacidad y se han corregido muchos errores de seguridad que tenía la versión 4, además ofrece soporte a nuevas redes de alto rendimiento (como ATM, Gigabit Ethernet, etc.).

#### 4.2.2 Protocolo TCP/IP y configuración en LabVIEW

LabVIEW cuenta con un módulo de comunicación que utiliza TCP/IP para comunicar tanto redes locales, como interconectadas (Internet).

Para lograr una conexión TCP/IP exitosa en LabVIEW se deben utilizar las funciones de: abrir conexión, leer, escribir y cerrar la conexión, en la Figura 4. 7 se ilustra todas las funciones de la herramienta de TCP/IP, la cual proporciona una interfaz de usuario simple:

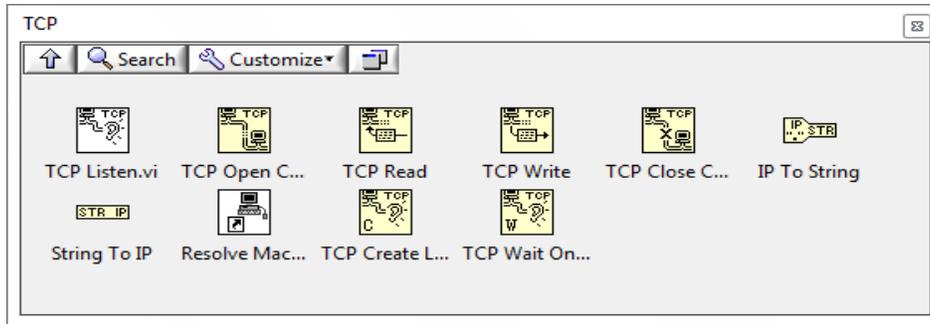


Figura 4. 7. Paleta de Funciones TCP/IP en LabVIEW  
[Fuente: propia]

Existen dos tipos de configuración para la comunicación TCP: modo servidor y modo cliente. El modo servidor espera conexiones entrantes mientras el modo cliente realiza una llamada a un número IP y al puerto del servidor, dependiendo del tipo de configuración que se quiera realizar se utiliza un bloque (También llamado vi) u otro. Figura 4. 7.

Con la configuración necesaria, un computador puede funcionar como cliente o como servidor. El diagrama de bloques de la Figura 4. 8 representa una aplicación cliente, la cual inicia una conexión a un servidor remoto utilizando la función “TCP Open Connection”, en donde el servidor “escucha” conexiones remotas y responde apropiadamente. En el establecimiento de conexiones TCP, se tiene que especificar la dirección y un puerto de esta. Los diferentes puertos en las direcciones dadas identifican diferentes servicios en estas direcciones.

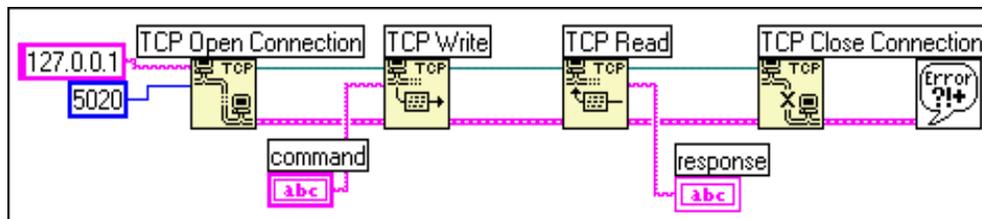


Figura 4. 8. Configuración de un Cliente en LabVIEW  
[Fuente: [53]]

Se debe utilizar la función TCP Open Connection para establecer una conexión con un puerto y una dirección específica. Si la conexión es correcta, la función retorna un número de conexión de red que identifica como única a esta conexión. Este número de conexión se utiliza para referir a la conexión en las llamadas, en los subsecuentes VI's o posteriores aplicaciones desarrolladas en LabVIEW.

De una manera detallada, para la creación de un cliente se deben seguir los siguientes pasos:

- Crear un TCP Server (Opcional).
- Usar la función “TCP Open Connection” para abrir una conexión al servidor. Se debe especificar el puerto al cual se va a conectar el cliente. Si se desea establecer una conexión a un computador remoto, además del puerto se debe especificar la

dirección IP del servidor, si esta no se especifica, LabVIEW establece una conexión al computador local.

- Usar la función “TCP Read” para leer un mensaje del servidor. Se debe especificar el número de bytes que se desea leer.
- Usar la función “TCP Write” para enviar un mensaje al servidor.
- Usar la función “TCP Close Connection” para cerrar la conexión al servidor. [54].

El diagrama de bloques de la Figura 4. 9 representa el esquema general de un servidor para el desarrollo del mecanismo propuesto. Éste utiliza el valor de la dirección remota de salida del “TCP Listen VI” para determinar si un cliente remoto tiene permiso para acceder al servidor:

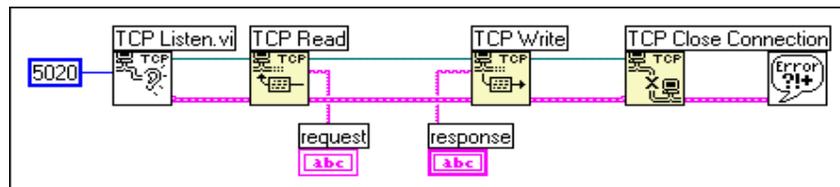


Figura 4. 9. Configuración de un Servidor en LabVIEW  
[Fuente: [53]]

Para la creación de un servidor en LabVIEW se deben realizar los siguientes pasos:

- Usar la función “TCP Listen VI” para esperar por una conexión, aquí se debe especificar el puerto, que debe ser el mismo puerto al cual el cliente intenta el acceso.
- Si la conexión es establecida, usar la función “TCP Write” para enviar un mensaje al cliente, los datos deberán estar de tal forma que el cliente los pueda aceptar.
- Usar la función “TCP Read” para leer un mensaje proveniente del cliente.
- Usar la función “TCP Close Connection” para cerrar la conexión [55].

Un aspecto que se debe tener en cuenta es que aunque LabVIEW provee una herramienta de comunicación utilizando el protocolo de comunicación UDP, el cual es un protocolo sin conexión con alto rendimiento, no es recomendable utilizarlo debido a que no asegura una transmisión de datos fiable. Por otra parte el protocolo TCP es orientado a conexión lo que hace que la transmisión de datos sea más segura [56].

En la Figura 4. 10 se ilustra el .vi específico que se desarrolló para realizar la comunicación bidireccional entre el dispositivo móvil y el servidor NI OPC, concretamente el envío de datos de las variables del proceso hacia el dispositivo. Es importante resaltar que este .vi hace las veces de servidor para que los datos puedan ser manipulados por el dispositivo móvil, que es el cliente.

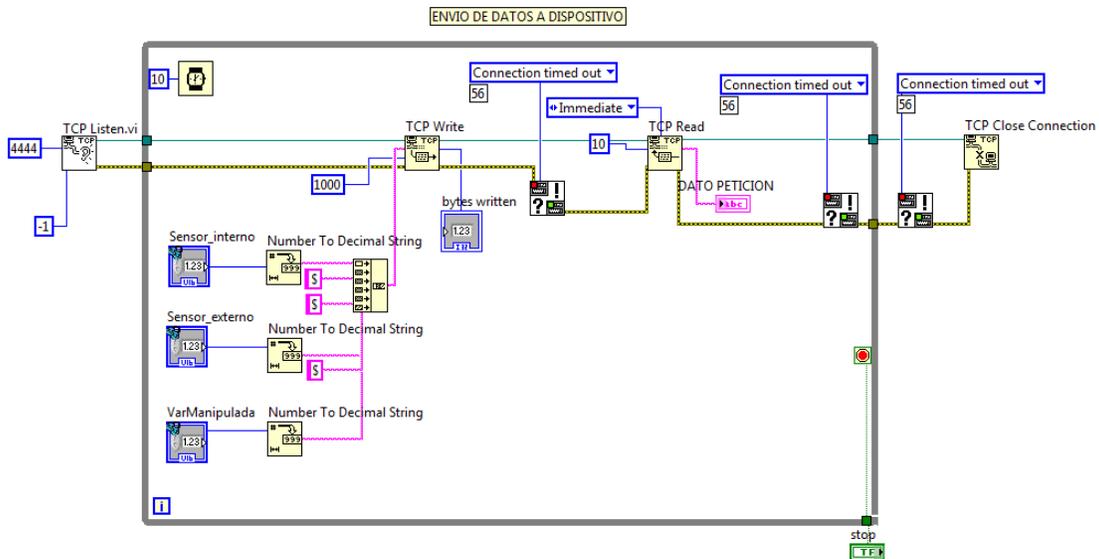


Figura 4. 10. .vi desarrollado para el envío y recepción de datos hacia y desde el dispositivo móvil.  
[Fuente: propia]

### 4.2.3 Protocolo TCP/IP en JAVA

Una vez que los datos se pueden enviar vía TCP/IP mediante el servidor creado en LabVIEW, es necesario que estos puedan ser “escuchados” y “manipulados” en el dispositivo móvil. En el presente proyecto se utilizó el lenguaje JAVA para llevar a cabo el desarrollo de la aplicación de supervisión de la planta. Aunque en la actualidad son muchas las plataformas y lenguajes en los que se puede llevar a cabo los desarrollos de aplicaciones móviles para el sistema operativo Android. JAVA es el lenguaje predilecto para desarrollar aplicaciones nativas.

La plataforma más utilizada y más completa para el desarrollo de aplicaciones móviles en Android es Eclipse. La programación en este entorno de desarrollo está basada en el lenguaje de programación JAVA, por lo que se debe tener claro el concepto de socket, debido a que es por medio de este que se pueden realizar conexiones usando el protocolo TCP/IP.

Un Socket (“Enchufe” por su traducción) es una abstracción software que se utiliza para representar los “terminales” de una conexión entre dos máquinas. Para una conexión dada se puede imaginar un “cable” hipotético en donde los dos extremos en los cuales se enchufan las máquinas son los sockets [57].

Para realizar una conexión TCP/IP en JAVA lo primero que se debe hacer es identificar la máquina y conectarse a ella, esto se logra por medio de la dirección IP a la que se quiere comunicar, esta dirección existe de dos maneras:

- La forma familiar DNS, en donde se tiene un dominio como “tesisandroid.com” y un computador de nombre “pc” en nuestro dominio, por lo tanto el nombre de dominio de “pc” es “pc.tesisandroid.com”, esta es una dirección que se utiliza y se incorpora en le world wide web (WWW)
- La forma del “cuarteto punteado” que consta de cuatro números separados por puntos, como 124.255.112.150.

En los dos casos la dirección IP se representa internamente como un número de 32 bits (ninguno de los cuatro números puede exceder de 255). Para representar esta dirección en un objeto JAVA se utiliza el método de la librería java.net:

### static InetAddress.getByName()

El resultado es un objeto de tipo **InetAddress**, que se usa para construir el socket y obtener la dirección IP actual o a la que se desea conectar. La dirección IP se ingresa en el método, ya sea escribiendo la dirección web, ingresando la forma del cuarteto punteado de la IP o si se va a utilizar el localhost de la máquina ingresando la sentencia "null", "localhost" o su forma en cuarteto punteado 127.0.0.1.

Además de la dirección IP también se necesita un puerto, puesto que puede haber muchos servidores en una misma máquina. Cada máquina IP contiene puertos y se debe establecer en que puerto se van a conectar tanto el cliente como el servidor. El puerto no es una ubicación física sino una abstracción software, un programa cliente sabe cómo conectarse a una máquina por medio de la dirección IP, pero no sabe a qué servicio de la máquina conectarse, es por esto que se utilizan los puertos como un segundo nivel de direccionamiento.

Luego de identificar la máquina se crea el socket, para esto existen dos clases de sockets dependiendo de la configuración que se quiera realizar: **ServerSocket** para crear un servidor y **Socket** para crear un cliente. Luego se crea un **InputStream** y un **OutputStream** (un lector y un escritor) a partir del socket creado para poder tratar la conexión como un objeto de flujo E/S (entrada y salida), estos objetos se obtienen con los métodos **getInputStream()** y **getOutputStream()**, y luego se envuelven en un **BufferedReader** y un **PrintWriter** respectivamente. El **BufferedReader** realiza la conversión a caracteres de los bytes que ingresan por el objeto **InputStream**, el objeto **OutputStream** se envuelve en un **PrintWriter** en donde se envían las líneas de texto, finalmente cuando toda la comunicación se realiza se debe cerrar el socket [57].

En la Figura 4. 11 se ilustra un resumen de lo expuesto anteriormente.

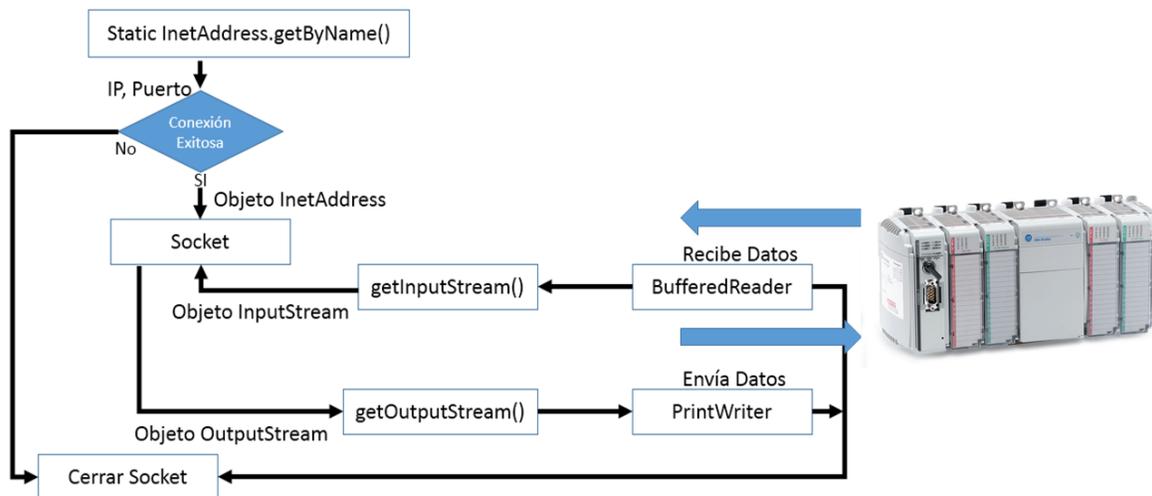


Figura 4. 11. Diagrama de Flujo de Datos Usando Sockets en Java [Fuente: propia]

### 4.2.1 Mecanismo de comunicación final.

Teniendo en cuenta todos los aspectos mencionados anteriormente se ha propuesto un esquema de comunicación bidireccional entre el PLC (fuente de datos) y el dispositivo móvil el cual tiene albergada una HMI industrial.



*Figura 4. 12. Mecanismo de Comunicación con el protocolo TCP/IP  
[Fuente: Propia]*

Se tuvieron en cuenta tres aspectos importantes para establecer esta comunicación.

El primero es la utilización del estándar OPC para lograr la interoperabilidad entre los diferentes software utilizados para lograr la supervisión del proceso en el dispositivo móvil.

El segundo es el uso del protocolo TCP/IP para el envío y recepción de los datos del proceso desde y hacia la HMI desarrollada y albergada en el dispositivo móvil.

Por último la utilización de la herramienta software LabVIEW que sirve como puente entre el servidor OPC a utilizar y el dispositivo móvil (ver Figura 4.12). Se aclara nuevamente que esta herramienta no es la más adecuada para establecer la comunicación entre el PLC y el dispositivo móvil, su uso en el desarrollo del proyecto es académico. Para un desarrollo a nivel industrial se deben explorar herramientas como el mismo software del fabricante que cuenta con librerías y herramientas OPC para establecer una determinada comunicación, o el software libre Python que cuenta con librerías OPC y componentes para establecer una comunicación bidireccional a través de TCP/IP.

En la Figura 4. 13 se observa con más detalle el mecanismo utilizado para realizar la conexión entre el PLC y el dispositivo Android.

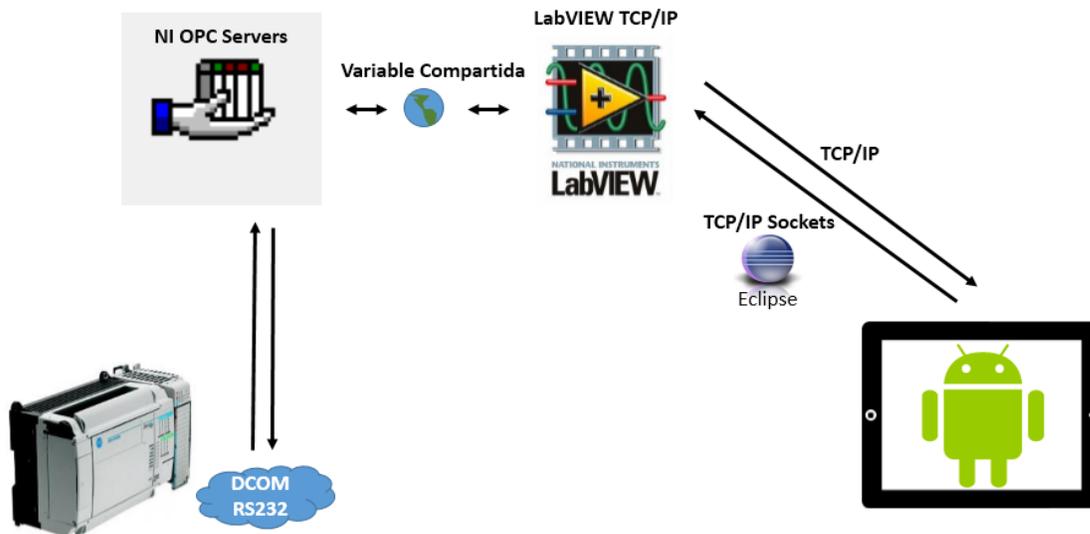


Figura 4. 13. Mecanismo de Comunicación con TCP/IP Programas  
[Fuente: Propia]

En primera instancia entre el PC y el PLC se realiza una comunicación serial utilizando un cable RS232.

La herramienta que LabVIEW utiliza para comunicarse con el PLC es NI OPC Servers basada en la especificación OPC DA, en este punto se crea una comunicación OPC con el PLC y se puede enviar y recibir datos por medio de variables compartidas que son las TAGS importadas del PLC. Estas variables, pueden adicionarse al código (VI) implementado en LabVIEW y se pueden complementar y usarse en otras herramientas de este software.

Para comunicar LabVIEW con el dispositivo Android de una manera bidireccional se utilizó el protocolo de comunicación TCP/IP, para esto, LabVIEW cuenta con la herramienta TCP/IP que permite este tipo de comunicación. Por parte del dispositivo móvil, la comunicación TCP/IP entre LabVIEW y este se hace por medio de Sockets que se implementan en el desarrollo de la aplicación, para esto se utiliza el Framework Eclipse que esta soportado sobre el lenguaje JAVA.

Con la comunicación establecida ya se puede realizar un intercambio de datos entre el dispositivo Android y el PLC, para llevar a cabo la supervisión de cualquier proceso industrial a través de una HMI desarrollada en una aplicación Android.

## 5. Conclusiones.

En el presente trabajo se propuso una guía para el diseño de interfaces gráficas industriales bajo el sistema operativo Android, teniendo como base fundamental el modelo de proceso de la ingeniería de la usabilidad y la accesibilidad (MPIu+a), que se enmarca dentro de la filosofía del diseño centrado en el usuario (DCU). Además se tuvieron en cuenta directrices y recomendaciones realizadas por varios autores referentes al diseño de HMIs industriales.

Finalmente se elaboró un prototipo que cumpliera con las recomendaciones propuestas en la guía, el cual fue aplicado a un proceso en el laboratorio de control de procesos de la Universidad del Cauca, en este contexto las principales conclusiones que se obtuvieron con el trabajo desarrollado fueron:

- El modelo mental que tiene el operador acerca de cómo funciona el proceso que se está llevando a cabo es fundamental para la supervisión de éste. Es por esta razón que para diseñar la HMI se debe trabajar estrechamente con el operador para entender sus modelos mentales y responsabilidades. Si no se realiza de esta manera, se podrá malinterpretar la información presentada, el usuario no sabrá el significado de los cambios que se realicen en el sistema, lo cual puede acarrear problemas como paros indeseados, fallas de producción o accidentes.
- La usabilidad es un aspecto muy importante en el desarrollo de cualquier sistema interactivo, debido a que es un factor de calidad para las interfaces de usuario de una gran variedad de dispositivos, aunque para el diseño de aplicaciones que logren efectivamente los propósitos de facilidad de uso, ésta no es una tarea sencilla debido a la existencia de varios métodos y técnicas para la evaluación de la usabilidad, las cuales no han sido integradas en un simple marco conceptual que facilite su uso por los desarrolladores.
- A través del tiempo se han desarrollado tecnologías que han generado grandes cambios (generalmente positivos) en el funcionamiento de las industrias. En la actualidad han aparecido como una nueva tecnología los dispositivos móviles inteligentes (Smartphones y tablets), los cuales también se han intentado incluir en el ambiente industrial, para llevar a cabo tareas de supervisión de procesos. La inclusión de esta tecnología unida a la filosofía del diseño centrado en el usuario, permite que se abran nuevas líneas de investigación encaminadas hacia el mejoramiento de los procesos productivos en las empresas.
- En el presente trabajo se realizó una comunicación utilizando la herramienta software LabVIEW como puente entre la aplicación Android y el proceso gobernado por el PLC de marca Allen Bradley. Aunque esta es una herramienta versátil, a un nivel empresarial aumentaría los costos para su implementación, por tal razón es conveniente que se exploren otras herramientas tanto software como hardware para lograr esta comunicación; como por ejemplo el mismo software del fabricante del PLC que cuenta con una serie de herramientas para lograr la obtención de las tags del proceso por medio de un servidor OPC y comunicarlas al dispositivo móvil

mediante servidores TCP, HTTP, entre otros. Otra solución es la del software libre Python que cuenta con una serie de librerías OPC para establecer una comunicación con el PLC y enviar los datos al dispositivo móvil, o por último el módulo de conexión Xbee y el software que permite su configuración.

- El modelo MPlu+a tiene una amplia flexibilidad y versatilidad porque permite el uso de diversos procedimientos como por ejemplo el libre uso de técnicas de prototipado y de evaluación. Además en la fase de implementación, se permite el uso libre de cualquier software para el desarrollo del prototipo final que se esté desarrollando. Esto hace que esta sea una metodología que se puede aplicar a cualquier desarrollo software (como por ejemplo una HMI industrial) sin importar su tamaño ni contexto (móvil, web, etc.).
- El uso de dispositivos móviles en la supervisión de procesos, permite mejorar la eficiencia del proceso productivo debido a la portabilidad que ofrece el dispositivo lo cual disminuye el esfuerzo que se debe realizar en un proceso de supervisión. Además de la integración de múltiples funciones que poseen los dispositivos inteligentes (como dar aviso de alarmas usando servicios u otros procesos en background, comunicar con otras aplicaciones para generar estadísticas, etc.), lo cual ayuda a dar una mayor funcionalidad a la supervisión.
- Usar únicamente el concepto de DCU (diseño centrado en el usuario) sin implementar el modelo MPlu+a, dificulta el desarrollo de las interfaces, esto debido a que al no tener la secuencia metódica que establece el modelo para la relación que se debe tener con el usuario, no van a tomarse en cuenta muchos aspectos importantes que solo son de conocimiento y preferencia del usuario, lo que conlleva a que se generen constantes y desordenadas realimentaciones para el desarrollo de la interfaz industrial que representa el proceso que se está llevando a cabo, lo cual acarrea una pérdida importante de tiempo, además que de que el usuario no siempre sabe que es lo que quiere y generará más requisitos.
- Para realizar la comunicación entre el dispositivo Android y el software que permite la comunicación con el PLC, es recomendable utilizar una red privada o una red de área local dentro de la empresa, esto debido a que los datos que maneja una HMI para el proceso de supervisión son datos sensibles a los cuales solo personal autorizado debe tener acceso. El realizar esta conexión por medio de una red pública con acceso a internet puede representar un peligro para el proceso debido a que personal externo puede tener acceso a los datos que comunica la interfaz con el software. Sin embargo si se desea utilizar una red pública se debe tener un nivel de seguridad avanzado para el tratamiento de los datos.
- Gracias a la inclusión de los dispositivos móviles en el diario vivir de las personas, es que se han llevado a cabo importantes estudios enfocados hacia la evaluación de la usabilidad y funcionalidad de los sistemas interactivos que se albergan dentro de estos dispositivos inteligentes. Estos estudios abren caminos cada vez más

grandes que conllevan a que las aplicaciones móviles sean desarrolladas de una manera más eficiente.

## 6. Trabajos Futuros.

Realizar un esquema de seguridad que proteja los datos que se envían entre el dispositivo Android y el servidor de LabVIEW.

Vincular los parámetros de IOS en la guía propuesta para desarrollar HMIs industriales en este sistema operativo.

Crear plantillas interactivas que permitan a sus usuarios diseñar HMIs que cumplan con los parámetros de la guía propuesta.

Realizar un mecanismo que permita una comunicación directa entre el dispositivo móvil y el PLC del proceso, en el cual se explore tanto como nuevo software como hardware que facilite la construcción de este mecanismo de comunicación.

## 7. Contribuciones.

Se elaboró el Artículo “Supervisión de un proceso de temperatura gobernado por un PLC, mediante una HMI desarrollada en Android”, el cual fue aceptado para ser presentado en el III congreso internacional de Ingeniería mecatrónica y automatización realizado en la universidad tecnológica de Bolívar en la Ciudad de Cartagena de Indias los días 22, 23 y 24 de Octubre del año 2014.



El trabajo fue tenido en cuenta para ser expuesto en la página del grupo de investigación GRIHO, en la sección de trabajos relacionados.



La guía de diseño propuesta sirvió como base teórica para la elaboración de la metodología: Metodología para el prototipado ágil de aplicaciones móviles.



## 8. Bibliografía

- [1] G. Johannsen, «Human Machine Interaction,» de *Control, Systems, Robotics and Automation*, Kassel, Germany, Heinz Unbehauen, 2009.
- [2] J. Colomer, M. Joaquim y A. Jordi, «Introducción a la monitorización y supervisión experta de procesos. Metodos y herramientas.,» CEA - IFAC, Girona, 2004.
- [3] L. Hurtado, «Desarrollo de una metodología de evaluación de usabilidad de interfaces Humano - Maquina para la mejora del proceso de toma de decisiones en tareas de supervisión industrial,» Universidad Nacional de Colombia, sede Manizales, Manizales, 2013.
- [4] S. Valencia Aguilar, «Guía Metodológica para diseño de interfaces de usuario par control y supervisión de procesos industriales.,» Medellín, 2012.
- [5] J. Garrido, «TFC Desarrollo de aplicaciones Mviles,» Universidad abierta de Catalunya, Catalunya, 2013.
- [6] GRIHO, «Modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad,» 2010. [En línea]. Available: <http://www.grihocitools.udl.cat/mpiu/usabilidad/usabilidad.html>. [Último acceso: 15 Octubre 2013].
- [7] J. Sanchez, «No solo usabilidad: revista sobre personas, diseño y tecnología,» 5 Septiembre 2011. [En línea]. Available: <http://www.nosolousabilidad.com/articulos/dcu.htm>. [Último acceso: 25 Octubre 2013].
- [8] O. Sotelo y C. Palacios, «Integración de Actividades de aprendizaje contextual en la plataforma .LRN a través de dispositivos móviles android con soporte NFC,» Universidad del Cauca, Popayan, 2011.
- [9] M. Baez, A. Borrego y J. Cordero, «Introducción a Android,» E.M.E Editorial., Madrid.
- [10] D. Angel, «Aplicación Móvil en Android para la gestión de entrenos de deportistas,» Universidad Politécnica de Valencia, Valencia, Septiembre de 2013.

- [11] Y. Molina, J. Sandoval y S. Toledo, «Sistema Operativo Android: Características y funcionalidad para dispositivos móviles,» Universidad tecnológica de Pereira, Pereira, 2012.
- [12] M. Fuentes y G. Johan, «Prácticas de gestión de energía en aplicaciones para dispositivos Android, soportadas en periféricos específicos,» Universidad del Cauca, Popayán, 2013.
- [13] Android, «Android Developers,» 2013. [En línea]. Available: <http://developer.android.com/index.html>. [Último acceso: 20 Enero 2014].
- [14] M. Araceli Moré, «MPIu+a Ágil: El modelo de proceso centrado en el usuario como metodología ágil,» Lerida, 2010.
- [15] T. Granollers Saltiveri, J. Lóres Vidal y J. Cañas Delgado, Diseño de sistemas interactivos centrados en el usuario, Barcelona: UOC, 2005.
- [16] M. Gea y F. Gutiérrez, «El diseño,» Granada, 2002.
- [17] E. Ruíz Lizama, J. L. Inche Mitma y A. R. Chung Pinzas, «Desarrollo de una interfaz hombre máquina orientada al control de procesos,» *Industrial Data*, vol. Vol. 11, pp. pp. 70 - 72, 2008.
- [18] A. Rodríguez Penin, Sistemas Scada, 2da ed., Barcelona: Marcombo S.A., 2007.
- [19] M. D. Network, «Microsoft Developer Network,» 2011. [En línea]. Available: [http://msdn.microsoft.com/es-es/library/aa969773\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/aa969773(v=vs.110).aspx). [Último acceso: 10 12 2013].
- [20] S. Gomez, «Pensamientos varios sobre programación, Android, .Net y Java,» 09 06 2011. [En línea]. Available: <http://www.sgoliver.net/blog/?p=1991>. [Último acceso: 11 12 2013].
- [21] E. d. e. i. y. empresariales, «Diseño de plantas, señalización y demarcación de áreas,» Bucaramanga.
- [22] P. Ponsa y A. Granollers, «Diseño de Pantalla,» Cataluña, 2007.
- [23] G. Ordoñez, «Introducción a los sistemas Scada,» 30 Octubre 2009. [En línea]. Available: [http://www.oocities.org/gabrielordonez\\_ve/MTU.htm#\\_top](http://www.oocities.org/gabrielordonez_ve/MTU.htm#_top). [Último acceso: 10 Diciembre 2013].
- [24] Inductive Automation, «Smarter SCADA alarming,» 2013.

- [25] W. Lidwell, H. Kritina y B. Jill, Universal principles of design, Beberly, Massachusetts: Rockport publishers, 2012.
- [26] J. Alberto, «Introduccion a PHP,» 5 Octubre 2012. [En línea]. Available: <http://introduccionphp.blogspot.com/2012/10/patrones-de-diseno.html>. [Último acceso: 25 Enero 2014].
- [27] P. Laja, How to Build Websites that Sell: The Scientific Approach to Websites, Hyperink Blog to Books, 2013.
- [28] J. Tomas, «APIs para graficos en Android,» 2013. [En línea]. Available: <https://polimedia.upv.es/visor/?id=68cd39db-0b86-f04a-96a4-f76d643c0dcb#>. [Último acceso: 20 Febrero 2014].
- [29] G. Martinez Muñoz, «Graficos dos dimensiones,» Madrid, 2012.
- [30] J. E. Amaro Soriano, El gran libro de programación avanzada con Android, Mexico D.F.: Alfaomega, 2013.
- [31] W. Maner, «SIDAR,» Zaragoza, 2000.
- [32] M. Davies, «Paper Prototyping as a core tool in the design of mobile phone user experiences,» John Wiley & sons, Ltd, UK, 2007.
- [33] A. Dix y J. Finlay, Human-Computer interaction, Harlow: Pearson, 2004.
- [34] J. G. Enriquez y S. I. Casas, «Usabilidad en aplicaciones moviles,» Rio Gallegos, 2013.
- [35] C. Cuadrat Seix, «Estudio sobre Evaluación de la Usabilidad Móvil y Propuesta de un Método para Tests de Usabilidad Cuantitativos basado en Técnicas de Eyetracking,» Lérída, 2012.
- [36] R. Inostroza, C. Rusu, S. Roncagliolo, C. Jiménez y V. Rusu, «Usability Heuristics for Touchscreen-based Mobile Devices,» de *2012 Ninth International Conference on Information Technology - New Generations*, Las Vegas, 2012.
- [37] E. Bertini, S. Gabrielli y S. Kimani, «Appropriating and Assessing Heuristics for Mobile Computing,» Venezia, 2006.
- [38] Nielsen Norman Group, «Nielsen Norman Group - Evidence-Based User Experience Research, Training, and Consulting,» 2014. [En línea]. Available: <http://www.nngroup.com/articles/author/jakob-nielsen/>. [Último acceso: 15 Abril 2014].
- [39] ISO Standars No. 9241-11, «Guidance on usability:,» Geneva, 1998.

- [40] J. Nielsen y R. Mack, *Usability inspection methods*, New York: John Wiley & Sons Inc, 1994.
- [41] M. Hernandez, «Usabilidad y UX en Colombia,» 18 Junio 2013. [En línea]. Available: <http://www.uxabilidad.com/usabilidad/10-heuristicas-o-principios-basicos-de-usabilidad.html>. [Último acceso: 18 Marzo 2014].
- [42] S. Po, S. Howard, F. Vetere y S. Mikael, «Heuristic Evaluation and Mobile Usability: Bridging the Realism Gap,» *Mark Dunlop's publications*, pp. 49 - 60, 2004.
- [43] Y. G. Ji, J. H. Park, C. Lee y M. H. Yun, «A Usability Checklist for the Usability Evaluation of Mobile Phone User Interface,» *International Journal of human - computer Interaction*, vol. 20, nº 3, pp. 207 - 231, 2006.
- [44] Departamento de Instrumentación y control Unicauca, «Planta de Temperatura: Reconocimiento,» Universidad del Cauca, Popayan.
- [45] M. Alva, «Metodología de medición y evaluación de la usabilidad en sitios web educativos,» Universidad de Oviedo, Oviedo, 2005.
- [46] C. Lagos, «One Touch Electro Industria,» Septiembre 2006. [En línea]. Available: <http://www.emb.cl/electroindustria/articulo.mvc?xid=562&edi=7&xit=protocolos-de-comunicacion-industrial>. [Último acceso: 3 Diciembre 2013].
- [47] Rockwell Automation, «Rockwell Automation,» [En línea]. Available: <http://ab.rockwellautomation.com/es/networks-and-communications/ethernet-ip-network#/tab2>. [Último acceso: 5 Diciembre 2013 ].
- [48] F. Sidnie, «Arquitectura , protocolos e implementación,» Mc-Graw Hill, España, 1998.
- [49] Universidad de Valencia, «Protocolo TCP/IP,» Universidad de Valencia, Valencia, 2000.
- [50] L. Garcia, «Protocolo de control de transmisión (TCP).,» *Redes y comunicaciones de datos*, 2007.
- [51] W. Stallings, *Comunicaciones y Redes de Computadores*, Madrid: Pearson Educación S.A, 2004.
- [52] C. F. Jimenez, «Conexión TCP/IP Entre Dos Estaciones Usando Labview 7 Express,» Febrero 2005.
- [53] National Instruments, «Basic TCP/IP Communication in LabVIEW,» 6 Septiembre 2006. [En línea]. Available: <http://www.ni.com/white-paper/2710/en/>. [Último acceso: 5 Diciembre 2013].

- [54] National Instruments, «Creating a TCP Client,» Agosto 2006. [En línea]. Available: [http://zone.ni.com/reference/en-XX/help/371361B-01/lvhowto/creating\\_tcp\\_client/](http://zone.ni.com/reference/en-XX/help/371361B-01/lvhowto/creating_tcp_client/). [Último acceso: 5 Diciembre 2013].
- [55] National Instruments, «Creating a TCP Server,» Agosto 2006. [En línea]. Available: [http://zone.ni.com/reference/en-XX/help/371361B-01/lvhowto/creating\\_tcp\\_server/](http://zone.ni.com/reference/en-XX/help/371361B-01/lvhowto/creating_tcp_server/). [Último acceso: 8 Diciembre 2013].
- [56] National Instruments, «TCP,» Agosto 2006. [En línea]. Available: <http://zone.ni.com/reference/en-XX/help/371361B-01/lvconcepts/tcp/>. [Último acceso: 5 Diciembre 2013].
- [57] B. Eckel, *Pensando en Java*, Madrid: Pearson Education S.A, 2002.
- [58] OPC Foundation, «What is OPC?,» 2011. [En línea]. Available: [http://opcf.org/Default.aspx/01\\_about/01\\_what\\_is\\_opc.aspx?MID=AboutOPC](http://opcf.org/Default.aspx/01_about/01_what_is_opc.aspx?MID=AboutOPC). [Último acceso: 6 Diciembre 2013].
- [59] A. Diaz, «Arquitectura Para la Integración de Plataformas de Control Basadas en PC Mediante OPC,» Popayán, 2011.
- [60] C. Acevedo y R. Rueda, «Implementación de Labview como Sistema SCADA para la Arquitectura de Control SNAC PAC Opto 22, Mediante una Aplicacion OPC,» Bucaramanga, 2010.
- [61] National Instruments, «Conecte LabVIEW a Cualquier PLC,» 11 Diciembre 2013. [En línea]. Available: <http://www.ni.com/white-paper/7906/es/>. [Último acceso: 29 Diciembre 2013].
- [62] M. Rodríguez Penabad, 2006. [En línea]. Available: <http://coba.dc.fi.udc.es/~penabad/delphi6.pdf>. [Último acceso: 03 12 2013].
- [63] G. Sites, «Metodología gestión de requerimientos,» 2012. [En línea]. [Último acceso: 20 Noviembre 2013].
- [64] G. Minaya Lozano, «El análisis de tareas como herramientas de ayuda en la gestión de la seguridad y de la salud,» *MAPFRE seguridad*, nº Nro 71, 1998.
- [65] P. Molina Moreno, «Especificación de interfaz de usuario: de los requisitos a la generación automática,» Valencia, 2003.
- [66] I. Sommerville, *Software Engineering*, 9th ed. ed., Boston: Pearson, 2011.

- [67] J. Garcia Escobedo, «Apuntes de informatica, trucos y algo mas,» 4 Noviembre 2012. [En línea]. Available: <http://javiergarbedo.es/index.php/desarrollo-android/80-primeros-pasos/283-dimensiones>. [Último acceso: 2 Marzo 2014].
- [68] M. Rosson Beth y J. Carroll, «Usability Engineering: Scenario-Based Development of Human-Computer Interaction (Interactive Technologies),» Morgan Kaufmann, San Francisco, 2002.
- [69] stackexchange, «stackoverflow,» Mayo 2011. [En línea]. Available: <http://stackoverflow.com/questions/5960615/help-in-android-edit-text>. [Último acceso: 28 Agosto 2013].
- [70] Human Factor Research Group, «MUMMS: Measuring the Usability of Multi-Media Software,» University College Cork, Ireland, 2004. [En línea]. Available: <http://www.ucc.ie/hfrg/questionnaires/mumms/index.html>.