Universidad de las Ciencias Informáticas

HMI para el sistema Arex sobre dispositivos móviles con sistema operativo Android

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Yoenia Lapido Ramirez

Tutor: Dr.C Antonio Cedeño Pozo

La Habana, 2015

AGRADECIMIENTOS

En la realización de este trabajo jugó papel fundamental mi tutor y pareja, Dr.C Antonio Cedeño Pozo. Gracias por ser mi guía en todo momento, por tus consejos, apoyo y cariño. A todas las amistades que descubrí en el proceso de ser ingeniera, principalmente a mi grupo, 5502, y especial agradecimiento a Rosa y Ale por su apoyo incondicional.

DEDICATORIA

Dedico este trabajo a las personas más importantes de mi vida, mi familia. A mis padres Yoania y Agustín, a mis abuelos María Elena y Reynaldo, a mis padrinos Ingrid y Jack, a mis hermanas Alejandra y Yesenia, a mis tíos Rey Maikel y Rey Roger y a mi novio Tony.

SÍNTESIS

Arex es un sistema de medición de procesos industriales, enfocado a la domótica, que se desarrolla en la Línea de Sistemas Embebidos del Centro de Informática Industrial. Actualmente los módulos HMI-Ejecución y Recolector de Arex se comunican mediante D-bus, lo que implica que deban ser ejecutados sobre un mismo nodo físico. Debido a que el Recolector utiliza interfaces seriales para comunicarse con las tarjetas de adquisición, no puede ser embebido en dispositivos móviles inteligentes. En este trabajo se describen las adaptaciones que se realizaron para permitir que el HMI-Ejecución pueda ser ejecutado en un nodo independiente y de esta manera se pueda embeber en dispositivos móviles inteligentes, específicamente en dispositivos con sistema operativo Android. Para ello se desarrolló un conjunto de componentes gráficos que permiten el diseño de despliegues en aplicaciones domóticas, por ejemplo estado de las luminarias, presencia, temperatura ambiente, humedad relativa, etc. Además se modificó el mecanismo de comunicación entre el HMI-Ejecución y el Recolector, en este caso se utilizó el protocolo TCP sobre comunicación Wifi. Para el desarrollo de los componentes gráficos se hizo uso de QML con el Entorno de Desarrollo Integrado QtCreator, mientras que para el resto del desarrollo se utilizó el lenguaje C++ y el framework de desarrollo Qt versión 5.3.0. Como metodología de desarrollo de software se utilizó AUP en su versión UCI. El desarrollo de los nuevos componentes gráficos y las modificaciones realizadas al HMI-Ejecución permitieron que Arex pueda ser utilizado en el área de la domótica haciendo uso de dispositivos móviles inteligentes.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La domótica es el uso simultáneo de la electricidad, la electrónica y la informática, aplicadas a la gestión técnica de locales. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado [1]. Esta gestión técnica consiste en la modificación, local o remota, de los parámetros de funciones como:

- Gestión energética: regulación de temperaturas, gestión de los consumos de electrodomésticos y de la potencia contratada, etc.
- Comunicaciones: telecontrol y telemetría, correo electrónico, etc.
- Confort: programaciones horarias, escenarios luminosos, riego automático, etc.
- Seguridad: sistemas de vigilancia basados en cámaras, basados en sensores, etc.

Para ello, la domótica usa multitud de dispositivos que pueden ser distribuidos por todo el local en función de las necesidades de los clientes. Básicamente estos dispositivos se pueden dividir en sensores, actuadores y controladores.

La Universidad de las Ciencias Informáticas ha desarrollado algunos proyectos relacionados con el tema, entre ellos se encuentra Arex, un sistema de medición de datos desarrollado en la Línea de Sistemas Embebidos del centro CEDIN perteneciente a la Facultad 5. Arex es un sistema para medir procesos de baja y mediana complejidad a partir de la adquisición y procesamiento de datos en tiempo real, que puede ser ejecutado en hardware con bajas prestaciones. Está diseñado para aplicarse en áreas como la hotelería, la industria azucarera, en estaciones meteorológicas, en el sector corporativo, el particular, entre otros.

Con el auge de la telefonía móvil y la aparición de teléfonos inteligentes se ha hecho común el desarrollo de aplicaciones para realizar control automatizado de procesos sobre estas tecnologías. Para lograr que Arex aumente la competitividad

es necesario que los despliegues gráficos puedan ser visualizados sobre teléfonos inteligentes o similares. A los módulos o componentes que realizan las actividades de visualización, y sobre los que los usuarios y operadores interactúan directamente se les conoce como interfaz hombre máquina en tiempo de ejecución, que en lo adelante se denominará HMI-Ejecución.

Aunque Arex es un sistema de propósito general para la automatización de diversos procesos, este trabajo centra su interés en el campo de la domótica, que es el área de aplicación donde en los últimos años se han introducido con mayor fuerza las aplicaciones sobre dispositivos móviles inteligentes. En la actualidad existe un importante número de aplicaciones en el área de la domótica cuyos HMI-Ejecución se implementan sobre dispositivos móviles inteligentes, en este sentido han ganado especial espacio las que se encuentran desarrolladas para el sistema operativo Android, aunque de igual forma para IOS, Windows Mobile y otros se pueden encontrar aplicaciones de este tipo, por ejemplo: Loxone Smart Home [2], Planner Mobile [3], WeMo Light Switch [4], ElkDroid Security and Automation [5], DroidSeer X10 Home Automation [6], etc.

La mayoría de estos sistemas son de carácter propietario, generalmente implementados para la comunicación con redes de sensores o elementos de hardware, que aunque ofrecen excelentes prestaciones, son muy costosos. Como una limitación generalizada se puede mencionar que en su mayoría están diseñados a la medida para determinados procesos, pues dentro de la domótica hay una importante diversidad de ambientes y por tanto de soluciones. Este último aspecto queda resuelto con el sistema Arex, que viene siendo un configurador de soluciones que aplicado al área de la domótica, puede adaptarse y dar solución a una amplia diversidad de requerimientos.

El gobierno cubano desde hace algunos años ha dictado como una política estratégica fomentar el desarrollo de soluciones nacionales, que por una parte contribuyan a la sustitución de importaciones y al desarrollo endógeno, y por otro lado garanticen la seguridad ante posibles intentos de sabotaje a determinados sectores. En los temas

de seguridad los sistemas que forman parte de la automatización de procesos, en los que se encuentra Arex, juegan un papel importante.

Como se ha mencionado, Arex es un sistema genérico que puede ser empleado en diferentes áreas. Sin embargo la versión actual de este sistema no cuenta con componentes gráficos especializados en la visualización de procesos del área de la domótica, que constituye el centro de interés de este trabajo.

Ante la problemática enunciada se arriba al siguiente **problema de investigación**: ¿Cómo lograr que el sistema Arex pueda ser empleado en el área de la domótica, haciendo uso de dispositivos móviles inteligentes con sistema operativo Android?

Se define como **objeto de estudio**: Los sistemas domóticos empleando dispositivos móviles.

Para dar respuesta al problema de investigación se define como **objetivo general**: Adaptar el HMI-Ejecución del sistema Arex para ser empleado en el área de la domótica haciendo uso de dispositivos móviles inteligentes con sistema operativo Android.

Mientras que el **campo de acción** se enmarca en: HMI-Ejecución para sistemas domóticos empleando dispositivos móviles inteligentes.

Para dar cumplimiento al objetivo se propone el desarrollo de las siguientes **tareas de investigación**:

1. Determinación de los referentes teóricos en que se fundamenta el desarrollo de aplicaciones HMI-Ejecución para sistemas domóticos en dispositivos móviles.

2. Selección de las tecnologías y herramientas adecuadas para el desarrollo.

    - Sistema operativo.
    - Lenguaje de programación.
    - Lenguaje de modelado.
    - Metodología de desarrollo de software.

3. Definición de la propuesta de solución.

4. Diseño e implementación del sistema.

5. Diseño de pruebas de software.

    - Pruebas de aceptación.
    - Pruebas de rendimiento.

6. Evaluación de los resultados de las pruebas de software.

En la investigación se combinan diferentes **métodos de investigación**, los fundamentales se mencionan a continuación:
Análisis-síntesis, aplicado en el estudio de los fundamentos y teorías relacionadas con la domótica y con los sistemas de medición de datos.
Observación participante, en el seguimiento del desarrollo de aplicaciones en el área de la domótica haciendo uso de dispositivos móviles inteligentes.
Análisis de documentos, en la consulta de literatura especializada relacionada con la teoría de la automatización de procesos.
Histórico-lógico, en el estudio de la evolución de diferentes sistemas domóticos.
Investigación-acción, para la constante realización de pruebas de concepto y prototipos tanto no funcionales como funcionales.

La tesis está estructurada en introducción, tres capítulos, conclusiones y recomendaciones. Los capítulos son los siguientes:

- Capítulo 1. Fundamentación teórica. En este capítulo se hace referencia a los conceptos y teorías fundamentales relacionadas con el desarrollo de aplicaciones para dispositivos móviles inteligentes con sistema operativo Android y dirigidas a la domótica.

- Capítulo 2. Diseño e implementación. Este capítulo expone la propuesta de solución al problema planteado en la introducción. Se describen las modificaciones que realizadas al HMI-Ejecución de Arex. Además se detallan

los procesos de ingeniería de software, análisis, y diseño que se involucran para el desarrollo.

- Capítulo 3. Diseño y realización de pruebas al sistema. Para comprobar el correcto funcionamiento del sistema se realizaron un conjunto de pruebas de aceptación y de rendimiento. En este capítulo se describen los pasos, condiciones de ejecución y resultados obtenidos de las mismas.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

En este capítulo se introducen los elementos teóricos fundamentales necesarios para la realización del presente trabajo. Estos conceptos están relacionados con los sistemas domóticos, sistemas de medición de datos y fundamentalmente con el desarrollo de aplicaciones para dispositivos móviles. Además se describen las principales herramientas y tecnologías utilizadas en el desarrollo de la solución.

1.1. Sistema de medición de datos Arex

Un Sistema de Medición de Datos (SMD) es la colección de instrumentos, indicadores, normas, operaciones, métodos, herramientas, software, personal, ambiente y suposiciones usadas para cuantificar las unidades de medición o evaluación de la característica que se mide. Un proceso de medición puede ser visto como un proceso de manufactura que produce números (datos) para sus producciones [7]. El objetivo de un SMD es indicar el valor de una o varias variables. Para ello toma del medio medido la información y la transforma en una cantidad obtenida a través de un sensor [8].

Por otra parte un Sistema de Adquisición de Datos está diseñado fundamentalmente para la medida y el análisis de una o varias magnitudes físicas, características de un sistema real, tomando varias muestras de las mismas (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otro dispositivo de cómputo (sistema digital). Estos sistemas ofrecen flexibilidad de procesamiento, posibilidad de realizar las tareas en tiempo real o en análisis posteriores (a fin de analizar los posibles errores), gran capacidad de almacenamiento, facilidad de automatización, rápido acceso a la información y toma de decisión, con su utilización se adquieren gran cantidad de datos para poder analizar, posibilidad de emular una

gran cantidad de dispositivos de medición y es posible activar varios instrumentos al mismo tiempo [9].

Arex es un sistema de medición de datos para medir procesos de baja y mediana complejidad, a partir de la adquisición y procesamiento de datos en tiempo real, que puede ser ejecutado en hardware con bajas prestaciones. Para lograr que el sistema aumente la competitividad, es necesario que los despliegues gráficos puedan ser visualizados sobre teléfonos inteligentes o similares, debido al auge que han alcanzado estos dispositivos en el mundo de la tecnología moderna. Para medir el proceso de forma continua Arex debe:

- Recolectar y procesar la información recibida, en forma continua y confiable.
- Representar gráficamente y de forma animada los datos y las variables de proceso.
- Monitorizar las variables de proceso por medio de alarmas.
- Configurar y modificar la evolución del proceso (variables, alarmas, despliegues).
- Transmitir información a dispositivos de campo y otros.
- Alertar al operador de cambios detectados durante un proceso, tanto aquellos que no se consideren normales (alarmas) como cambios que se produzcan en la operación diaria (eventos).
- Almacenar los datos de las variables, eventos y alarmas ocurridas en el sistema.

Arex está formado por cuatro elementos fundamentales: HMI-Edición, HMI-Ejecución, Recolector y Tarjetas de adquisición de datos. Se tienen en cuenta aspectos esenciales en este tipo de sistemas, por ejemplo el manejo de puntos y alarmas. La comunicación con las tarjetas de adquisición se realiza mediante las interfaces de red Ethernet y Serie sobre el protocolo Modbus en sus variantes TCP y RTU.

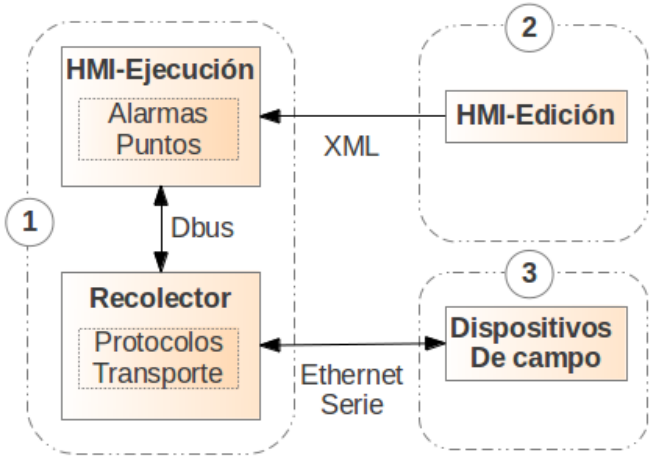

Figura 1.1: Vista general de la arquitectura de Arex.

A continuación se describen brevemente los principales módulos de Arex [10].

- Recolector: Este módulo es el encargado de gestionar la comunicación con las Tarjetas de adquisición de datos mediante el protocolo Modbus en sus variantes TCP y RTU. El recolector recibe la configuración de los dispositivos además de sus variables y procede mediante un manejador a la creación de bloques de encuesta, actividad que se realiza atendiendo a ciertos parámetros como son: el tipo de dato, el periodo de muestreo de las variables, el tamaño máximo permitido, la densidad del bloque, entre otros. Este componente de adquisición posee un planificador que permite adicionar y ejecutar tareas de lectura y de escritura sobre las tarjetas de adquisición de datos. La información de los valores de las variables se publica en un servidor de datos.

- HMI-Edición: Este módulo puede ser ejecutado en una computadora convencional, permite al usuario crear y configurar despliegues que representen lógicamente el ambiente donde va a ser desplegado el sistema. Permite además la configuración de los dispositivos, sus variables asociadas así como las alarmas. Las variables son vinculadas a elementos gráficos que forman parte de los despliegues. Finalmente las configuraciones son exportadas en formato XML.

- HMI-Ejecución: Permite la representación en tiempo de ejecución de los

procesos mediante la actualización de los componentes gráficos ubicados en los despliegues. Permite además la generación de sumarios de puntos y ofrece opciones para el manejo de las alarmas generadas en el sistema. Otra de las funcionalidades soportadas es el envío de comandos de escritura, que posibilita al operador ejercer control sobre los procesos.

El sistema provee las funcionalidades necesarias para contribuir a la seguridad de un local, tributar al ahorro energético y al confort de los habitantes de un recinto mediante la medición de variables como: estado de las luces, temperatura, humedad, apertura o cierre de puertas y ventanas, detección de intrusos (presencia), detección de humo, detección de fuga de gas, detección de inundaciones, detección de polvo, etc.

Arex es un sistema de propósito general para la automatización de diversos procesos, pero este trabajo centra su interés en el campo de la domótica, que es el área de aplicación donde en los últimos años se han introducido con mayor fuerza las aplicaciones sobre dispositivos móviles inteligentes.

1.2. Domótica

La domótica es la aplicación de procesos o sistemas inteligentes para viviendas, los mismos tienen como finalidad gestionar diversas áreas, por ejemplo: comunicación, información, seguridad, iluminación, climatización, electrodomésticos; generando comodidad, seguridad y ahorro de energía. La domótica ha logrado unir la automatización y las comunicaciones, generándose con esto un protagonismo de la misma en los últimos años en lo relacionado al desarrollo tecnológico [11].

Existen dicímiles definiciones de la domótica, por ejemplo, según la Real Academia Española, es el conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda. Por otra parte Jeffer Chaparro [12] se refiere a la domótica como la adopción, integración y aplicación de las nuevas tecnologías informáticas y comunicativas al hogar. Incluye principalmente el uso de electricidad, dispositivos electrónicos, sistemas informáticos y diferentes dispositivos de telecomunicaciones,

incorporando la telefonía móvil e internet.

Las características que gobiernan a estos tipos de sistemas y que hacen que se estén convirtiendo en una tendencia mundial son [13]:

- Interacción: se refiere a la comunicación entre el usuario y el sistema, interactuando con el mismo a través de una interfaz gráfica donde se muestra toda la información necesaria.

- Simplicidad y facilidad de uso: la interfaz es un punto clave para la interacción con el usuario, siendo esta sencilla e intuitiva, el sistema debe estar adecuado al usuario final, con el propósito de aumentar el confort.

- Flexibilidad: es la capacidad de añadir nuevos servicios y elementos a los sistemas existentes. Un edificio inteligente ha de prever que sus usuarios tendrán nuevas necesidades en el futuro y debe tener un margen de aceptación de nuevos elementos que ayuden a cubrir dichas necesidades sin tener que rediseñar completamente la instalación.

- Fiabilidad: la implementación de los sistemas domóticos se efectúa por medio de módulos independientes, lo que genera mayor fiabilidad en el sistema, ya que algún fallo en uno de los módulos no va a afectar a otro, y la implementación de módulos nuevos, no perjudicará los anteriores.

- Interrelación: a pesar de que el sistema está dividido por módulos, este debe poder integrarse con el resto de los subsistemas para facilitar la comunicación y permitir el intercambio de información.

- Tele-operación o manejo a distancia: a través de un esquema de comunicación con los distintos equipos (mando a distancia, bus de comunicación, etc.), reduce la necesidad de moverse dentro de la vivienda, este hecho puede ser particularmente importante en el caso de personas de la tercera edad o discapacitadas. Además permite al usuario un mejor aprovechamiento de su tiempo.

Aunque se pudiera pensar que este tipo de tecnología es sólo un lujo para personas acaudaladas, tiene importantes ventajas por las que puede ser una magnífica inversión dentro de una organización, sin importar lo pequeña que sea, ya que proporciona una gran variedad de ventajas, entre las que se pueden destacar el ahorro de energía, el incremento en los niveles de seguridad, mayor y mejor control centralizado sobre todas las áreas o habitaciones, mejor comunicación, optimización de recursos, automatización, ahorro de tiempo, calidad de vida y sobre todo un alto grado de confort. Algunos de los elementos que se pueden controlar en una casa domótica son los siguientes: la iluminación, la climatización, puertas, ventanas, persianas, electrodomésticos o el suministro de agua, gas y electricidad. En general, puede llegar a controlarse todo aquello que se desee, aún no se ha puesto un límite a la domótica. Para ello, la misma usa multitud de dispositivos que pueden ser distribuidos por todo el local en función de las necesidades de los clientes. Básicamente estos dispositivos se pueden dividir en sensores, actuadores y controladores [13].

- **Sensores:** dispositivos electrónicos, capaces de detectar señales físicas o químicas, y transformarlas en señales eléctricas para ser procesadas e interpretadas. Pueden ser sensores de detección de presencia, de medición de la temperatura, de detección de humo, etc.

- **Actuadores:** dispositivos capaces de recibir una orden del controlador y realizar la acción pertinente utilizando cualquier tipo de energía para activar un elemento final, como un motor, una válvula, entre otros.

- **Controladores:** sistemas que tienen como fin manejar un hardware, proporcionando una interfaz amigable para el usuario.

Con el auge de la telefonía móvil y la aparición de teléfonos inteligentes se ha hecho común el desarrollo de aplicaciones para realizar control automatizado de procesos sobre estas tecnologías.

1.2.1. Aplicaciones domóticas para dispositivos móviles

Paralelo al auge de la domótica se está produciendo el auge de los teléfonos inteligentes (smartphones) que permiten una gran variedad de interacciones, son cada vez más potentes y también más asequibles. Disponen, además, de una alta variedad de soluciones de conectividad, son más ligeros y su autonomía es mayor [14]. Actualmente, la movilidad es un componente plenamente integrado en los sistemas domóticos, utilizándose los dispositivos móviles como sistema de control. Para facilitar la utilización de las soluciones domóticas, los proveedores de las mismas ofrecen aplicaciones móviles compatibles con los sistemas de control que desarrollan. De este modo, un usuario puede enviar una orden desde su dispositivo móvil (smartphone o tableta táctil) al sistema de comando para controlar un aparato a distancia. Seguidamente se citan algunas de las aplicaciones de domótica para dispositivos móviles:

- Loxone Smart Home Automation: La solución domótica Loxone, es un sistema de origen austriaco desarrollado por la empresa Loxone Electronics, fundada en el 2008 que hace hincapié en el desarrollo de aplicaciones para domótica. El núcleo de esta aplicación es un módulo llamado miniserver, a través del cual se realizan las operaciones de configuración y control para lo que incluye un servidor web, disponible en internet. Es un sistema propietario que cuesta alrededor de 500 euros por cada habitación que se quiera automatizar [2].

- WeMo Light Switch: A través de la wifi, WeMo Light Switch, permite encender y apagar las luces de las habitaciones automatizadas desde cualquier lugar. El sistema reemplaza a los interruptores de luz estándar en un recinto cerrado y puede ser controlado de forma remota con un dispositivo móvil inteligente. Wemo trabaja con una red wifi y en cualquier lugar usando un teléfono inteligente o tableta tiene una conexión a internet (3G o 4G LTE). El sistema es compatible con dispositivos de la compañía Apple que usan sistema operativo iSO 5 o versiones más actualizadas. Además funciona para dispositivos con

sistema operativo Android 4.0 o superior. Para su utilización se requiere un router wifi con una conexión a internet por un precio en total de 49.99 dólares americanos [4].

- DroidSeer X10 Home Automation: Solución domótica para dispositivos móviles con sistema operativo Android. Desarrollado por la compañía SPVSOFT de origen ucraniana. Es un control remoto para la famosa aplicación HomeSeer y requiere la misma en la versión 2.x o superior. Una vez configurado, el usuario será capaz de controlar cualquier dispositivo que esté vinculado al sistema HomeSeer; además de leer los estados de los dispositivos como termostatos, puede mostrar los datos en una aplicación Android. Droidseer está diseñado para ser fácil de usar y permite administrar una casa desde cualquier lugar donde se puede obtener una conexión a internet, se puede conseguir por un precio de 2.33 euros por su fichero apk [6].

- Planner Mobile: Planner es una aplicación de ABB que permite controlar y supervisar en tiempo real viviendas desde cualquier lugar, ya sea desde la propia vivienda o desde cualquier lugar del mundo a través de internet. Con la aplicación móvil PLANNER MOBILE, desarrollada para dispositivos que usan el sistema Android a partir de la versión 2.3, se tiene la posibilidad de controlar funciones de una vivienda interactuando sobre un teléfono inteligente o tableta. Para poder controlar PLANNER desde la aplicación para dispositivos móviles inteligentes será necesario introducir una tarjeta SIM ligada a un plan de datos de cualquier operador en el módulo GSM de PLANNER [3]. Las funciones que brinda son las siguientes:

    1. Controles: Control de todos los circuitos que están creados en la vivienda. Se puede regular los ambientes de luz, subir y bajar las persianas de la vivienda, se puede activar el riego del jardín, etc.
    2. Ambiente: Permite la reproducción de todas las escenas que tenga generada la vivienda. Se podrán reproducir los ambientes de luz que mejor

se ajusten a cada momento, por ejemplo modo lectura, modo cine, etc.

3. Presencia: Permite la activación o desactivación de los cuatro circuitos de simulación de presencia. Simulando que hay presencia en la vivienda, por ejemplo si por cualquier motivo el dueño de la vivienda se tiene que ausentar durante un periodo prolongado de tiempo.

4. Alarmas: Permite la visualización y el control del estado de las alarmas, por ejemplo intrusión, alarmas técnicas (agua, gas, humo y fuego), alarma de caída de tensión. También se puede recibir un aviso de una alarma de emergencia, pánico o enfermedad que se podría reproducir en la vivienda por parte de un familiar en apuros.

5. Control de temperatura: Permite el control total de la temperatura de la vivienda al visualizar en todo momento la temperatura de la misma y el estado en el que se encuentra. Pudiendo controlarla de una forma automática si se desea.

La mayoría de estos sistemas son de carácter propietario, generalmente implementados para la comunicación con redes de sensores o elementos de hardware, que aunque ofrecen excelentes prestaciones, son muy costosos. A pesar de que se han mencionado algunas aplicaciones, en esta área existe una gran cantidad de sistemas afines. Como una limitación generalizada se puede mencionar que en su mayoría están diseñados a la medida para determinados procesos, pues dentro de la domótica hay una importante diversidad de ambientes y por tanto soluciones. Este último aspecto queda resuelto con el sistema Arex, que viene siendo un configurador de soluciones que aplicado al área de la domótica puede adaptarse y dar solución a una amplia diversidad de requerimientos.

En la actualidad existe un importante número de aplicaciones en el área de la domótica cuyos HMI-Ejecución se implementan sobre dispositivos móviles inteligentes, en este sentido han ganado especial espacio las que se encuentran desarrolladas para el sistema operativo Android, aunque de igual forma para otros

sistemas operativos se pueden encontrar aplicaciones de este tipo. Android, iOS, Windows Phone, BlackBerry OS, todos ellos tienen algo en común: son los sistemas operativos que mueven la actual generación de teléfonos móviles e inteligentes. A continuación se muestra un análisis detallado de los elementos más importantes que los caracterizan.

1.3. Sistemas operativos para dispositivos móviles

Los sistemas operativos para dispositivos móviles se vuelven cada día más importantes ya que con el crecimiento y avance de la tecnología es imprescindible contar con sistemas que ofrezcan al usuario un acceso de mayor facilidad y confort. Los sistemas operativos móviles son más simples que los empleados en las computadoras, están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos [15]. Existen numerosas plataformas de sistemas operativos para el desarrollo de aplicaciones móviles, varían en cuanto al lenguaje de programación que utilizan, las herramientas disponibles, la forma de generar los gráficos y presentarlos en la interfaz de usuario. Cada uno de estos sistemas cuenta con un mercado de aplicaciones independiente donde algunas son gratis y otras no. Android es una de estas plataformas, que rompió paradigmas al ser liberado su código de fuente; existen además Black Berry OS, iOS, Palm OS, Windows Phone, entre otros.

A *priori* se puede pensar que la variedad de sistemas operativos existente obedece a una directriz de las compañías por defender sus estándares y afianzar su posición en el mercado, lo cierto es que cada uno de ellos ofrece características diferentes y están pensados para públicos distintos. Aunque el presente trabajo es desarrollado para dispositivos con sistema operativo Android a continuación se mencionan algunos de los principales sistemas operativos para dispositivos móviles que actualmente cuentan con un mayor reconocimiento en el mundo tecnológico; así como características de los mismos.

1.3.1. IOS

IOS (por sus siglas en inglés, iPhone/iPod/iPad Operating System) es un sistema operativo desarrollado por Apple [16], inicialmente sólo para el teléfono inteligente de la compañía (el iPhone), luego fue extendido a otros dispositivos como el iPod Touch y el iPad. Lo que caracteriza a iOS frente a otros es que es un sistema operativo cerrado. Apple no permite que se modifiquen características internas del sistema más allá de las limitadas opciones que da en los ajustes. Un sistema cerrado permite, sin embargo, ofrecer siempre una experiencia más estable y segura como la diseñó el fabricante en un principio. Sin embargo a muchos usuarios, que buscan una mayor personalización, se les pueden quedar cortas las opciones que le da Apple. La principal ventaja de este modelo es la adaptación total del sistema operativo al dispositivo, puesto que ambos han sido concebidos como partes de un todo, por lo tanto ofrecen la misma experiencia de uso para todos los consumidores. Por otro lado, como también suele ser habitual en los productos de la empresa, no se licencia a terceros.

1.3.2. Windows Phone

Desarrollado por Microsoft como sucesor de Windows Mobile. Windows Phone tiene el mérito de ofrecer un diseño fresco y diferente. La interfaz de usuario de Windows Phone es altamente intuitiva y sencilla de usar a tal punto que en un par de minutos cualquier usuario reconoce la mayoría de los gestos y formas de utilizar el sistema operativo. Como fabricante del sistema, Microsoft requiere que todo teléfono que desee ejecutar Windows Phone disponga de unas características mínimas, para asegurar la consistencia de todos los usuarios del sistema [17].

- Procesador Qualcomm Snapdragon S4, procesador de doble núcleo.
- Memoria RAM de 512 MB para pantallas WVGA y de 1 GB de RAM mínimo para pantallas 720p / WXGA.
- Memoria de almacenamiento interno de al menos 4 GB.

- GPS y A-GNSS. También es compatible con GLONASS.
- Entrada de auriculares de 3.5 mm.
- Cámara en la parte trasera con autoenfoque y flash LED o Xenon. Opción de cámara delantera de calidad VGA como mínimo e indispensable botón dedicado a la cámara.
- Sensores de proximidad y luminosidad, acelerómetro y posibilidad de vibración.
- Conectividad wifi B/G (N es opcional) y Bluetooth.
- Gráficos DirectX con soporte de hardware y aceleración Direct3D para utilizar GPU programable.
- Pantalla Multi-Touch capacitiva con un mínimo de cuatro puntos simultáneos.

A partir de estas características, los fabricantes son libres de ampliarlas en algunos casos y están obligados a cumplirlas con exactitud en otros. Cuenta con una reducida variedad de dispositivos móviles que utilizan este sistema operativo. Actualmente Microsoft ha dejado atrás el nombre Windows Phone para hablar sólo de Windows 10 [18]. Así, Microsoft quiere que sus clientes sientan que, no importa si están usando un smartphone, una tablet o una PC, están navegando con Windows 10.

1.3.3. BlackBerry OS

Anteriormente conocido como RIM. Es un sistema operativo de código cerrado para móviles, desarrollado por BlackBerry; para los dispositivos BlackBerry. El SO está claramente orientado a su uso profesional como gestor de correo electrónico y agenda. Al igual que en el SO Symbian, desarrolladores independientes también pueden crear programas para BlackBerry, pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrollador.

1.3.4. Android

El sistema operativo de Google, número uno en cuanto a popularidad, se caracteriza por ser de código abierto, gratuito, no requiere pago de licencias y disponible para cualquier fabricante interesado en utilizarlo para sus dispositivos móviles. Ofrece un fácil acceso al Sistema Operativo mostrando una interfaz gráfica práctica y didáctica para todos [19]. Analizando el mercado actual de teléfonos inteligentes, un gran número de los usuarios de teléfonos móviles están optando por los teléfonos con Android. Esta disponibilidad ha traído consigo encontrar innumerables dispositivos de miles de formas y funcionalidades con todas las versiones de Android existentes. Éste ha logrado consolidarse en el mercado como líder absoluto. Dada la posibilidad de que Android pueda instalarse prácticamente en todo tipo de dispositivos, sean móviles, portátiles e incluso microondas, hace que Android siempre esté presente en los terminales más potentes del mercado, siendo una apuesta importante por fabricantes y operadoras por la posibilidad de que independientemente del potencial, gama o prestaciones del dispositivo, Android podría adaptarse a la perfección a todo tipo de necesidades.

En Cuba el uso masivo de la tecnología móvil ha crecido considerablemente durante los últimos años, debido al aumento de las ventas de líneas y teléfonos celulares a toda la población. La Empresa de Telecomunicaciones de Cuba S.A (ETECSA) es la encargada de comercializar los accesorios y servicios de telecomunicaciones en el país. Ésta cuenta con entidades colaboradoras como la Universidad de las Ciencias Informáticas (UCI) y DESOFT para el desarrollo de aplicaciones para dispositivos móviles. El mismo en muchos casos se ha visto afectado por las consecuencias del bloqueo económico, que se evidencian en la denegación del acceso a servicios informáticos tales como sitios web y foros tecnológicos, además en la difícil adquisición de algunas tecnologías modernas. Por tal motivo el país está inmerso en el proceso de alcanzar la soberanía tecnológica mediante la migración a software

libre, ya que su utilización no implica gastos adicionales por concepto de cambio de plataforma ni la adquisición de licencias de software privativos [20].

1.4. Herramientas y tecnologías

Para el desarrollo del presente trabajo de diploma se utilizó un conjunto de herramientas de software libre de las cuales se expone a continuación una breve descripción.

1.4.1. Entorno de Desarrollo Integrado

Un Entorno Integrado de Desarrollo (IDE, Integrated Development Environment) es una aplicación que consiste generalmente en la combinación de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede ser exclusivo para un solo lenguaje de programación o ser utilizado para varios.

QtCreator es un IDE multiplataforma hecho a la medida para los desarrolladores de Qt. Se puede ejecutar sobre los sistemas operativos Windows XP y versiones superiores, GNU/Linux 2.6.X y Mac OS X 10.4 o superior, permitiendo crear aplicaciones para diferentes plataformas de escritorio y de dispositivos móviles. Una de las mayores ventajas del QtCreator es que permite a los equipos de desarrolladores compartir un proyecto a través de diferentes plataformas de desarrollo (Microsoft Windows, Mac OS X y Linux) mediante una herramienta común para el desarrollo y la depuración. El objetivo principal de QtCreator es satisfacer las necesidades de los desarrolladores de Qt que buscan simplicidad, facilidad de uso, productividad y extensibilidad [21]. Para el desarrollo del presente trabajo se utiliza en su versión 3.1.2.

1.4.2. Lenguaje de programación

Un lenguaje de programación es un lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana [22].

QML (en inglés, Qt Meta Language) es un lenguaje basado en JavaScript creado para diseñar aplicaciones enfocadas a la interfaz de usuario. Es parte de Qt Quick, el kit de interfaz de usuario creado por Nokia junto al framework Qt. El lenguaje QML se usa principalmente para aplicaciones móviles, donde la entrada táctil, las animaciones fluidas y una buena experiencia de usuario son cruciales.

C++ es un lenguaje de programación de propósito general, de la familia de los lenguajes imperativos y orientados a objetos, basado en el lenguaje de programación C, al que se le han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, funciones inline y sobrecarga de operadores. Es un lenguaje potente para el desarrollo de aplicaciones críticas o complejas, por la facilidad que brinda la manipulación de la memoria del ordenador a través de los punteros. Como lenguaje orientado a objetos brinda ventajas propias de esta filosofía de programación, como son: la abstracción, el encapsulamiento, la herencia y el polimorfismo. Lo que nos permite entre otras funcionalidades, poder representar un conjunto de características y comportamientos similares de objetos de la vida real y agruparlos en una unidad básica, brindando una interfaz para permitir la interacción con la misma[23].

Para el desarrollo de los componentes gráficos se utiliza QML como lenguaje de programación; mientras que C++ es empleado para el desarrollo del mecanismo de comunicación entre el HMI-Ejecución y el Recolector.

1.4.3. Lenguaje de modelado

Los lenguajes de modelado poseen como objetivo principal visualizar de manera gráfica el sistema que se desarrollará, como ayuda técnica a los ingenieros implicados. Éstos pueden utilizarse a la hora de la comunicación con el o los clientes, grupo de desarrollo y otros factores que intervengan en este proceso.

En este trabajo se utiliza el **Lenguaje Unificado de Modelado**, que es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Es uno de los lenguajes de modelado de sistemas

de software más conocidos y utilizados en la actualidad, respaldado por el Grupo de Administración de Objetos (OMG, por sus siglas en inglés). Proporciona ventajas en la representación del ciclo de vida de un software y de los artefactos específicos del Proceso Unificado de Desarrollo del Software. Posibilita la comunicación sencilla y rápida entre programadores y los clientes del software que se desarrolla, brindando la posibilidad de que estos últimos puedan expresar su conformidad con el producto o las nuevas mejoras que desean ver introducidas.

1.4.4. Framework de desarrollo

Qt es un framework multiplataforma ampliamente usado, desarrollado como un software libre y de código abierto a través de Qt Project. Qt es distribuido bajo los términos de GNU Lesser General Public License (y otras). Utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en varios otros lenguajes de programación. Para el desarrollo del presente trabajo se utiliza en su versión 5.3.1.

1.4.5. Herramienta CASE

Visual Paradigm es una de las herramientas CASE (Cumputer Aided Software Engineering), considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo de software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Para el desarrollo del presente trabajo se utiliza en su versión 8.0.

1.4.6. Android NDK y SDK

Para compilar el HMI-Ejecución para Android en la presente investigación se utiliza el SDK r24.2 de Android y el NDK r10c en sus versiones para Linux.

El **NDK** (en inglés, Native Development Kit) de Android es un conjunto de herramientas que permiten implementar partes de una aplicación utilizando lenguajes de código nativo, como C y C ++. Típicamente el NDK se utiliza para el desarrollo de aplicaciones críticas desde el punto de vista de utilización de recursos de

hardware como el CPU y la RAM, tales como procesamiento de señales, simulaciones de procesos físicos, etc [24].

El **SDK** (en inglés, Software Development Kit) de Android provee todas las herramientas necesarias para la compilación, traceo y realización de pruebas para Android sobre Windows, Mac y Linux. Está compuesto por diferentes módulos o paquetes que se pueden descargar por separado, algunos de los más relevantes son los que se listan a continuación:

- SDK Tools
- SDK Platform-tools
- SDK Platform
- System Images

1.4.7. Herramienta para la edición de imágenes

GIMP (en inglés, GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la licencia pública general de GNU. Existen versiones totalmente funcionales para Windows, para Mac OS X, y se incluye en muchas distribuciones GNU/Linux. Se le puede considerar como una alternativa firme, potente y rápida a Photoshop para muchos usos, aunque no se ha desarrollado como un clon de él y posee una interfaz bastante diferente [25]. Para el desarrollo del presente trabajo se utiliza en su versión 2.8.

1.4.8. Metodología de desarrollo de software

El desarrollo de aplicaciones para proveer servicios móviles, difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas en esos casos difieran de las del software clásico. Las características especiales de los entornos móviles como el canal de radio, la capacidad de los terminales, la

portabilidad, el tiempo de salida al mercado, la movilidad del usuario, entre otras; exigen nuevas tendencias para desarrollar software [26].

Las metodologías ágiles poseen ciertas características que las hacen totalmente aplicables al dominio del software en los dispositivos móviles, surgieron como una solución inmediata para el desarrollo de software, garantizando la realización de proyectos en corto plazo [26].

Algunas metodologías ágiles son: Programación Extrema (XP), SCRUM, Crystal Clear, Método de Desarrollo de Sistemas Dinámicos (DSDM), Open Unified Process (OpenUP), entre otras. Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc), la Universidad de las Ciencias Informáticas (UCI) ha dictaminado que la metodología usada en todos los proyectos desarrollados en el centro debe ser una variación de la metodología Proceso Unificado Ágil (AUP, por sus siglas en inglés). A continuación se describe dicha metodología y las variaciones realizadas para adaptarla a las necesidades de desarrollo de software en la UCI.

Variación de AUP para la UCI

AUP es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Se basa en los siguientes principios [27]:

- Simplicidad: Todo se describe concisamente utilizando poca documentación.
- Agilidad: El ajuste a los valores y principios de La Alianza Ágil.
- Centrarse en actividades de alto valor: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
- Herramienta de la independencia: puede usarse cualquier conjunto de herramientas que se desee con el AUP. Se sugiere utilizar las herramientas

más adecuadas para el trabajo, que a menudo son las herramientas simples o incluso herramientas de código abierto.

- Adaptación para satisfacer necesidades propias: La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario una herramienta especial, o tomar un curso, para adaptar esta metodología.

AUP UCI es la metodología que se utiliza en este trabajo, la misma consta de tres fases [28]: Inicio, Ejecución y Cierre. Además define ocho disciplinas, que son: modelado de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación, pruebas de aceptación y despliegue.
Entre las técnicas ágiles que utiliza AUP se encuentra el modelado ágil, se hace uso de esta técnica para los proyectos que necesitan, por sus características, encapsular sus requisitos funcionales en historias de usuarios o en descripción de requisitos por procesos.

Conclusiones parciales

A partir de la revisión de algunas aplicaciones existentes se pudo verificar que los sistemas más completos aparecen sobre todo para el área de la automatización de hogares, y en su mayoría están diseñados como parte de redes de sensores y elementos de hardware costosos. De igual forma se pudo verificar que se han desarrollado un buen número de soluciones pequeñas, encaminadas sobre todo a dar solución a problemas puntuales, generalmente este tipo de aplicaciones están diseñadas a la medida con pocas posibilidades de extensión. Por último se hizo una elección de herramientas y tecnologías para el desarrollo de la solución, QtCreator como entorno de desarrollo integrado, C++ y QML como lenguajes de programación, UML como lenguaje de modelado y AUP UCI como metodología de desarrollo.

Capítulo 2

DISEÑO E IMPLEMENTACIÓN

Este capítulo expone la propuesta de solución al problema planteado en la introducción. Se describen las modificaciones realizadas al HMI-Ejecución de Arex, detallando los procesos de ingeniería de software, análisis, y diseño que se involucran en el desarrollo.

2.1. Descripción de la propuesta

El sistema Arex cuenta con cuatro módulos: HMI-Edición, HMI-Ejecución, Recolector y Tarjeta de adquisición de datos, estos fueron descritos en el Capítulo 1. Actualmente dos de dichos módulos, HMI-Ejecución y Recolector, sólo pueden establecer comunicación entre ellos mediante D-bus, que es un sistema de comunicación entre procesos (IPC) [29]. Existen variedades de tecnologías que son para la comunicación entre procesos, entre ellas se encuentra CORBA, DCE, DCOM, DCOP, XML-RPC, SOAP, MBUS, Internet Communications Engine (ICE), etc. Cada una de ellas ha sido diseñada para un tipo particular de aplicación. D-bus está diseñado para dos casos específicos [30]:

- Comunicación entre aplicaciones de escritorio en la misma sesión, facilitando la integración de aplicaciones dentro de un mismo entorno de escritorio y el tratamiento de asuntos relativos al ciclo de vida de procesos.

- Comunicación entre el sistema operativo y la sesión de escritorio, incluyendo dentro del sistema operativo al núcleo y algunos demonios o procesos.

Por tal motivo actualmente los módulos HMI-Ejecución y Recolector sólo pueden comunicarse si se encuentran sobre el mismo nodo de ejecución. Para una mejor comprensión se muestra la siguiente figura:

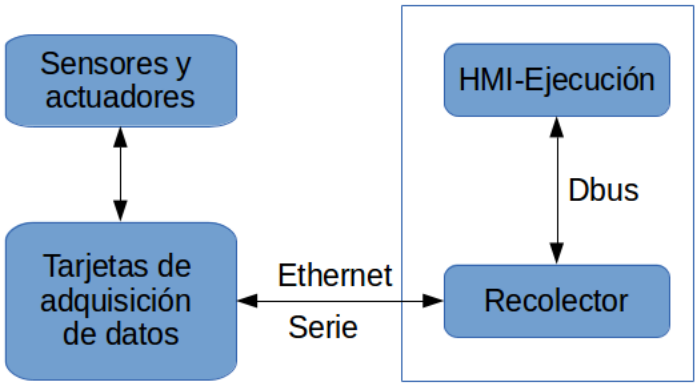

Figura 2.1: Comunicación entre los módulos HMI-Ejecución y Recolector del sistema Arex.

Teniendo en cuenta que el Recolector de Arex utiliza la interfaz Serie para la comunicación con las tarjetas de adquisición de datos, el mismo no puede ejecutarse sobre dispositivos móviles inteligentes, ya que por lo general este tipo de dispositivos no cuenta con interfaces seriales. De esta forma, para lograr que el HMI-Ejecución del sistema Arex pueda ser ejecutado sobre dispositivos móviles inteligentes, se propone que la comunicación entre los dos módulos mencionados anteriormente sea mediante el Protocolo de Control de Transmisión (TCP, por sus siglas en inglés), uno de los principales protocolos de la capa de transporte del modelo TCP/IP [31]. Este es un protocolo orientado a conexión, es decir, permite que dos nodos que están comunicados controlen el estado de la transmisión. El nodo que solicita la conexión generalmente se denomina cliente, y el receptor servidor. Por esta razón se suele decir que en este caso existe una comunicación en un entorno Cliente-Servidor. Los nodos de dicho entorno se comunican en línea y en ambas direcciones.

Por tanto se propone modificar la arquitectura actual del sistema Arex. Para ello es necesario la implementación del submódulo TCP-Server, integrado al Recolector y la implementación de otro submódulo TCP-Client que será incluido al HMI-Ejecución. De esta manera es posible ejecutar el HMI-Ejecución sobre cualquier dispositivo móvil inteligente con Android, que es uno de los principales objetivos de este trabajo.

Una vista arquitectónica de la solución se muestra en la Figura 2.2.

Se debe mencionar que el Recolector de Arex es el módulo encargado de la mayor parte del procesamiento en el sistema, la modificación propuesta permite además distribuir la carga de trabajo en dos nodos físicos, lo que debe contribuir de manera positiva al desempeño.

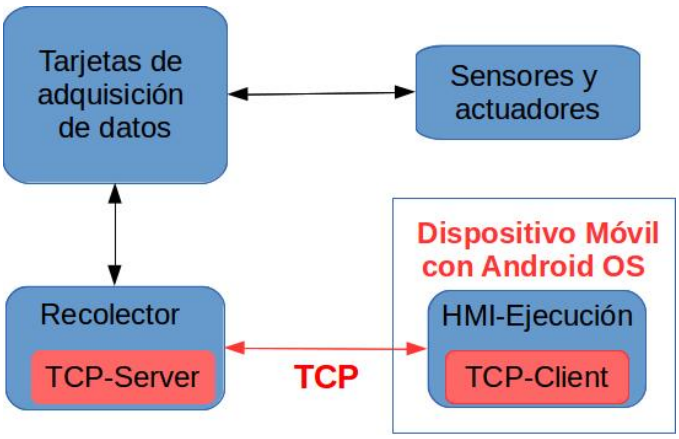

Figura 2.2: Vista general de la arquitectura.

Los elementos en color rojo son las modificaciones aportadas por la presente solución al sistema Arex en lo relacionado a la comunicación TCP.

2.2. Modelo arquitectónico

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución [32]. Es una vista estructural de alto nivel que define un estilo o una combinación de estilos para una solución. Se concentra en requerimientos no funcionales y resulta esencial para el éxito o fracaso de un proyecto.

Por su parte un estilo arquitectónico proporciona una plantilla de trabajo que unifica la manera en que todos los miembros del equipo ven el sistema e impone una transformación al diseño del mismo. Existen diferentes estilos, por ejemplo: de flujos de datos, centrado en datos, de código móvil, basados en eventos, orientados a servicios, entre otros. En este trabajo se utilizan dos estilos que pueden clasificarse dentro de la categoría de llamada y retorno, el estilo orientado a objetos y el estilo

de arquitectura basada en componentes (ver Figura 2.3). Los mismos plantean que los componentes de un sistema encapsulan los datos y las operaciones que deben aplicarse para manipular estos datos. La comunicación y coordinación entre los componentes se consigue mediante el paso de mensajes [33].

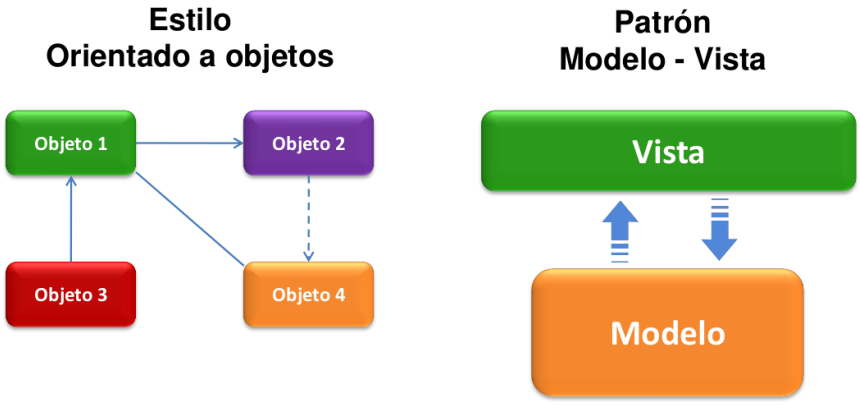

Figura 2.3: Estilo y patrón arquitectónico.

Como patrón arquitectónico se utiliza el principio de separación modelo-vista, que es una variante del patrón N-capas. En este caso la vista se representa por las ventanas y los componentes gráficos del HMI-Ejecución, mientras que el modelo es la lógica implementada mediante la cual se obtienen los valores de las variables y las alarmas (ver Figura 2.3).

Las clases de la vista son responsables de la entrada y salida, y de capturar los eventos de la interfaz gráfica de usuario, pero no mantienen datos ni proporcionan directamente ninguna funcionalidad de la aplicación. Es conveniente que no exista un acoplamiento directo de otros componentes con los objetos ventana, puesto que las ventanas están relacionadas con una aplicación específica, mientras que (idealmente) los componentes que no son ventanas podrían reutilizarse en nuevas aplicaciones o conectarse a una nueva interfaz [34].

2.3. Requerimientos del sistema

Entre los artefactos que define la metodología seleccionada se encuentran las historias de usuario, las cuales representan una breve descripción del comportamiento del sistema, emplean terminología del cliente sin lenguaje técnico, se realiza una por

cada característica principal del sistema, se emplean para hacer estimaciones de tiempo, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

2.3.1. Historias de usuario

A continuación se muestran las historias de usuario definidas para el sistema.

Historia de usuario	
Número:1	**Nombre del requisito:** Establecer la comunicación entre el Recolector y el HMI-Ejecución vía WIFI.
Programador: Yoenia Lapido Ramirez.	**Iteración Asignada:** 1
Prioridad: Alta	**Tiempo Estimado:** 20 días
Riesgo en Desarrollo:	**Tiempo Real:** 3 semanas
Descripción: En la versión actual el HMI-Ejecución no puede ser ejecutado en un nodo diferente al Recolector. Para que éste se pueda ejecutar sobre un dispositivo Android se requiere que la comunicación con el Recolector sea inalámbrica.	

Tabla 2.1: Historia de usuario 1

Historia de usuario	
Número:2	**Nombre del requisito:** Adaptar el módulo HMI-Ejecución para ser ejecutado sobre dispositivos móviles inteligentes con sistema operativo Android.
Programador: Yoenia Lapido Ramirez.	**Iteración Asignada:** 2
Prioridad: Alta	**Tiempo Estimado:** 30 días
Riesgo en Desarrollo:	**Tiempo Real:** 4 semanas
Descripción: La versión actual del HMI-Ejecución no se puede ejecutar sobre dispositivos móviles con sistema operativo Android. Para esto se deben realizar un conjunto de modificaciones que permitan crear una aplicación para Android en la que se ajusten las dimensiones de los objetos gráficos.	

Tabla 2.2: Historia de usuario 2

Historia de usuario	
Número:3	**Nombre del requisito:** Diseñar un grupo de componentes gráficos para la configuración de soluciones domóticas usando el Editor de Arex.
Programador: Yoenia Lapido Ramirez.	**Iteración Asignada:** 3
Prioridad: Alta	**Tiempo Estimado:** 30 días
Riesgo en Desarrollo:	**Tiempo Real:** 4 semanas
Descripción: Los componentes gráficos que actualmente usa el sistema Arex no son adecuados para construir despliegues en el área de la domótica. Por tanto es necesario el diseño de un grupo de componentes que cumplan con estas características.	

Tabla 2.3: Historia de usuario 3

2.3.2. Requisitos no funcionales

Un requisito no funcional es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar. Los requisitos no funcionales que se tuvieron en cuenta en la presente solución son los siguientes:

- El Recolector debe permitir como máximo 10 conecciones TCP.
- En el HMI-Ejecución se deben poder visualizar hasta 10 despliegues con 10 variables y 10 componentes cada uno.

2.4. Identificación y descripción de componentes gráficos

La versión actual del sistema Arex solamente cuenta con un reducido grupo de componentes gráficos, los mismos son genéricos, es decir que pueden emplearse en proyectos de diversa índole. En este caso se encuentran componentes como Button, Switch, DialControl, entre otros. Como se mencionó en la introducción, estos

componentes no fueron diseñados para ser visualizados en dispositivos móviles, y por otro lado resultan insuficientes para diseñar despliegues en el área de la domótica. Uno de los requisitos fundamentales en la utilización de dispositivos móviles para la domótica es mostrar una interfaz de usuario amigable, por esta razón es necesario diseñar e implementar componentes gráficos acordes a este tipo de escenarios. A continuación se realiza una breve descripción de los nuevos componentes que se proponen en este trabajo:

- **Estado de luminarias (LightState):** Este componente puede ser asociado a una variable *booleana* que indica el estado de las luminarias. Son dos estados posibles: *ON* cuando la luminaria está encendida y *OFF* cuando está apagada (ver Figura 2.4). Este componente permite el envió de comandos de escritura asociado al evento *onClick*. Es decir, cuando el componente se encuentra en un despliegue del HMI-Ejecución, si el operador da *click* sobre él inmediatamente se envía un comando de escritura con el valor opuesto al que tenía en el momento anterior a la acción. En un lenguaje coloquial se puede decir que se le da la orden a la luminaria de apagarse o encenderse.

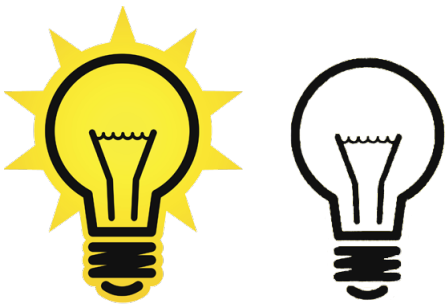

Figura 2.4: Componente LightState

- **Intensidad de la luz (LightIntensity):** Este componente se encarga de mostrar el valor de la intensidad de la luz (ver Figura 2.5). Debe asociarse a una variable en la que se reciba el valor correspondiente dado en *Lumens*(lm), que es la unidad de medida más utilizada para expresar la cantidad total de luz visible emitida por una fuente [35]. Este componente es para variables de solo lectura, es decir que no permite el envío de comandos.

Figura 2.5: Componente LightIntensity

- **Temperatura ambiente (Temperature):** Este componente permite representar, en grados *Celsius* y *Fahrenheit*, valores de temperatura. El componente debe ser asociado a una variable que almacene el valor de la temperatura en grados *Celsius*, a partir de éstos se calcula el valor en *Fahrenheit* y se realiza la visualización de ambos (ver Figura).

Figura 2.6: Componente Temperature

- **Presencia (Presence):** Entre los principales aportes de la domótica se encuentra el tributar a una mayor seguridad. Una de las vías que posibilita ésto es la instalación de sensores para la detección de movimiento en locales cerrados. El componente *Presence* se encarga de representar visualmente si se detecta o no movimiento en un local (ver Figura 2.7). Para ello debe estar asociado a una variable *booleana*, que a su vez reciba el valor de cualquier sensor destinado para estos fines.

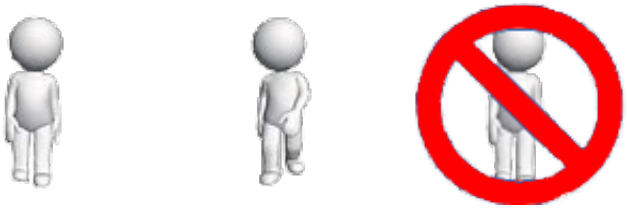

Figura 2.7: Componente Presence

- **Humedad relativa (Humidity):** Este componente se encarga de visualizar la humedad relativa dada en por ciento (ver Figura 2.8). *Humidity* debe asociarse con variables de solo lectura que reciban los valores correspondientes de sensores de humedad relativa.

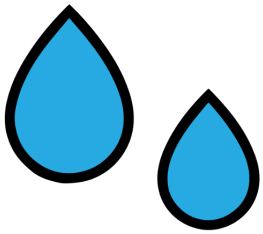

Figura 2.8: Componente Humidity

- **Contactos magnéticos para puertas y ventanas (DoorState, WindState):** Estos componentes se encargan de graficar el estado de apertura o cierre de puertas y ventanas (ver Figura 2.9). Deben asociarse a variables *booleanas*, dichas variables deben recibir los valores de sensores fabricados a partir de contactos magnéticos.

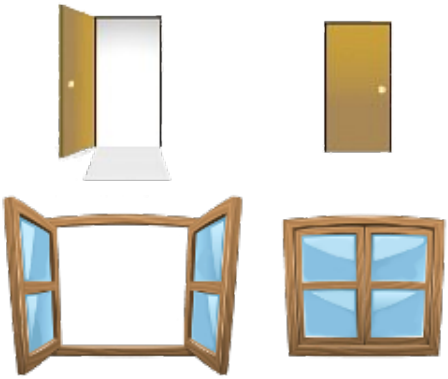

Figura 2.9: Componentes DoorState y WindState

- **Nivel de CO2 (CO2):** Este componente se encarga de visualizar el nivel de dióxido de carbono dado en por ciento (ver Figura 2.10). Debe asociarse a una variable que reciba el valor correspondiente de algún sensor. En aplicaciones relacionadas con el confort generalmente se mide el nivel de CO2, para determinar cuándo se debe ventilar el local y evitar que las personas se expongan a un ambiente enrarecido.

Figura 2.10: Componente CO2

2.5. Implementación de los componentes gráficos en QML

Para el desarrollo de los componentes gráficos se siguió el mismo patrón de implementación utilizado en los componentes de Arex. Todos los componentes utilizan una variable de tipo *property* para almacenar el valor de la variable que se asocia a cada componente. La misma se nombra *m_value*, ya que el HMI-Editor del sistema Arex reconoce a las variables que tienen el prefijo *m_* como editables. En el caso de los componentes CO2, Humidity, Temperature y LightIntensity la propiedad *m_value* es de tipo *real*, mientras que el resto de los componentes es de tipo *bool* (ver Figura 2.11).

```
property real m_value : 0
signal changeValue(variant p_value)
```

Figura 2.11: Implementación de componentes gráficos

En el caso del componente LightState, que es el único que permite el envío de comandos, además de la propiedad *m_value*, cuenta con una señal (*signal*) que se emite cuando se acciona sobre el componente para enviar un comando (ver Figura 2.11).

2.5.1. Animaciones

En el caso de los componentes *LightState* y *Presence* se implementaron animaciones sencillas de tipo *SequentialAnimation*, que es un tipo de animación definido en QML y hereda directamente de la clase *Animation*. Como su nombre indica, una *SequentialAnimation* permite realizar animaciones secuenciales de diferentes

tipos y comportamientos. En el caso de los componentes mencionados se utiliza *SequentialAnimation* para realizar transacciones sobre algunos de sus atributos. En *LightState* la animación es a partir de la modificación de las dimensiones (*width, hight*) (ver Figura 2.12). Mientas que en *Presence* la animación se realiza variando el atributo *source* (ver Figura 2.13).

```
SequentialAnimation on source {
    id: seq1
    running: false
    loops: Animation.Infinite
    PropertyAnimation { to: "walk2.png" ; duration: 500}
    PropertyAnimation { to: "walk1.png" ; duration: 500}
}
```

Figura 2.12: Código de la animación en el componente LightState

```
SequentialAnimation on width {
    id: seq1
    running: false
    loops: Animation.Infinite
    PropertyAnimation { to: light.width+5 ; duration: 500}
    PropertyAnimation { to: light.width ; duration: 500}
}
SequentialAnimation on height {
    id: seq2
    running: false
    loops: Animation.Infinite
    PropertyAnimation { to: light.height+5 ; duration: 500}
    PropertyAnimation { to: light.height ; duration: 500}
}
```

Figura 2.13: Código de la animación en el componente Presence

2.6. Diagramas de clases

Para lograr un mejor entendimiento de la solución a continuación se describe brevemente el diseño de clases de los submódulos TCP-Server incorporado al Recolector y TCP-Client implementado como parte del HMI-Ejecución.

En el diagrama que se observa en la Figura 2.14 la clase Collector representa a la clase principal y controladora del módulo Recolector de Arex, esta clase es la encargada

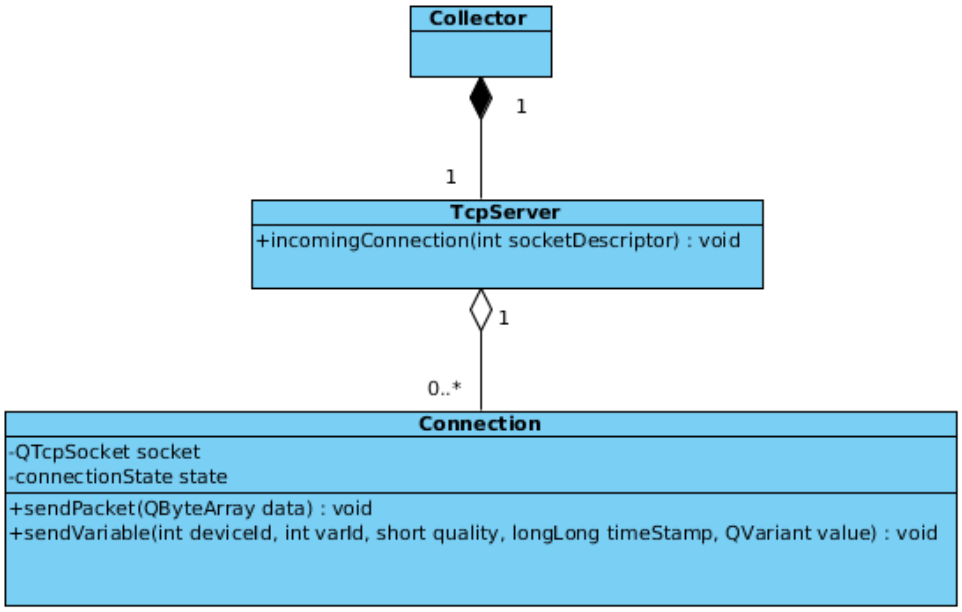

Figura 2.14: Diagrama de clase del submódulo TCP-Server

de crear una instancia de la clase TcpServer. TcpServer tiene la responsabilidad de iniciar un servidor de conexiones TCP. Las conexiones con los clientes son atendidas por instancias de la clase Connection. Cuando la conexión se cierra, o al ocurrir algún error en la comunicación se elimina la instancia de Connection correspondiente. En su conjunto el submódulo TCP-Server tiene la responsabilidad de atender a las peticiones de conexión, recibir comandos de escritura y notificar el estado de las variables configuradas en el sistema.

El diseño del submódulo TCP-Client se observa en la Figura 2.15. La clase TcpClient tiene la responsabilidad de establecer comunicación mediante el protocolo TCP con el Recolector de Arex. Al establecer la comunicación con el mismo se reciben las notificaciones de cambio en los atributos y valores de las variables. Por otro lado permite el envío de comandos de escritura hacia el Recolector.

2.7. Patrones de diseño

Se puede decir que los patrones de diseño son soluciones a diferentes clases de problemas conocidos que pueden ser aplicados a diferentes códigos. Simplificando, un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones

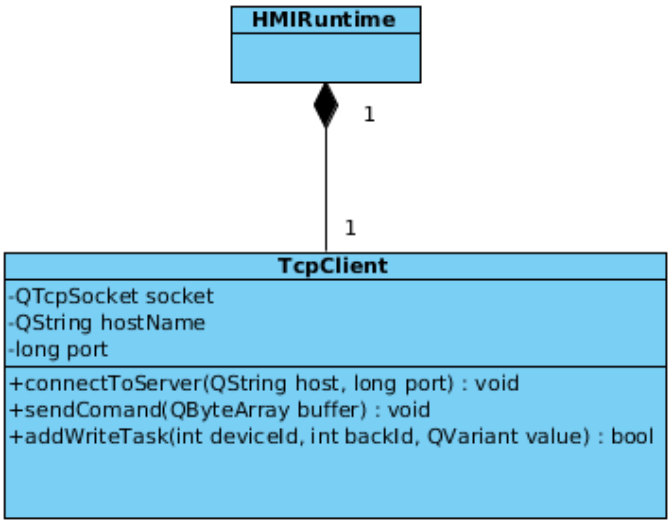

Figura 2.15: Diagrama de clase del submódulo TCP-Client

sobre sus compromisos [34]. Entre estos patrones se encuentran los conocidos como patrones GoF (Gang of Four) y los GRASP (General Responsibility Assignment Software Patterns), que ayudan a elegir las clases adecuadas y a decidir cómo éstas deben interactuar. En los acápites siguientes se exponen brevemente algunos de los patrones que se utilizan en la solución propuesta.

2.7.1. Patrones GoF

- **Singleton:** Este patrón está diseñado para restringir la creación de objetos de una misma clase, específicamente permite la creación de un único objeto y provee un solo punto de acceso global para acceder al mismo (ver Figura 2.16).

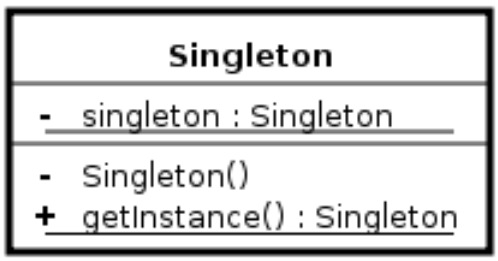

Figura 2.16: Esquema general del patrón Singleton

En la solución que se propone se utiliza este patrón en las clases TcpClient y TcpServer (ver Figuras 2.14 y 2.15). En los módulos correspondientes se crea

una única instancia de estas clases y se provee una única forma de acceder a dichos objetos.

- **Facade:** Éste es un patrón de tipo estructural que facilita el manejo de un conjunto de objetos en un subsistema. Permite que a partir de una única interfaz se pueda acceder a las funcionalidades de todo el subsistema. En el HMI-Ejecución se utiliza este patrón en la clase GraphicsLoader, esta clase tiene la responsabilidad de cargar las diferentes bibliotecas de componentes gráficos y crear una estructura a partir de los archivos QML.

2.7.2. Patrones GRASP

Estos patrones, o en este caso más bien buenas prácticas, están muy ligados al diseño orientado a objetos. Generalmente en proyectos de mediana complejidad suelen utilizarse varios, pues el propio paradigma de programación orientada a objetos supone su empleo. En este trabajo se destaca el patrón **Creador**, que sugiere que las instancias deben ser creadas por la clase que tiene la información necesaria para la creación del objeto, por ejemplo en el HMI-Ejecución hay una clase llamada XmlLoader que cuenta con las funcionalidades para cargar el archivo XML que contiene toda la configuración del proyecto, a medida que se va escaneando la configuración se informa a la clase principal que la información para crear los diferentes objetos se ha leído correctamente y por tanto es hora de crear dicho objeto.

El platrón **Controlador** también es ampliamente utilizado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Precisamente el patrón arquitectónico que se utiliza en el HMI-Ejecución es el principio de separación modelo-vista (ver Figura 2.3).

2.8. Diagramas de paquete y despliegue

Los paquetes se pueden utilizar para mostrar agrupaciones lógicas de objetos. Cada paquete agrupa un conjunto cohesivo de responsabilidades. Ésta es la práctica básica de aplicar la modularidad para dar soporte a la separación de intereses [34]. En la Figura 2.17 se muestran estas agrupaciones lógicas de objetos y las dependencias entre ellas, reflejadas en los submódulos que componen el sistema, evidenciando en la misma el patrón arquitectónico seleccionado.

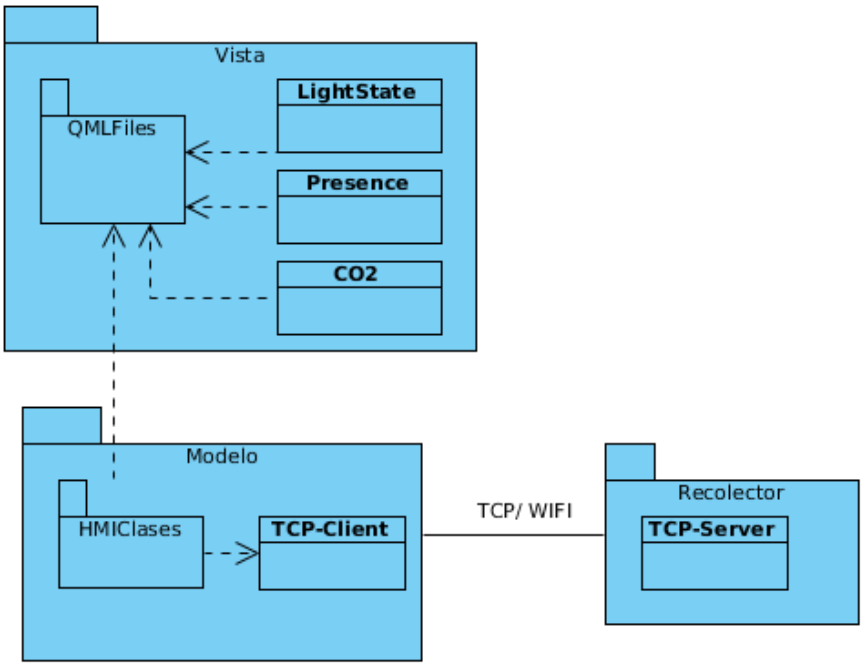

Figura 2.17: Diagrama de paquete

Un diagrama de despliegue muestra cómo se configuran las instancias de los componentes de software para su ejecución en los nodos (hardware con memoria y servicios de procesamiento) y la comunicación entre ellos [21]. En la configuración propuesta se cuenta con un ordenador en el cual se ejecuta el HMI-Editor, otro en el que es ejecutado el Recolector y un dispositivo móvil con sistema Android para la visualización del HMI-Ejecución. El HMI-Editor genera un archivo XML con la configuración realizada, el cual es utilizado por el HMI-Ejecución y el Recolector. En la Figura 2.18 se muestra la configuración de los nodos en el sistema propuesto.

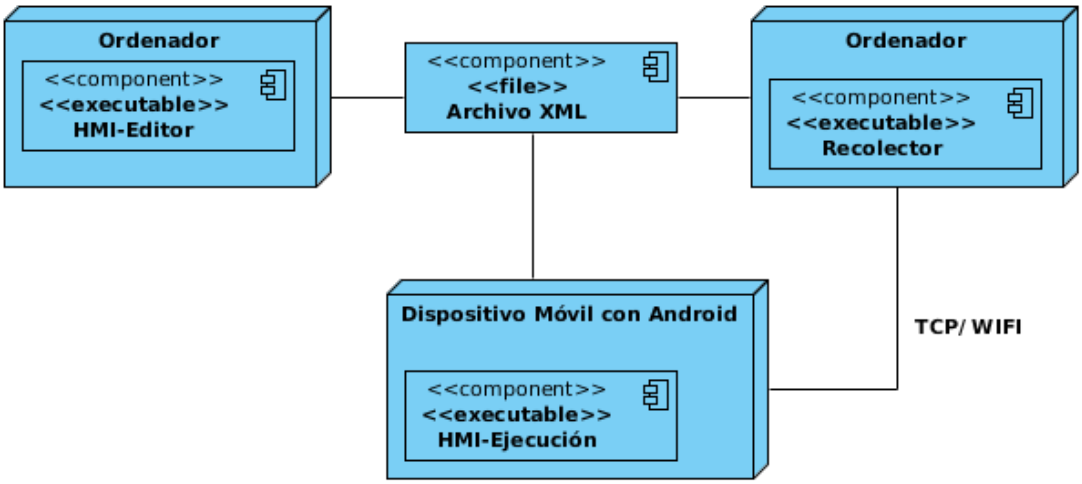

Figura 2.18: Diagrama de despliegue

2.9. Insuficiencias de la propuesta de solución

Con la realización del presente trabajo se dió solución a un conjunto de problemas identificados por los clientes del sistema. De igual forma fueron descubiertas algunas de las insuficiencias que trae consigo la presente propuesta de solución.

- **El uso del NDK:** Como se mencionó en el Capítulo 1, el NDK permite implementar aplicaciones nativas para Android, utilizando lenguajes tales como C y C++. En este trabajo, solicitado por el cliente de la solución, se utiliza el NDK para el desarrollo del HMI-Ejecución. Aunque las interfaces gráficas de Arex se desarrollan haciendo uso de QML, que es un lenguaje especialmente diseñado para el desarrollo de interfaces gráficas para entornos touchscreem; Google no recomienda el uso del NDK para este fin [24].

- **Utilización de un solo hilo para atender a los clientes TCP:** El submódulo TCP-Server, que se incluye en el Recolector, utiliza un único hilo para la atención de todos los clientes TCP. Aunque se utiliza un mecanismo asíncrono que garantiza la atención simultánea de varios clientes, el mismo basa su funcionamiento en el mecanismo de signals/slots. Teniendo en cuenta los requerimientos de Arex, donde se plantea que el máximo de variables en todo el sistema es 100 y que no debe sobrepasar los 10 clientes, el esquema diseñado

debe ser suficiente. Sin embargo si en el futuro aumentaran los requerimientos de Arex en cuanto a cantidad de variables y clientes, el mecanismo actual debería ser sustituido, por ejemplo podría pensarse la utilización de un pool de hilos.

Conclusiones parciales

En este capítulo se describe el diseño de un conjunto de componentes gráficos especializados para representar procesos en el área de la domótica. La solución propuesta incluye el módulo TCP-Server implementado en el Recolector de Arex y el TCP-Client integrado al HMI-Ejecución. Estas modificaciones permiten que el HMI-Ejecución pueda ser utilizado sobre dispositivos móviles inteligentes en el área de la domótica.

Capítulo 3

DISEÑO Y REALIZACIÓN DE PRUEBAS AL SISTEMA

Las pruebas de software son el proceso de ejecución de un programa bajo condiciones controladas, evaluando y registrando los resultados con la intención de descubrir un error, el objetivo fundamental es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo [36].

Una vez desarrollados los submódulos TCP-Client, TCP-Server y los componentes gráficos que fueron descritos anteriormente el HMI-Ejecución puede ser ejecutado sobre dispositivos móviles inteligentes. En este capítulo se describe el diseño y la ejecución de un conjunto de pruebas funcionales que se realizaron para validar el correcto funcionamiento del sistema.

3.1. Pruebas de software

Las historias de usuario son la principal fuente de información a la hora de construir las pruebas de aceptación. A una historia de usuario se le puede definir más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento. El objetivo de las mismas es verificar el cumplimiento de los requisitos, y es responsabilidad del cliente comprobar su ejecución para tomar decisiones al respecto.

Por otra parte, las pruebas de sistema tienen como objetivo ejercitar profundamente el producto, comprobándolo de forma global. Esto posibilita alcanzar una visión similar a su comportamiento en el entorno de producción. Uno de los tipos de pruebas de sistema son las pruebas de rendimiento, las cuales consisten en determinar

que los tiempos de respuesta estén dentro de los intervalos establecidos en las especificaciones del sistema.

3.2. Ambientes de pruebas

Las pruebas se realizaron sobre seis ambientes diferentes. A continuación se listan los elementos del ambiente en el que se realizaron las mismas:

- Una PC con el módulo Arex HMI-Editor.
- Una PC con el módulo Arex Recolector.
- Una PC donde se ejecuta un simulador del protocolo Modbus llamado ModSim32 (ver Figura 3.1).
- Un celular Moto E con Android 4.4.2 y 1 GB de RAM.
- Un celular Samsung Galaxy S2 modelo GT-I9100 con Android 2.3.5 y 1 GB de RAM.
- Una tableta modelo TAB264 con Android 4.0.3 y 512 MB de RAM.

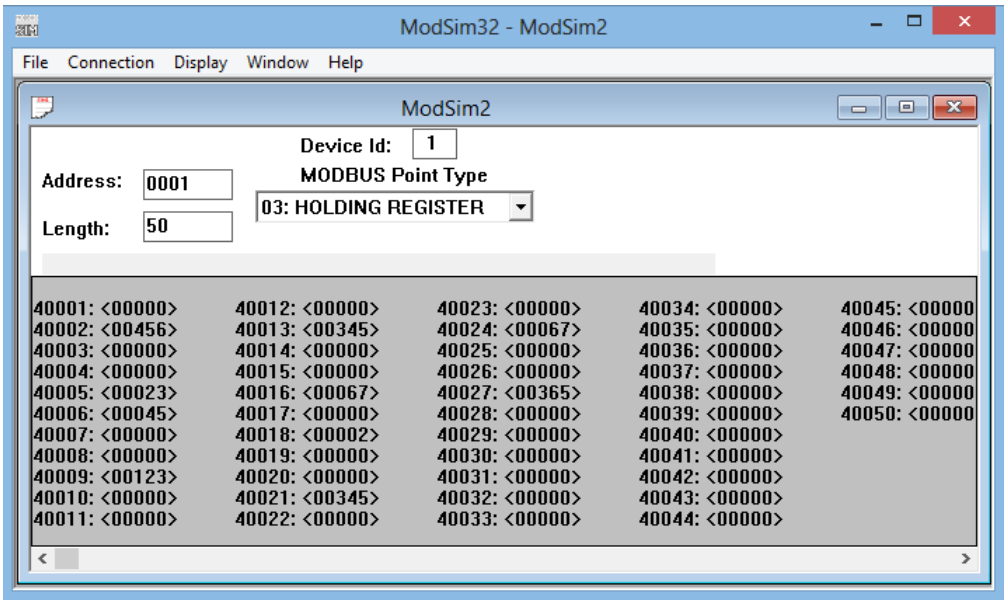

Figura 3.1: Simulador de Modbus ModSim32

3.3. Descripción de la realización de las pruebas

Para la realización de las pruebas se utilizó el módulo Arex HMI-Editor en el diseño y configuración de despliegues gráficos. Para ello se utilizaron solamente los componentes que se describen en este trabajo. El HMI-Editor utiliza un archivo XML para guardar la configuración de los diseños que se realicen.

Seguidamente se ejecuta el módulo Arex Recolector y el simulador del protocolo modbus ModbusSim (ambos pueden ejecutarse sobre una misma PC o separados) con la configuración necesaria para que puedan comunicarse. A la PC donde se encuentra el Recolector se debe poder acceder desde una red Wifi pues es la interfaz de comunicación que utiliza el HMI-Ejecución cuando se ejecuta sobre dispositivos móviles inteligentes.

Finalmente se ejecuta el HMI-Ejecución previamente instalado en los dispositivos móviles que se listaron anteriormente. La aplicación debe mostrar la interfaz de Arex con las vistas que fueron diseñadas. Para observar el funcionamiento del componente LightState se puede accionar sobre el mismo y verificar el cambio en el ModbusSim. El resto de los componentes no permiten el envío de comandos, lo que significa que para verificar su funcionamiento se deben modificar las variables correspondientes en el ModbusSim.

En los acápites siguientes se describen un conjunto de pruebas de aceptación y de rendimiento.

3.4. Pruebas de aceptación

Caso de prueba: 1
Número historia: 1, 2, 3
Nombre: Visualización de los ocho componentes gráficos implementados.
Descripción: Se diseñaron dos despliegues (sala, comedor), en el primero se colocaron los componentes gráficos (LightIntensity, Humidity, CO2 y Temperature) que fueron asociados a variables analógicas. En el segundo despliegue se colocaron los componentes (LightState, WindState, DoorState y Presence) que fueron asociados a variables discretas.
Condiciones de ejecución: Debe existir una variable asociada a cada componente.
Entradas/Pasos de ejecución: Ver 3.3.
Resultado esperado: Los componentes se muestran correctamente y con el valor correspondiente según la configuración del ModSim32.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.1: Caso de prueba No.1/ Historia de usuario No. 1, 2, 3

Caso de prueba: 2
Número historia: 1, 2, 3
Nombre: Animaciones y envío de comandos.
Descripción: Se diseñó un despliegue (sala), en el que se colocaron los componentes gráficos que tienen animaciones (LightState y Presence), ambos componentes fueron asociados a una misma variable discreta.
Condiciones de ejecución: Debe existir una variable discreta configurada.
Entradas/Pasos de ejecución: Ver 3.3. Se debe enviar un comando de escritura al presionar el componente LightState.
Resultado esperado: Los componentes se muestran correctamente y con el mismo valor correspondiente según la configuración del ModSim32. Al presionar sobre el componente LightState se envía un comando de escritura y los componentes cambian de estado, cuando el estado es 1 se activan las animaciones en ambos componentes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.2: Caso de prueba No.2/ Historia de usuario No. 1, 2, 3

Caso de prueba: 3
Número historia: 1, 2, 3
Nombre: Falla de comunicación entre el HMI-Ejecución y el Recolector.
Descripción: Se utilizan los pasos descritos en 3.3 y se desactiva la conección wifi en el móvil durante 1 minuto. Al cabo de ese tiempo se vuelve a activar.
Condiciones de ejecución: Se debe desactivar la conección wifi en el móvil durante 1 minuto.
Entradas/Pasos de ejecución: Ver 3.3. Desactivar la conección wifi en el móvil por 1 minuto.
Resultado esperado: Al desactivar la conección wifi y conectarla nuevamente los componentes se muestran con el valor correspondiente según la configuración del ModbusSim.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.3: Caso de prueba No.3/ Historia de usuario No. 1, 2, 3

3.5. Pruebas de rendimiento

Caso de prueba: 4
Número historia: 1, 2, 3
Nombre: Visualización de 100 componentes gráficos y dos variables.
Descripción: Se diseñaron 10 despliegues en los que se distribuyen 50 componentes de estado on/off y 50 componentes que pueden ser asociados a variables analógicas. Los primeros fueron asociados a una misma variable discreta y los segundos a una misma variable analógica. Esta prueba se evaluó durante 24 horas.
Condiciones de ejecución: Debe existir dos variables configuradas, una discreta y otra analógica.
Resultado esperado: Durante el tiempo de prueba en cada uno de los despliegues los componentes mantienen los valores correspondientes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.4: Caso de prueba No.4/ Historia de usuario No. 1, 2, 3

Caso de prueba: 5
Número historia: 1, 2, 3
Nombre: Visualización de 100 componentes asociados a 100 variables, que es el máximo definido en los requerimientos de Arex.
Descripción: Se diseñaron 10 despliegues en los que se distribuyen 50 componentes de estado on/off y 50 componentes que pueden ser asociados a variables analógicas. Los primeros fueron asociados a diferentes variables discretas y los segundos a diferentes variables analógicas. Esta prueba se evaluó durante 24 horas.
Condiciones de ejecución: Debe existir 100 variables configuradas, 50 discretas y 50 analógicas.
Resultado esperado: Durante el tiempo de prueba en cada uno de los despliegues los componentes mantienen los valores correspondientes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.5: Caso de prueba No.5/ Historia de usuario No. 1, 2, 3

Caso de prueba: 6
Número historia: 1, 2, 3
Nombre: Utilización de dos dispositivos móviles con el HMI-Ejecución y la misma configuración.
Descripción: Se estableció la comunicación con el recolector mediante el HMI-Ejecución instalado en dos dispositivos móviles.
Condiciones de ejecución: Debe instalarse en los dispositivos móviles el HMI-Ejecución.
Resultado esperado: Durante el tiempo de prueba en cada uno de los dispositivos los componentes mantienen los valores correspondientes.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 3.6: Caso de prueba No.6/ Historia de usuario No. 1, 2, 3

Conclusiones parciales

Las pruebas de aceptación y de rendimiento realizadas en este capítulo, permitieron comprobar el correcto funcionamiento y visualización de los componentes gráficos diseñados. Se probó además que el nuevo HMI-Ejecución puede ser ejecutado satisfactoriamente sobre distintos dispositivos con sistema operativo Android, con versiones y prestaciones diferentes.

CONCLUSIONES

Al término de la presente investigación se arriba a las siguientes conclusiones:

- Los nuevos componentes gráficos desarrollados y la implementación de los submódulos TCP-Server y TCP-Client, permiten que el sistema Arex pueda ser utilizado en el área de la domótica, y que el HMI-Ejecución pueda ser visualizado sobre dispositivos móviles inteligentes con Android.
- Las pruebas realizadas muestran que la nueva versión del HMI-Ejecución funciona correctamente sobre dispositivos móviles de diferentes fabricantes y con diferentes prestaciones.

RECOMENDACIONES

Al concluir la presente investigación se proponen, a manera de recomendación, una serie de tareas para ampliar y dar continuidad al trabajo realizado:

- Diseño de nuevos componentes gráficos tanto para la domótica como para otras áreas de aplicación.
- Desarrollo de un módulo para garantizar la seguridad del sistema.
- Especialización de los diseños y las vistas del HMI-Ejecución para ser visualizado en diferentes entornos (escritorio, teléfonos inteligentes, tabletas electrónicas, entre otros).

REFERENCIAS BIBLIOGRÁFICAS

[1] A. A. de Energía Renovable, "Energía estratégica," 2015 (Consultado Junio 8, 2015). Disponible en: http://www.energiaestrategica.com/.

[2] L. Electronics, "Loxone smart home," 2014 (Consultado Enero 18, 2015). Disponible en: http://www.loxone.com/enuk/products/software/apps.html.

[3] NIESSEN and A. Ltd., "Planner mobile," 2013 (Consultado Enero 14, 2015). Disponible en: https://play.google.com/store/apps/details?id=com.niessen.planner.

[4] I. Belkin International, "Wemo light switch," 2014 (Consultado Enero 10, 2015). Disponible en: http://www.belkin.com/us/F7C030-Belkin/p/P-F7C030/.

[5] W. Automation, "Elkdroid security and automation," 2014 (Consultado Enero 16, 2015). Disponible en: http://www.appszoom.com/android_applications/tools/elkdroid-security-automation_jwkr.html.

[6] SPVSOFT, "Droidseer x10 home automation," 2014 (Consultado Enero 11, 2015). Disponible en: http://www.appszoom.com/android_applications/communication/droidseer-x10-home-automation_bias.html.

[7] P. Reyes Aguilar, "Análisis de los sistemas de medición (msa)," 2007.

[8] L. M. Fernández Sánchez, "Sistemas de medición," 2009.

[9] L. M. Fernández Sánchez, "Sistemas de adquisición de datos," 2009.

[10] A. Cedeño Pozo, K. Delgado Alón, A. Jiménez López, A. Moreno Limonte, and A. Denis Acosta, "Arex, una solución minimalista para la supervisión y control de procesos," *Memorias de la Jornada Científica del ICIMAF*, 2013.

[11] F. Urdiales Ponce and A. Machado, "La domótica y su contribución en el uso racional de recursos energéticos. diseño de solución con tecnologías libres y de bajo costo (caso universidad ecotec)," 2015.

[12] J. Chaparro, "Domótica: La mutación de la vivienda," 2003.

[13] V. Aguirre Muñoz, "Prototipo de sistema de control domótico por medio de dispositivos android, utilizando processing," 2013.

[14] J. Alcalá Correa, "Software accesible para control de entorno mediante dispositivos móviles," 2010.

[15] G. O. Pedrozo Petrazzini, "Sistemas operativos en dispositivos móviles," 2012.

[16] A. Inc., "qué es ios?," 2015 (Consultado Marzo 9, 2015). Disponible en: https://www.apple.com/es/ios/what-is/.

[17] Microsoft, "Windows phone 8.1 chassis requirements specification," 2015 (Consultado Marzo 9, 2015). Disponible en: https://dev.windowsphone.com/en-US/OEM/docs/Phone_Bring-Up/2__Components.

[18] L. GeekWire, "Live from microsofts big windows event in san francisco," 2015 (Consultado Marzo 9, 2015). Disponible en: http://www.geekwire.com/2014/live-microsofts-enterprise-focused-windows-unveiling/.

[19] S. Panth and M. Jivani, "Home automation system (has) using android for mobile phone," *International Journal of Electronics and Computer Science Engineering*, vol. 3, no. 1, 2013.

[20] C. B. Larramendi Ferrás, "Propuesta de arquitectura para desarrollo de aplicaciones compuestas para dispositivos móviles con android," 2012.

[21] K. Delgado Alón and A. Jiménez López, "Módulo hmi para una tarjeta basada en microcontroladores," 2012.

[22] D. Florentino, "Historia del computador y evolución, arquitectura, lenguajes de programación y algoritmos," 2007 (Consultado Enero 6, 2015). Disponible en: http://nectacellcomputer.jimdo.com.

[23] B. Stroustrup, *The Design and Evolution of C++.* 1994.

[24] A. Developers, "Android ndk," 2015 (Consultado Junio 8, 2015). Disponible en: https://developer.android.com/tools/sdk/ndk/index.html.

[25] M. Natterer and S. Neumann, "Gun image manipulation program," 2015 (Consultado Mayo 27, 2015). Disponible en: http://www.gimp.org/.

[26] M. C. Gasca Mantilla, L. L. Camargo Ariza, and B. Medina Delgado, "Metodología para el desarrollo de aplicaciones móviles," *Tecnura*, vol. 18, no. 40, pp. 20–35, 2014.

[27] J. L. Cordero, "Metodologías ágiles, proceso unificado ágil (aup)," 2015.

[28] "Metodología de desarrollo para la actividad productiva de la uci," 2015.

[29] I. Red Hat, "D-bus tutorial," 2015 (Consultado Marzo 16, 2015). Disponible en: http://dbus.freedesktop.org/doc/dbus-tutorial.html.

[30] D-Bus, "D-bus documentation," 2015 (Consultado Junio 8, 2015). Disponible en: http://www.freedesktop.org/wiki/Software/dbus/.

[31] C. I. Kioskea, "Protocolo tcp," 2015 (Consultado Marzo 16, 2015). Disponible en: http://es.kioskea.net/contents/281-protocolo-tcp.

[32] IEEE1471-2000, "Recommended practice for architecture description of software-intensive systems," 2000.

[33] R. S. Pressman, *Ingeniería de software, un enfoque práctico.* McGraw-Hill, 2005.

[34] C. Larman, *UML y Patrones. s.l.* Prentice Hall, 2002.

[35] R. H. Bryant, "Lumens, illuminance, foot-candles and bright shiny beads," *The LED Light*, 2010.

[36] R. S. Pressman, *Ingeniería de software, un enfoque práctico.* McGraw-Hill, 2002.