UNIVERSIDAD NACIONAL DE CATAMARCA
FACULTAD DE TECNOLOGÍA Y
CIENCIAS APLICADAS

INGENIERÍA EN INFORMÁTICA

TRABAJO FINAL

SISTEMA EMBEBIDO DE CONTROL DE ACCESO Y REGISTRO AUTOMÁTICO DE ASISTENCIA CON TECNOLOGÍA NFC PARA LA FACULTAD DE TECNOLOGÍA Y CIENCIAS APLICADAS

AUTORA: BARRIONUEVO, NOELIA GISELE

DIRECTOR: MS. ING. ARANDA, MARCOS DARÍO

Catamarca Argentina, Mayo 2020

AGRADECIMIENTOS

A mis padres, Miriam y Oscar por ser el cimiento principal de mi vida, por su constancia en momentos difíciles, por depositar su confianza en mí y alentarme cuando todo estaba cuesta arriba, por ser fuente de inspiración y esfuerzo para tener visión hacia un futuro mejor. Por haberme forjado como la persona que soy y jamás haberme abandonado con su apoyo incondicional, para lograr mí meta.

A mi Nona Hilda, gracias por tu entereza, por enseñarme el camino de la vida, gracias por tus consejos, por el amor que me has dado y por tu apoyo incondicional en mi vida. Gracias por llevarme en tus oraciones porque estoy segura que siempre lo haces.

A mis familiares, amigos y compañeros por su constante apoyo y aliento para seguir adelante y alcanzar este ansiado objetivo. Gracias por la ayuda, por compartir conocimientos, por las bromas y las risas. De igual manera, a todos quienes fueron aportando con su ayuda a lo largo de esta carrera universitaria.

A mi director de tesis, Ms. Ing. Magister Marcos Darío Aranda, por su apoyo y acertado asesoramiento. Por guiarme en el proceso con entusiasmo y compromiso, siempre dispuesto a colaborar.

A la Directora del Departamento de Informática Lic. Valeria Poliche, a la Secretaria de asuntos Académicos Esp. Lic. Natalia Fernández, a la Directora del I.D.I. Esp. Lic. Marisa I. Korzeniewski, a la Directora de Asuntos Académicos Lic. Miriam Cisternas, y a todos los docentes de la Facultad de Tecnología y Cs. Aplicadas que me formaron como futura profesional.

Dedicatoria

El presente proyecto lo dedico a mi madre, por ser la insignia de fortaleza en mi familia, por siempre estar a mi lado para recordarme que puedo conseguir todo aquello que me proponga en la vida.

A mi padre, por ser símbolo de responsabilidad y esmero durante toda su vida para ser el sostén del hogar. Esta alegría es fruto, esfuerzo, apoyo y trabajo de ustedes también, sin su ayuda todo esto no hubiese sido posible. Su aporte con cada consejo y los valores inculcados fueron esenciales para mi formación personal y académica.

Fue un largo camino con muchas pruebas, derrotas y victorias, pero llegamos a la meta. Les prometo que el esfuerzo no fue en vano y que días mejores están por venir. Hoy puedo decir que no fue fácil, pero al fin lo logramos. De todo corazón, mil gracias por acompañarme en este viaje. Esto recién empieza.

INDICE

Resumen

El siguiente trabajo final de tesis presenta un prototipo llamado ***SCARAA*** (Sistema de Control de Acceso y Registro Automático de Asistencia). Este proyecto surge de la necesidad de ofrecer un control de asistencia automatizado para el auditorio de la FTyCA (Facultad de Tecnología y Ciencias Aplicadas), el cual es un dispositivo de identificación confiable. Dentro del marco de renovación se incluyeron algunas novedades como es el uso de la tecnología NFC (Near Field Communication - Comunicación por Campo Cercano), tarjetas inteligentes NFC, el desarrollo de una APP móvil y el manejo de una pantalla HMI (Humman Machine Interface - Interfaz Hombre Maquina).

En esta producción académica se diseñó y desarrollo un sistema informático que permita, a través de un sistema embebido registrar la entrada y salida de cada concurrente a los cursos que se dictan. El modo de funcionamiento del sistema permite una preinscripción online que se puede llevar a cabo a través de una aplicación móvil y corporativa, quedando los datos registrados en un servidor de base de datos.

Es por ello que este trabajo incluye, para los concurrentes a cursos dictados en la FTyCA, este novedoso dispositivo con sistematización informática y electrónica que cuenta con un control de asistencia fiable, seguro y a su vez, sencillo de manejar por los usuarios. De esta manera, el sistema de control de asistencia para el auditorio asegura la eficacia mediante el uso de la tecnología NFC, concretamente la comunicación se realiza a través de un celular con dicha tecnología y un módulo que cuente con un chip NFC Pn532 conectado a la placa Intel Galileo.

Palabras clave: control, embebido, NFC, auditorio.

CAPITULO I

1 Introducción

La velocidad con que avanza la tecnología en cuanto a la automatización de tareas, alivia el trabajo del hombre y ahorra una importante cantidad de tiempo ofreciendo mejores respuestas a los problemas de la vida cotidiana. Este desarrollo de la tecnología que atiende las demandas del mercado actual ha migrado del sistema mecánico con personal especializado, al sistema automatizado con diferentes tipos de tecnologías y dispositivos adecuados para el control.

Se puede afirmar que cuando pensamos en el control de acceso, en un inmueble de hace tiempo, nos imaginamos un portero o alguien que anotaba sus datos y le permitía el acceso. Con el devenir de los años, esta identificación se hace a través de tarjetas inteligentes, muchas de las cuales usamos cotidianamente. En nuestros días ya con el advenimiento de las nuevas tecnologías, se cuenta con la autenticación biométrica, y la facial, entre otras herramientas de control.

Ciertamente, el acceso a los sistemas de control, se ha adaptado a las diferentes necesidades de las generaciones que requieren cada vez más soluciones fáciles, ágiles, seguras y flexibles. En la actualidad existe una gran variedad de tecnologías que permiten el control de acceso entre las cuales se encuentran las tarjetas magnéticas, las tarjetas de proximidad RFID (Radio Frequency Identification – Identificación por Radiofrecuencia), los lectores biométricos y la tecnología NFC.

En este trabajo de Sistema de Control de Acceso y Registro Automático de Asistencia, la unidad central de procesamiento es una placa Intel® Galileo Gen 1, la tecnología NFC, una aplicación móvil, una pantalla táctil HMI y la gestión de los accesos en una base de datos que se encuentra en un servidor. Con este sistema de control de acceso desarrollado con la Intel® Galileo Gen 1 se pretende demostrar que es un sistema fiable, eficiente y moderno pudiendo ser muy satisfactorio para controlar la entrada y salida de los concurrentes en el auditorio de la FTyCA.

De este modo se busca dar respuesta a la ausencia de personal destinado a realizar tareas de registro y control automatizado de asistencia, para la entrada y la salida al auditorio de la FTyCA cuando sea necesario, por ejemplo actividades académicas de posgrado, cursos,

jornadas, etc., logrando por medio de un dispositivo automático embebido con un monitoreo confiable, mejorado y optimizado.

La tecnología hoy nos brinda una serie de dispositivos móviles, entre otros (celulares, tablets, laptops, etc.) instrumentos de uso cotidiano en la vida de cualquier persona, que son capaces de reproducir software o programas. Debido a ello es que la aplicación que se desarrolló puede ser montada en un dispositivo móvil que tenga integrada comunicación NFC.

El presente trabajo final se dividió en 8 capítulos, sobre los cuales se hace una breve reseña a continuación:

- Capítulo 1 “Introducción”: Se presenta el trabajo propuesto, el problema planteado, la justificación, los objetivos y una descripción del sistema en general.
- Capítulo 2 “Marco Teórico”: Se introducen los sustentos teóricos sobre las tecnologías y herramientas utilizadas en el desarrollo.
- Capítulo 3 “Herramientas de Análisis y Tecnológicas”: Se realiza un análisis del sistema para comprender su funcionamiento y su estructura interna.
- Capítulo 4 “Desarrollo de la aplicación móvil”: Se describen los requisitos funcionales y no funcionales y el proceso de desarrollo de la aplicación móvil utilizando la metodología de desarrollo de software XP.
- Capítulo 5 “Diseño del prototipo”: Se describe el proceso de desarrollo del prototipo del dispositivo SCARAA, sus dos modos de operación y el desarrollo del firmware que se ejecutará sobre dicha plataforma.
- Capítulo 6 “Pruebas”: Se describen las pruebas realizadas para probar su correcto funcionamiento.
- Capítulo 7 “Trabajos futuros”: Se proponen mejoras observadas a lo largo del desarrollo principal.
- Capítulo 8 “Conclusiones”: Se describen las conclusiones del presente trabajo, poniendo en manifiesto el cumplimiento de los objetivos iniciales.

1.1 Planteamiento Del Problema

Actualmente el proceso de control de asistencia en el auditorio de la FTyCA se realiza de forma presencial y manual, no cuenta con un sistema automatizado que permita economizar RRHH, ahorrar tiempo y agilizar la tarea, haciendo además que este trámite sea poco ágil y desorganizado. De esta manera la comunicación se ve afectada y en muchos casos no es la correcta. Por lo tanto se desea incorporar nuevas tecnologías de control de acceso, en este caso particular se desea utilizar el NFC como método de solución innovadora, ya que provee un alto grado de seguridad y que implementa de manera conjunta con la placa Intel® Galileo Gen 1 un sistema que recopile y procese la información.

Este sistema de control de acceso propuesto en la investigación surge a partir de indagar la manera de solucionar esta problemática y realizar un sistema con tecnología de punta NFC escalable, confiable, seguro y con interfaces amigables. Este tipo de tecnología adquirió en estos últimos años una presencia importante en nuevos proyectos, y muchos de ellos de grandes dimensiones. Además entre los alcances de esta tecnología es de destacar su capacidad para interactuar con otros dispositivos sea de forma intuitiva, simple y segura; promete ser útil y atractiva al usuario, contando con la ventaja de la interoperabilidad con otras tecnologías inalámbricas existentes.

Es innegable y a la vez un signo de estos tiempos que los dispositivos móviles se han convertido en elementos imprescindibles en la vida diaria. Como muestra, se menciona el crecimiento de teléfonos inteligentes, el acceso a internet a las distintas regiones y los avances tecnológicos que nos ofrece el mercado; esto permite diseñar un sistema para solucionar los problemas de control de asistencia que son una constante debilidad para reflejar el seguimiento de los participantes al auditorio en sus diversas ofertas. Esta situación se puede solucionar integrando conceptos de la informática en conjunto con la electrónica. De esta forma, los asistentes acceden a la preinscripción por medio de una aplicación previamente instalada en su dispositivo móvil, validando su asistencia al curso a través de su celular que cuente con tecnología NFC y el dispositivo embebido ubicado en la puerta de acceso del auditorio.

Está claro que el uso de nuevas tecnologías, como son las placas de desarrollo basadas en procesadores orientados a conectar dispositivos, por ejemplo la placa Intel® Galileo Gen 1 y también las tecnologías NFC o la aplicación móvil en Android para ejecutar la implementación de un sistema integral de hardware y software pueden solucionar el problema antes mencionado.

1.2 Justificación

Es necesario que el auditorio de la FTyCA, tenga un sistema de control de asistencia que facilite este trabajo, ayudando que se cumplan las tareas de manera más eficiente. El proyecto propuesto tiene como objetivo reducir el tiempo empleado en el proceso de control de asistencia a eventos; esto beneficiara por una parte a los usuarios, al ver reflejado en menor tiempo el registro de su asistencia. Propone un proceso automatizado, el cual beneficiara al personal encargado del evento realizar su trabajo de forma eficiente. Además, este prototipo mejora el sistema de promoción de los cursos, esto beneficia a todos los departamentos académicos pertenecientes a la FTyCA que organicen un evento.

1.3 Objetivos:

1.3.1 Objetivo General

Desarrollar un sistema de registro para optimizar el control de asistencia automatizado en el auditorio de la FTyCA, mediante el uso de sistemas embebidos y la tecnológica NFC.

1.3.2 Objetivos Específicos

- Investigar la situación del control de asistencia vigente en el auditorio de la FTyCA, para plantear una nueva funcionalidad.
- Estudiar los conceptos implicados con la placa Intel® Galileo Gen 1 y la tecnología NFC.
- Desarrollar un sistema embebido utilizando la placa Intel® Galileo Gen 1 manejando tecnología NFC.
- Demostrar las virtudes de la Metodología XP para el desarrollo de la aplicación.
- Desarrollar una aplicación móvil que muestre la información necesaria para la preinscripción del correspondiente curso, considerando las siguientes funciones:
    - Login de usuario
    - Nuevo registro de usuario
    - Cursos disponibles
    - Confirmar Inscripción a los eventos
    - Informar sobre el cupo disponible
- Establecer la comunicación entre la app móvil y una base de datos que se adapte al sistema y la conexión con el dispositivo.

- Establecer la comunicación de SCARAA, mediante una interfaz Ethernet que le permite conectarse a una red LAN y utilizar un servicio web.

1.4 Descripción

El desarrollo consta de un sistema embebido sobre la plataforma de Intel® Galileo Gen 1, un módulo NFC/RFID el cual contiene un chip Pn532 y una pantalla HMI fabricada por Nextion Itead. Todos los componentes antes mencionados se conectan para formar SCARAA, permitiendo la inscripción en el lugar donde se dicta el curso. Una aplicación móvil desarrollada en Android Studio permite la carga de los datos de la inscripción previa al curso y con la posibilidad de conocer la disponibilidad del mismo.

La comunicación de SCARAA, se realiza mediante una interfaz Ethernet que le permite conectarse a una red LAN y utilizar un servicio web para la consulta de los datos.
(Ver Figura 1).

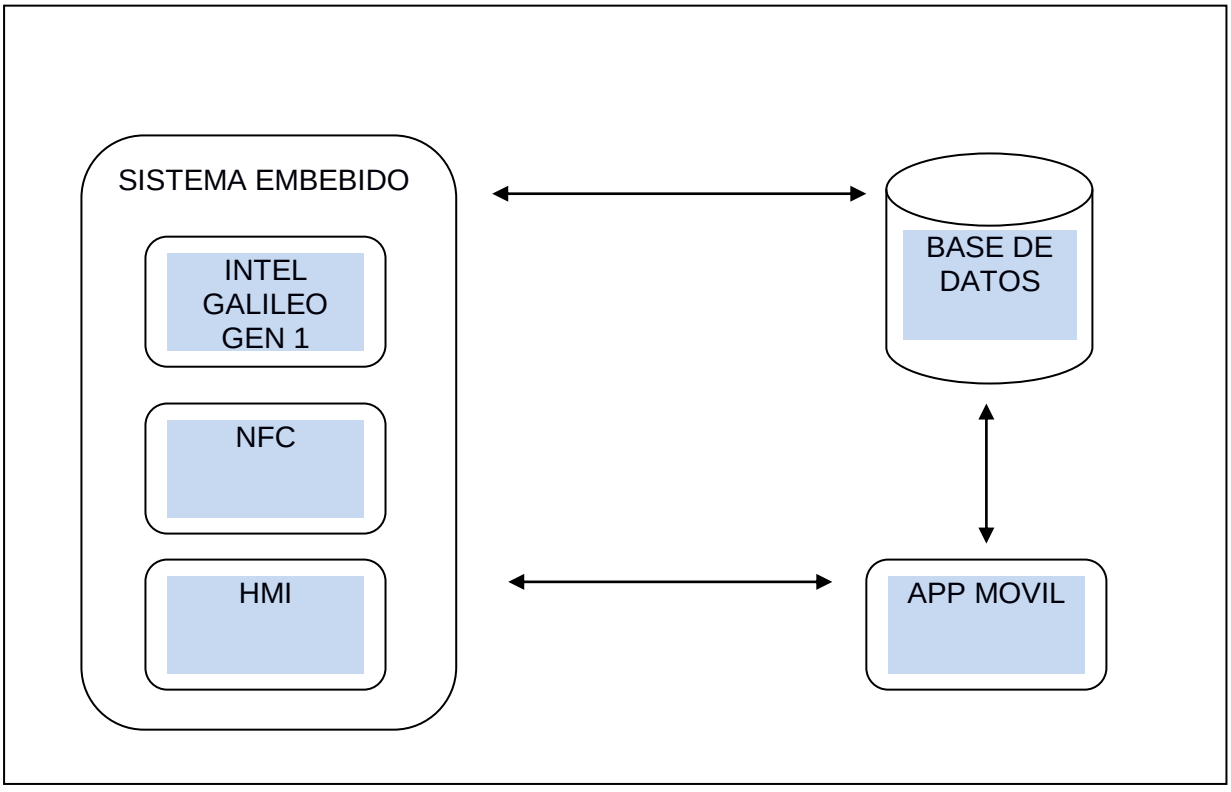


Figura 1: Componentes del proyecto SCARAA

CAPITULO II

2 Marco Teórico

2.1 Digitalizar el control de asistencia

Tradicionalmente en todo tipo de centros educativos se ha llevado el control de asistencia de alumnos de manera manual y todavía son muchas las instituciones en las que se sigue haciendo de esta manera porque no han incorporado aún sistemas más modernos para gestionar.

Existen algunas desventajas de seguir usando el papel y lápiz para el control de asistencia. Primeramente los datos obtenidos se pueden disipar fácilmente**,** generalmente no se realiza una copia de la información para el caso que se pierda y los alumnos no disponen de la información en el momento para corroborar algún desacuerdo de ella. Por suerte en la actualidad han surgido diferentes soluciones con la intención de mejorar esta tarea imprescindible en cualquier institución educativa y/o laboral. Como por ejemplo:

Apps de control de asistencia: permiten llevar un seguimiento no sólo de la asistencia del alumno, sino también de sus notas y evaluación personal, se trata de aplicaciones disponibles para computadoras personales, tablets o teléfonos móviles. También permiten exportar los datos en una plantilla de Excel. Por ejemplo la app “Pasalista” (disponible en Play Store).

Apps de cuaderno del profesor: una herramienta de la tienda oficial de aplicaciones para androide Play Store, disponible para dispositivos android que sirve para tener siempre una completa ficha de los alumnos incluido un cuaderno de tutoría por cada alumno, es más completa que la anterior ya que además de permitir controlar la asistencia, incluyen todas las funciones normales del cuaderno del profesor (gestionar el calendario escolar, gestionar las calificaciones, control de asistencia, evaluación del seguimiento del programa, etc.).

Softwares de gestión escolar integrales: “Gestión Escolar” es un programa diseñado por Riversoft para gestionar cualquier tipo de centro educativo. Es completamente gratuito, sin límite de alumnos o de tiempo. Se adapta a cualquier etapa educativa como Infantil, ciclos formativos o bachillerato. Entre todas sus funciones también está incluido el control de la asistencia. La seguridad de los datos es unas de las ventajas de controlar la asistencia de manera virtual, como así también respetando la ley de protección de datos.

2.2 Tipos de control de asistencia

Un sistema de control de asistencia o de horario es aquél con el que se supervisan horas de entrada y de salida de los asistentes con lo que se verifica el número de horas cumplidas.

Al momento de elegir un sistema de control de asistencia, hay que tener en cuenta todas las posibilidades disponibles hoy día en el mercado. Los tipos de Sistemas de control de asistencia son:

-Tarjetas de proximidad:

Es una tarjeta plástica que utiliza la tecnología RFID y lleva en su núcleo un circuito integrado y una antena de comunicación. Cada tarjeta cuenta con un número de serie que es leído por frecuencia de radio, por lo general la más usada es la de 125kHz. Las tarjetas de proximidad son capaces de transmitir un código único, de más de 1 billón de combinaciones, y la lectura se produce sin contacto físico, por lo que el deterioro es mínimo.

-Sistemas de control biométricos:

Son sistemas muy sofisticados que eliminan la suplantación de identificación de los empleados. Son el resultado de la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos de una persona para verificar su identidad. Como ejemplo de características físicas serían las huellas dactilares, los patrones faciales o la geometría de la palma de la mano.

- ***Identificación por huellas dactilares:*** El sistema de identificación automatizada de huellas dactilares. Funciona por reconocimiento de la huella digital de alguno de los dedos de la mano.
- ***Identificación por biometría facial:*** Su rendimiento puede verse afectado por circunstancias ajenas a las personas porque es una tecnología que requiere de unas condiciones muy concretas para su verificación, sobre todo de luz.
- ***Biometría de perfil de mano:*** Funciona por reconocimiento de la morfología de la mano y es muy precisa en condiciones adversas, muy usadas para control de asistencia en fábricas, minas o campo.

2.3 Sistemas Embebidos

2.3.1 Definición

“SE” (Sistema embebido) es el nombre genérico que reciben los equipos electrónicos que realizan el procesamiento de datos e información, pero que a diferencia de una computadora personal, están diseñados para satisfacer una función específica, como en el caso de un reloj, un teléfono celular, el sistema de control de un automóvil, de un satélite o de una planta nuclear. Es un sistema electrónico que esta contenido ("embebido") dentro de un equipo completo que incluye, por ejemplo, partes mecánicas y electromecánicas. [1]

En términos generales un SE es cualquier dispositivo que incluye una computadora programable, pero no es en sí mismo un sistema de computación integrado, aunque las PC a menudo se usan para construir un sistema de computación incorporado. Pero una máquina de fax o un reloj construido a partir de un microprocesador es un sistema informático integrado. [2]

Un SE también se puede definir como una combinación de hardware y firmware (software) diseñado para realizar una función específica (Ver Figura 2). En algunos casos estos sistemas embebidos son componentes dentro de un sistema de mayor escala. [3]

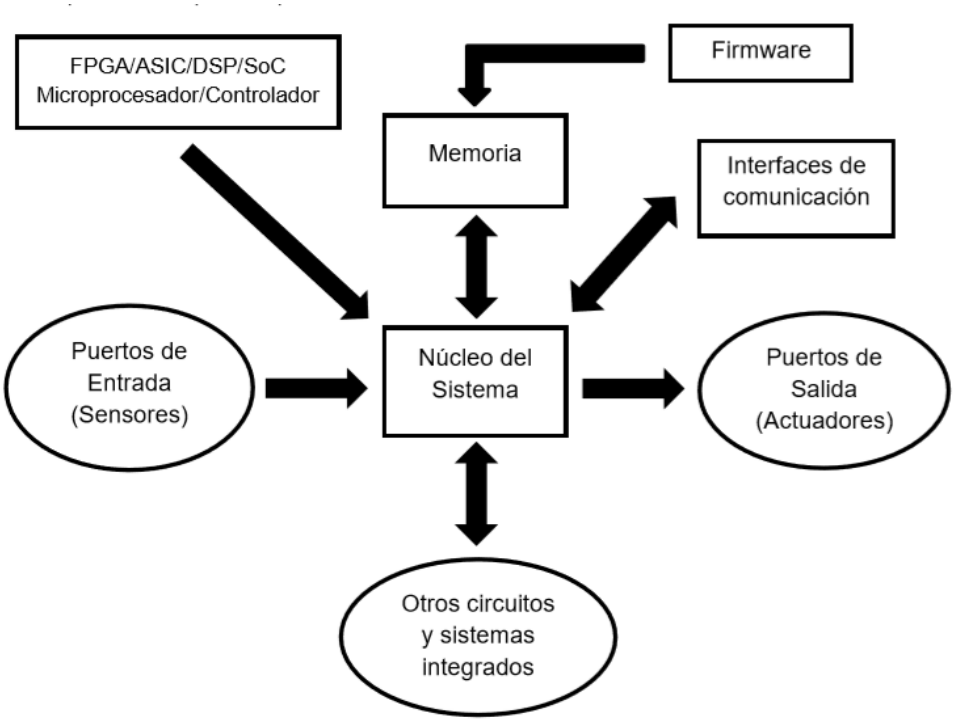


Figura 2: Elementos de un Sistema Embebido [3]

2.3.2 Hardware de Sistemas Embebidos

El componente fundamental en este tipo de sistemas es su unidad programable o procesador embebido, que porta el firmware del SE. Esta unidad programable se puede implementar con distintos dispositivos según la capacidad de procesamiento que se requiera. Los más usados son: Microcontroladores y Microprocesadores.

Microcontrolador: La alternativa más usual es utilizar microcontroladores de 4 a 32 bits que incluyen memoria principal y secundaria además de periféricos en un único chip. La memoria principal se utiliza para albergar las variables del programa mientras que la memoria secundaria (típicamente FLASH) contiene el firmware. Para microcontroladores de hasta 16 bits suele implementarse programación básica, es decir que se programa la aplicación de manera directa sin necesidad de utilizar un RTOS (Sistema Operativo de Tiempo Real). Cuando se utilizan microcontroladores de 32 bits, dependiendo de la aplicación, suelen utilizar un RTOS. El lenguaje de programación que se utiliza normalmente es “C” y se complementa con ensamblador cuando se necesita un control estricto del tiempo de ejecución para alguna tarea.

Microprocesador: se han desarrollado fundamentalmente pensando en aplicaciones que requieren una gran capacidad de cálculo, manejo de gran cantidad de memoria y gran velocidad de procesamiento. A diferencia del microcontrolador que internamente incluye los buses, el banco de memoria, el reloj y temporizadores, el microprocesador necesita soporte externo para poder funcionar. Así, es necesario que en una placa madre se rutean los buses de datos, dirección y control para poder acceder a los bancos de memoria tanto RAM como ROM y los periféricos, que también deben incluirse en la placa madre. Teniendo en claro esta diferencia, es común hacer referencia a la unidad de procesamiento del SE como el procesador embebido indistintamente ya sea que se haya utilizado un microcontrolador o un microprocesador. Típicamente, el uso de una u otra variante vendrá de la mano de la velocidad de procesamiento necesaria y la necesidad de uso de un sistema operativo.

2.3.3 Firmware

El firmware es el conjunto de instrucciones de un programa que se encuentra almacenado en la memoria del sistema embebido. Estas instrucciones fijan la lógica primaria que ejerce el control de los circuitos que conforman el SE. El firmware forma parte del hardware ya que se encuentra integrado a la electrónica, pero también está considerado como parte del software al estar desarrollado bajo algún lenguaje de programación. Podría decirse que el

firmware funciona como el nexo entre las instrucciones que llegan al dispositivo desde el exterior y sus diversas partes electrónicas. [4]

Entre sus funciones básicas se destacan:

- Proporcionar al SE las rutinas de funcionamiento y respuesta a las peticiones de usuario.
- Controlar y gestionar el arranque del SE

El software embebido puede ser desarrollado siguiendo uno de los siguientes métodos:

- Escribir el programa en un lenguaje de programación de alto nivel como C/C++ usando un entorno de desarrollo integrado (IDE).
- Escribir el programa en lenguaje Ensamblador usando las instrucciones soportadas por el procesador/controlador.

El conjunto de instrucciones para cada familia de procesadores/controladores es diferente y el programa escrito en cualquiera de los métodos mencionados anteriormente debe ser convertido a un código de maquina entendible por el procesador antes de ser cargado en la memoria.

Para principiantes en el campo de software embebido es recomendable usar un lenguaje de alto nivel para el desarrollo del firmware. La razón de esto es: escribir código en un lenguaje de alto nivel es fácil, el código es altamente portable lo que significa que el código puede ser usado para correr en diferentes procesadores/controladores con una pocas modificaciones.

El proceso de desarrollo de software embebido en lenguaje ensamblador es tedioso y consume mucho tiempo. Para ello, el desarrollador necesita conocer todo el conjunto de instrucciones del procesador/controlador. Un programador que escribe el programa usando lenguaje ensamblador escribe el programa según su punto de vista. Por lo tanto, el software es dependiente del desarrollador, lo que resulta difícil para otra persona entender el código escrito en ensamblador si no fue bien documentado.

2.4 Tecnología NFC

2.4.1 Tecnologías de transmisión de corto alcance

Las tecnologías inalámbricas de corto alcance tienen una gran aceptación en la actualidad y están tan inmersas en nuestra sociedad que casi pasan desapercibidas aunque sean tan útiles. La importancia que tienen se la han ganado por la ayuda y por la manera en que facilitan la vida cotidiana de las personas que es casi imposible imaginarnos, por ejemplo, un teléfono móvil sin Bluetooth con el que se pueda transferir y receptar archivos inclusive a las PCs u otros dispositivos como cámaras digitales haciendo que nos olvidemos de los molestosos cables. Así es como se presenta el escenario de las tecnologías inalámbricas de corto alcance, un ambiente que cada vez presenta más aplicaciones en las que puedan encajar, con innovaciones llamativas que las diferencian y que las hacen competir entre ellas para ganar más popularidad pero también brindando compatibilidad con las ya existentes. [5]

2.4.2 Ventajas y desventajas

Las diferentes tecnologías inalámbricas de corto alcance nos dan la oportunidad de ayudar no solo en cuanto se refiere a la comunicación entre personas, sino también de facilitar los procesos mediante su aplicación dentro de un sin número de escenarios. Cada una de éstas compite con las otras brindando sus mejores características para ganarse un espacio y triunfar a la hora de elección entre los usuarios. Sus diferentes configuraciones y modos de funcionamiento así como características de velocidad, alcance, tiempo de establecimiento de la comunicación, seguridades, etc., hace que se ajusten a las distintas necesidades que las personas tienen. A continuación se presenta la tabla 1 en la que se compara las distintas características que las tecnologías de corto alcance ofrecen: [5]

	NFC	BLUETOOTH	RFID	ZIGBEE
Establecimiento de la comunicación	Menos a 0,1s	6s	Menor a 0,1s	30ms
Velocidad de transmisión	424Kbps 848 Kbps	Sobre los 2,1 Mbps La versión 3.0 soporta sobre los 24Mps	424 Kbps	250 kbps
Alcance	10cm	10cm depende de la versión	Más de 3m	70m
Consumo de baterías	Bajo	Alto	Bajo	Bajo
Costo de equipos	Mediano	Relativamente mediano	Bajo	Bajo
Seguridad	Alta	Alta con encriptación	Vulnerable	----------
Experiencia de conexión	Simplemente con un toque	Necesita configuración	Sin configuración	Sin configuración

Tablas 1: Comparación entre las principales tecnologías de corto alcance [5]

2.4.3 Modos de comunicación

Existen dos modos de comunicación NFC:

- **Modo de comunicación pasiva**: en este caso el dispositivo emisor proporciona un campo portador y el dispositivo receptor responde modulando el campo existente. En este modo el receptor obtiene el poder de operación por el campo electromagnético proporcionado por el emisor. Un ejemplo de este modo es la comunicación entre un teléfono móvil y una etiqueta NFC, es decir una conexión entre un dispositivo activado por energía y una etiqueta pasiva. [6] (Ver Figura 3)
- **Modo de comunicación activa**: tanto el emisor como el dispositivo receptor se comunican por la generación alternada de sus propios campos. Uno de los dispositivos debe desactivar su campo de radio frecuencia mientras espera la llegada de datos. Para este modo ambos dispositivos deben contar un una fuente de poder o energía. Un claro ejemplo de este tipo de comunicación es la realizada entre dos teléfonos móviles, es decir la comunicación entre 2 dispositivos que tengan una fuente de poder activa y capacidades computacionales. [6]

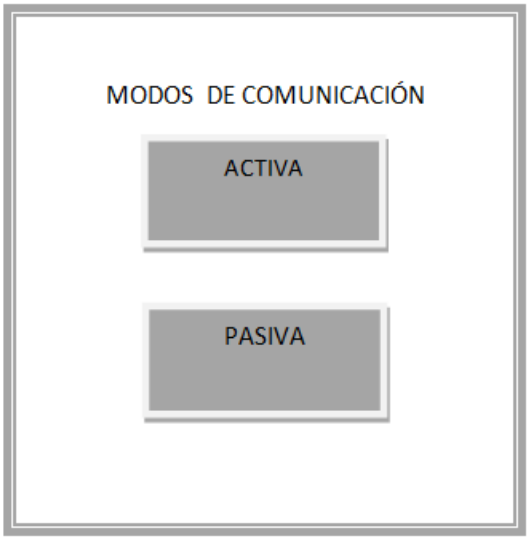


Figura 3: Modos Comunicación de NFC

2.4.4 Modos de operación

Los dispositivos son capaces de operar en tres modos distintos (Ver Figura 4):

- **Modo punto a punto**: este es el modo clásico de operación de *Near Field Communication*, permite una conexión de datos a una velocidad aproximada de 424 kBit/seg. Las propiedades de electromagnetismo y el protocolo usado se encuentran estandarizadas en la ISO 18092 y ECMA 320/340 (ISO/IEC 18092 (ECMA-340)).
- **Modo escritura / lectura**: una funcionalidad adicional de los dispositivos NFC es la habilidad de leer y escribir etiquetas y tarjetas inteligentes. Como ya se indicó en el modo de comunicación pasiva, el dispositivo puede actuar como un emisor y la etiqueta como un receptor pasivo. En este modo de operación la velocidad de transmisión de datos se aproxima a los 106 Kbit/seg.
- **Modo de emulación de etiquetas**: en este modo los dispositivos NFC pueden emular el comportamiento y propiedades de una tarjeta inteligente con el estándar ISO 14443. Un lector no tiene la capacidad de distinguir entre un dispositivo operando en modo de emulación o una tarjeta inteligente ordinaria. Esto implica una gran ventaja, puesto que actualmente existe una infraestructura de lectura desarrollada para tarjetas inteligentes, estas no tienen que ser reemplazadas y se las puede aprovechar con tecnología NFC. [6]

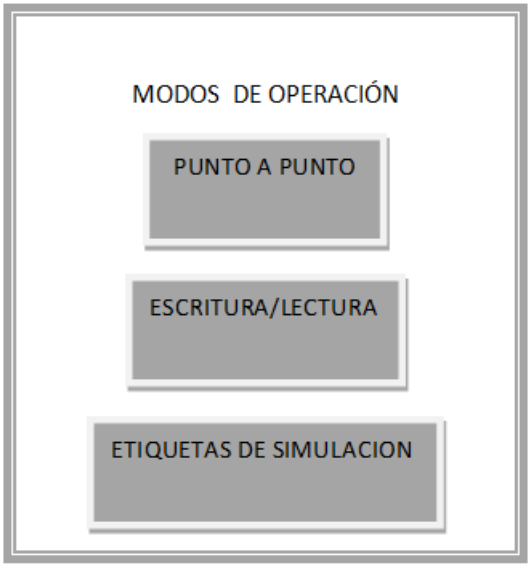


Figura 4: Modos Operación de NFC

2.4.5 Especificaciones técnicas

NFC fue aprobado como un estándar ISO/IEC el 08 de diciembre del 2003 y posteriormente como un estándar ECMA8. Al igual que la ISO/IEC 14443 (Estándar internacional relacionado con tarjetas de identificación electrónicas), se comunica vía inducción de campo magnético, donde dos lazos de antena son localizados dentro de cada campo cercano del otro, formando efectivamente un transformador núcleo de aire.

NFC opera dentro de la banda de radio frecuencia de los 13.56 MHz en ISM9 que está disponible en bandas no licenciadas es decir que no se requiere pagar para su utilización, además posee un ancho de banda de 2MHz. NFC incorpora una variedad de estándares pre-existentes incluyendo ISO/IEC 14443 de ambos tipos, tipo A (normal) y tipo B (banking/short range), y FeliCa10. Por lo tanto los teléfonos habilitados para NFC muestran interoperabilidad básica con módulos que ya existen como RFID.

La distancia a la que operan dos dispositivos con NFC está situada entre los 15 cm y 20 cm de proximidad, además las velocidades de transmisión de esta tecnología van entre los 106, 212, 424 u 848 Kbps.

La comunicación NFC es bidireccional, por lo tanto los dispositivos NFC son capaces de transmitir y recibir datos al mismo tiempo. De esta manera, ellos pueden verificar el campo de Radio Frecuencia y detectar una colisión si la señal recibida no coincide con la señal transmitida. [6]

2.4.6 Estándares de comunicación

El NFC Forum es una asociación industrial sin fines de lucro fundada por NXP Semiconductors, Sony Corporation y Nokia para regular el uso de la interacción inalámbrica de corto alcance en la electrónica de consumo, dispositivos móviles y los PCs. Promueve la implantación y la estandarización de la Tecnología NFC como mecanismo para la interoperabilidad entre dispositivos y servicios. [7]

Dentro de los estándares de NFC se ha establecido un formato común de datos para que los dispositivos NFC puedan compartir información entre sí. Estos estándares señalan las especificaciones que permiten la comunicación.

- NFC Data Exchange Format (NDEF) Especifica un formato común y compacto para el intercambio de datos.
- NFC Record Type Definition (RTD) Especifica tipos de registros estándar que pueden ser enviados en los mensajes intercambiados entre los dispositivos NFC.
    - Smart Poster RTD: Para posters que incorporen etiquetas con datos (URLs, SMSs o números de teléfono).
    - Text RTD: Para registros que solo contienen texto.
    - Uniform Resource Identifier (URI) RTD: Para registros que se refieren a un recurso de Internet

2.4.7 Mensajes NDEF

Los mensajes NDEF proporcionan un método estandarizado para que un lector se comunique con un dispositivo NFC. El mensaje NDEF contiene múltiples registros, como se muestra. Obtiene soporte NDEF solo cuando se trabaja con etiquetas estandarizadas; las etiquetas propietarias generalmente no brindan este soporte. El estándar NFC admite cinco tipos de etiquetas, todas compatibles con el mismo formato de mensaje NDEF. [8]

Descifrar los registros NDEF

El campo Formato de nombre de tipo (TNF) identifica el tipo de contenido que contiene el registro. Estos son los tipos estándar de contenido que puede encontrar en un registro NDEF:

- **0 - Vacío:** El registro no contiene ninguna información.

- **1 - Bien conocido:** Los datos están definidos por la especificación de definición de tipo de registro (RTD) disponible en NFC Forum.
- **2 - Extensiones multipropósito de correo de Internet (MIME):** Este es uno de los tipos de datos que normalmente se encuentran en las comunicaciones de Internet según lo definido por RFC 2046.
- **3 - Identificador uniforme de recursos (URI):** Este es un puntero a un recurso que sigue la sintaxis de RFC 3986.
- **4 - Externo:** Estos son datos definidos por el usuario que se basan en el formato especificado por la especificación RTD.
- **5 - Desconocido:** El tipo de datos es desconocido, lo que significa que debe establecer la longitud del tipo en 0.
- **6 - Sin cambios:** Algunas cargas son fragmentadas, lo que significa que los datos son demasiado grandes para caber dentro de un solo registro. En este caso, cada registro contiene una parte de los datos, un fragmento. Este TNF indica que este no es el primer registro en el fragmento: es uno de los registros intermedios o de terminación. El TNF no se modifica con respecto al tipo de datos encontrados en el primer registro del conjunto fragmentado.
- **7 - Reservado:** Este valor está reservado para uso futuro.

La bandera de IL le dice si el registro contiene un campo de longitud de ID. No especifica la longitud de la ID, simplemente le dice que este valor está disponible.

El indicador SR determina si el registro es un registro corto. Un registro corto es uno con una longitud de carga útil <= 255 bytes. Los registros normales pueden tener longitudes de carga superiores a 255 bytes hasta un máximo de 4 GB. Muchos casos de uso requieren la mayor economía del tamaño del mensaje. El indicador SR permite el uso de un encabezado de registro comprimido al especificar la longitud de la carga útil en un solo byte en lugar de requerir los cuatro bytes normales. La bandera CF le dice cuándo se fragmenta un registro. En otras palabras, si ve este marcador configurado, leer un solo registro no le proporcionará todos los datos para ese elemento de datos. Debe leer todos los registros asociados con ese elemento de datos para obtener la información completa al respecto.

Un mensaje NDEF puede contener múltiples registros. El primer registro de un mensaje tiene el indicador MB (inicio de mensaje) establecido en verdadero para que sepa que este es el primer registro. El último registro en el mensaje tiene la bandera ME establecida para que sepa que este es el último registro. Todos los registros intermedios tienen los indicadores de MB y ME configurados en falso.

El campo Longitud de tipo contiene la longitud del tipo de carga útil en bytes. El tipo de carga útil especifica el tipo exacto de datos encontrados en la carga útil. Por ejemplo, el simple hecho de saber que el TNF es un tipo de datos MIME no es suficiente: debe conocer el tipo MIME preciso (como “text / plain") para procesar los datos.

El campo Longitud de la carga útil contiene la longitud de la carga en bytes. Un registro puede contener hasta 4, 294, 967, 295 bytes (o 2^{32} - 1 bytes) de datos. El campo Longitud de ID contiene la longitud del campo ID en bytes. El tipo viene a continuación. Esta es una definición del tipo preciso de datos que contiene la carga útil.

El campo ID proporciona los medios para que las aplicaciones externas identifiquen toda la carga útil transportada dentro de un registro NDEF. Solo el primer registro contiene una identificación; los registros NDEF intermedios o de terminación no tienen un campo ID.

Finalmente, después de definir todos estos detalles, llegas a la carga útil. La carga útil es la información. Sin embargo, sin conocer toda esta otra información, la carga útil podría no tener sentido. Necesita toda esta otra información para comprender con qué tipo de datos está trabajando. [8]

2.5 Metodología de desarrollo XP

2.5.1 Introducción a XP

La eXtreme Programming (programación extrema) también llamado XP, es una metodología que tiene su origen en 1996, de la mano de Kent Beck, Ward Cunningham y Ron Jeffries.

A diferencia de Scrum, XP propone solo un conjunto de prácticas técnicas, que aplicadas de manera simultánea, pretenden enfatizar los efectos positivos de en un proyecto de desarrollo de Software. [9]

“La programación extrema (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.” [10]

A continuación presentaremos las características esenciales de XP:

1. Está orientada a quien produce y usa el software.
2. Reduce el costo del cambio en todas las etapas del ciclo de vida del sistema.
3. Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo.

Esta metodología además prescribe un conjunto de valores y prácticas, que permite a los desarrolladores dedicarse a lo que hacen mejor: programar. La XP elimina la necesidad de dedicar tiempo a labores tediosas y burocráticas, prescritas por los procesos no ágiles, tales como: exhaustivos documentos de proyecto, diagramas de Gantt, enormes volúmenes de listas de requerimientos, juntas de revisión interminables, etcétera.

La mayoría de los proyectos han seguido una metodología de desarrollo tradicional. Dicha metodología exige, en general, cumplir con estas actividades:

1. Levantamiento exhaustivo de requerimientos.
2. Detección de defectos en las fases iniciales.
3. Reducción en el número de cambios, tanto como sea posible.
4. Análisis y diseño, tan completo como sea posible.
5. Diseño genérico, intentando anticiparse a futuras necesidades.

Se realizan las siguientes prácticas:

- El juego de la planificación
- Pequeñas entregas
- Metáfora
- Diseño simple
- Pruebas
- Refactorización
- Historias de usuarios

2.5.2 Ciclo de Vida del Software en XP

El ciclo de vida de la metodología XP, es de carácter interactivo en el desarrollo y su vez incremental. Una iteración de desarrollo es un periodo de tiempo en el que se desarrolla las Historias de Usuario. Esta metodología, consta de las siguientes fases, que, se tomarán en cuenta durante el desarrollo del proyecto.

Fase de la exploración

Es la fase en la que se define el alcance general del proyecto, los clientes plantean las Historias de Usuario, que serán lo primero que se desarrollará para la primera entrega al cliente. Dura típicamente un par de semanas y el resultado es una visión general del sistema, y un plazo total estimado.

Fase de planificación

En esta fase se planificará las prioridades de las historias de usuario. También, se describe el alcance del primer Release (pequeño programa). Por reglamento de la metodología, la versión 1 de todo el sistema no excede los 2 meses.

Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, adquiriendo con el cliente todos los datos que sean necesarios. Al final de la última iteración, el sistema estará listo para ser entregado y llevarlo a la fase producción.

Fase de producción

En esta fase, el Release puede cambiar, de acuerdo a las observaciones del cliente. Las sugerencias y observaciones pospuestas por el cliente, se las documentan, para posteriores iteraciones.

Fase de mantenimiento

En esta fase, el desarrollo del sistema puede ser lento, debido a que se realizan modificaciones de las observaciones que realizó el cliente en anteriores iteraciones.

Fase de muerte

Ésta, es la fase en la que el cliente no tiene más Historias de Usuario (observaciones) para modificarlas al sistema final. Solo quedan tareas de rendimiento confiabilidad del sistema, y se genera la documentación final del sistema.

2.5.3 Roles XP

La metodología XP de acuerdo con la propuesta original de Kent Beck propone los siguientes roles:

- Cliente:
    - Escribe las historias de usuario (HU)
    - Explica las HU
    - Encargado de las pruebas funcionales
    - Establece las prioridades, puede decidir sobre el sistema
- Programador:
    - Pieza básica en desarrollos XP y más responsabilidad que en otros modos de desarrollo
    - Define las tareas a partir de las HU
    - Responsable del código y del diseño
    - Responsable de la integridad del sistema (pruebas)
    - Capacidad de comunicación
- Encargado de pruebas:
    - Apoya al cliente en la preparación y realización de las pruebas funcionales
    - Ejecuta las pruebas funcionales y publica los resultados
- Entrenador (coach)
    - Experto en XP, responsable del proceso, guía
    - Identifica los problemas y los ataja rápidamente

- Consultor
    - Apoya al equipo XP
- Jefe del proyecto
    - Cubre las necesidades del equipo de XP
    - Confía y asegura que se alcanzaran los objetivos

2.5.4 Reglas y Prácticas

La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en:

Reglas y prácticas para la planificación

- Historias de usuarios

Las “Historias de usuarios”, estas “historias” son escritas por el cliente, son descripciones cortas de lo que el sistema debe realizar se encuentra en el nivel de detalle requerido, deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo.

- Plan de entregas (“Release Plan”)

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. Se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores.

- Plan de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

- Reuniones diarias de seguimiento

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones.

Reglas y prácticas para el diseño

- Simplicidad: XP propone implementar un diseño lo más simple posible.
- Soluciones “Spike”: son pequeños programas de pruebas para explorar diferentes soluciones.
- Recodificación: consiste en escribir nuevamente el código de un programa, sin cambiar su funcionalidad, para hacerlo más simple, conciso y/o entendible.
- Metáforas: para explicar el propósito de proyecto, usando la misma nomenclatura ara los métodos y las clases.

Reglas y prácticas para el Desarrollo

- Disponibilidad del cliente

Al comienzo del proyecto, el cliente debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de “alto nivel”, no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo.

Reglas y prácticas para las Pruebas

- Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. En este sentido, el sistema y el conjunto de pruebas deben ser guardadas junto con el código, para que pueda ser utilizado por otros desarrolladores.

- Detección y corrección de errores

Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

- Pruebas de Aceptación

Permite confirmar que la historia ha sido implementada correctamente.

CAPITULO III

3 Herramientas De Análisis Y Tecnológicas

3.1 Herramientas de Análisis

Un proceso muy importante para la investigación es la recolección de elementos para obtener datos de la realidad, para esto se crean instrumentos que debe contemplar en su construcción la forma (técnica para cumplir con la tarea) y contenido (especificación de datos que necesitamos conseguir). El instrumento permite sintetizar toda la labor previa de la investigación y por esto la importancia que tiene para poder obtener datos verdaderos.

3.1.1 Encuestas y resultados

Se realizó una encuesta de forma anónima a personas entre ellas: usuarios, docentes y no docentes que trabajan en el auditorio para recabar información en forma verbal, a través de preguntas para conocer la situación actual y el manejo del control de acceso y de asistencia al mismo y la pérdida de tiempo en el acceso al mismo. Se los entrevistó de forma individual y se captó el parecer de todas las personas sobre el prototipo que se construyó.

Análisis estadístico de las encuestas

Los resultados que se obtuvieron de la recolección de datos cualitativos, se extraen de preguntas abiertas, los resultados fueron unánimes como se ve en la tabla 2.

Preguntas:

1. ¿Cree usted qué se pierde mucho tiempo en el acceso al auditorio de la FTyCA al momento de realizar algún tipo de evento?
2. ¿Cree usted qué es necesario la implementación de un sistema de control de acceso y monitoreo en el auditorio?
3. ¿Cree usted qué este sistema facilitaría el control del ingreso y acceso al auditorio?
4. ¿Cree usted que se necesita sistematizar el control de asistencia de las actividades del auditorio para agilizar el proceso?
5. ¿Sabía usted que cada asistente demora en promedio más de 3 minutos en inscribirse, y si en un auditorio ingresan 60 personas necesitaríamos 180 minutos para realizar la inscripción del evento?

6. ¿Qué tipo de control de asistencia utiliza usted? (hoja en Excel, listado, contando las personas una por una).

Pregunta 1	Respuesta = SI
Pregunta 2	Respuesta = SI
Pregunta 3	Respuesta = SI
Pregunta 4	Respuesta = SI
Pregunta 5	Respuesta = NO
Pregunta 6	Listado

Tablas 2: Resultado de la encuesta

De acuerdo a los resultados obtenidos con las personas seleccionadas de la FTyCA es posible concluir de manera unánime la necesidad de incorporar un sistema de control de acceso y monitoreo en el auditorio para sistematizar esta actividad.

3.2 Herramientas Tecnológicas

Para el diseño de SCARAA, inicialmente se realizó un proceso de análisis y selección de cada uno de los elementos que hacen parte de un sistema de este tipo (arquitectura principal, elementos periféricos) y a partir de esto se realizó el acondicionamiento necesario y la programación para cada uno de los elementos, con el fin de cumplir con las características y funciones planteadas para este proyecto.

3.2.1 Hardware

Intel® Galileo Gen 1

Galileo es un microcontrolador basado en un procesador Intel® Quark Soc X1000 con una arquitectura de 32-bit. Es la primera placa basada en la arquitectura Intel® diseñada para ser pin-compatible tanto en hardware como en software con los shields Arduino UNO. [11]

Se utilizó la Galileo porque posee un procesamiento de gran alcance comparado a una Arduino, esta permite guardar la información en la tarjeta SD, los usuarios son capaces de transmitir archivos, el acceso a internet debe estar configurado a través de conexiones Ethernet o wifi. Esta posee pines digitales y analógicos, comunicación serie a través de una ranura FTDI usado para comunicarse con la consola Linux dentro de ella, también posee comunicación I2C, también posee algunos pines para la configuración por PWM. [11]

Características principales

- 14 pines entrada/salida digitales
  -posee 6 pines que pueden ser utilizados como salidas PWM (modulación de ancho de pulso).
  -operan con 3.3 V o 5 V.
  -cada pin provee una corriente máxima de 10 mA. Y puede recibir hasta 25 mA.
  -resistencia interna de pull-up (desconectada por defecto) de 5.6 K a 10 KOhms.
- A0-A5: son 6 entradas analógicas, conectadas con un conversor analógico-digital (AD7298)
  -resolución de 12 bits (4096 valores diferentes)
  -rango entre 0 y 5V.
- Bus I2C, TWI:
  -TWI (Two Wire Interface): utiliza los pines A4 o SDA y A5 o SCL.
  Para esta comunicación la trasmisión de datos en serie se realiza en modo asíncrono. Este protocola utiliza solo dos cables para la comunicación entre dos o más circuitos integrados. Posee dos líneas de comunicación bidireccionales llamadas SDA (Serial Data) y SCL (Serial Clock) con resistencias de pull-up, las cuales se utilizan para trasferencias de datos entre dispositivos. Uno de los dos dispositivos que controla todo el proceso se conoce como maestro y la otra que responde a las consultas de maestro se conoce como dispositivo esclavo. La placa de desarrollo Intel® Galileo solo puede funcionar como maestro.
- SPI
  -rango de 4 Mhz-25Mhz
  Este tipo de comunicación tiene solo 3 líneas (MISO, MOSI y SCK) para la transmisión de datos, así como para handshake. En la comunicación SPI solo hay un controlador maestro y un controlador esclavo y por lo tanto el esclavo no requiere direccionamiento. Es un proceso de comunicación de datos en serie “Full duplex”.
  Aunque intel Galileo posee un controlador nativo para SPI, actuara solo como maestro y no como esclavo.

- UART (puerto serie)
  -velocidad configurable
  -pines: 0 (Rx) y 1 (Tx)
- ICSP (SPI)
  -son 6 pines para programación serie que permiten la conexión de shields externos. Estos pines permiten la comunicación SPI utilizando la librería adecuada.

- VIN
  -se permite alimentar la placa mediante este pin de entrada utilizando una fuente externa (deberá ser de 5V)
- Pin 5V salida:
  -este pin provee una salida de 5V con una corriente máxima de 800 mA.
- Pin 3.3V salida:
  -este pin provee una salida de 3.3V con una corriente máxima de 800 mA.
- GND
  -pines de masa general
- IOREF
  -este pin permite a un shield conectado y configurado apropiadamente adaptarse al voltaje previsto por la placa.
  -este pin está controlado por un jumper en la placa el cual permite el uso de 3.3V o 5V.
- RESET botón/pin:
  -este pin mediante una entrada en bajo, permite resetear la placa.
- AREF
  -este pin no es utilizado.
  -permite utilizar un voltaje de referencia externo.

Detalles de arquitectura:

- Procesador Intel Quark SoC X1000 400Mhz 32-bits.
- ISA (Set de instrucciones) compatible con Intel® Pentium.
- 16 KBytes memoria cache L1.
- 512 KBytes SRAM.
- Programación simple: Un hilo, un núcleo, velocidad constante de clock.
- CPU compatible con ACPI, con estados de sleep.
- RTC (Real Time Clock – Reloj de tiempo real).
- Conector Ethernet 10/100.

- Slot de tipo PCI Express 2.0 mini-sd slot.
- Puerto USB 2.0 Host.
- Puerto USB 2.0 Cliente.
- Botón Reboot para reiniciar el procesador.
- Botón Reset para reiniciar el sketch.
- Opciones de almacenamiento:
  - Memoria 8MB flash para firmware y el ultimo sketch.
  - Entre 256 y 512 KB son dedicados al almacenamiento del sketch.
  - 512KB SRAM embebida, habilitada por defecto por el firmware.
  - 256MB DRAM, habilitada por defecto por el firmware.
  - Micro SD hasta 32gb de almacenamiento. (Preferentemente clase 10).
  - 11 KB memoria EEPROM programable a través de librería EEPROM.
- Fuente de alimentación: La placa Intel® Galileo utiliza un transformador AC-DC, y es conectada por un conector tipo plug de 2.1mm cuyo rango de funcionamiento es de 5V y 3A.
- Comunicación: Galileo provee una comunicación serie UART TLL (5V/3.3V), que está disponible en los pines digitales 0 (RX) y 1 (TX). Además, soporta una segunda UART (RS-232) que es conectada vía un conector Jack 3.5mm.Los puertos USB permiten comunicaciones serie (CDC) a través de USB. Esto provee una conexión serial al “Serial Monitor” u otra aplicación. El puerto USB Host permite que Galileo actúe como un host USB para periféricos conectados, tales como mouse, teclados y teléfonos inteligentes.

  Galileo está basada en una placa Arduino que provee un slot mini PCI Express (mPCIe). Este slot permite conectar módulos mPCIe en tamaño completo o en mitad de tamaño (con un adaptador) para que puedan ser conectados a la placa y, también, proveer un puerto USB Host adicional. Cualquier módulo estándar mPCIe puede ser conectado y usado para proveer aplicaciones como WiFi, Bluetooth o conectividad celular. Inicialmente, el slot mPCIe provee soporte para la librería WiFi.

  Un conector Ethernet RJ45 permite conectar a Galileo con redes cableadas. Cuando se conecta a una red, se debe asignar una dirección IP y una dirección MAC. La interfaz Ethernet es totalmente compatible y no requiere el uso de la interfaz SPI como Arduino.
- El lector de tarjetas SD es accesible a través de las librerías SD. La comunicación entre Galileo y la tarjeta SD es provista por un controlador SD integrado y no requiere el uso de la interfaz SPI como otras placas Arduino. (Ver Figura 5)

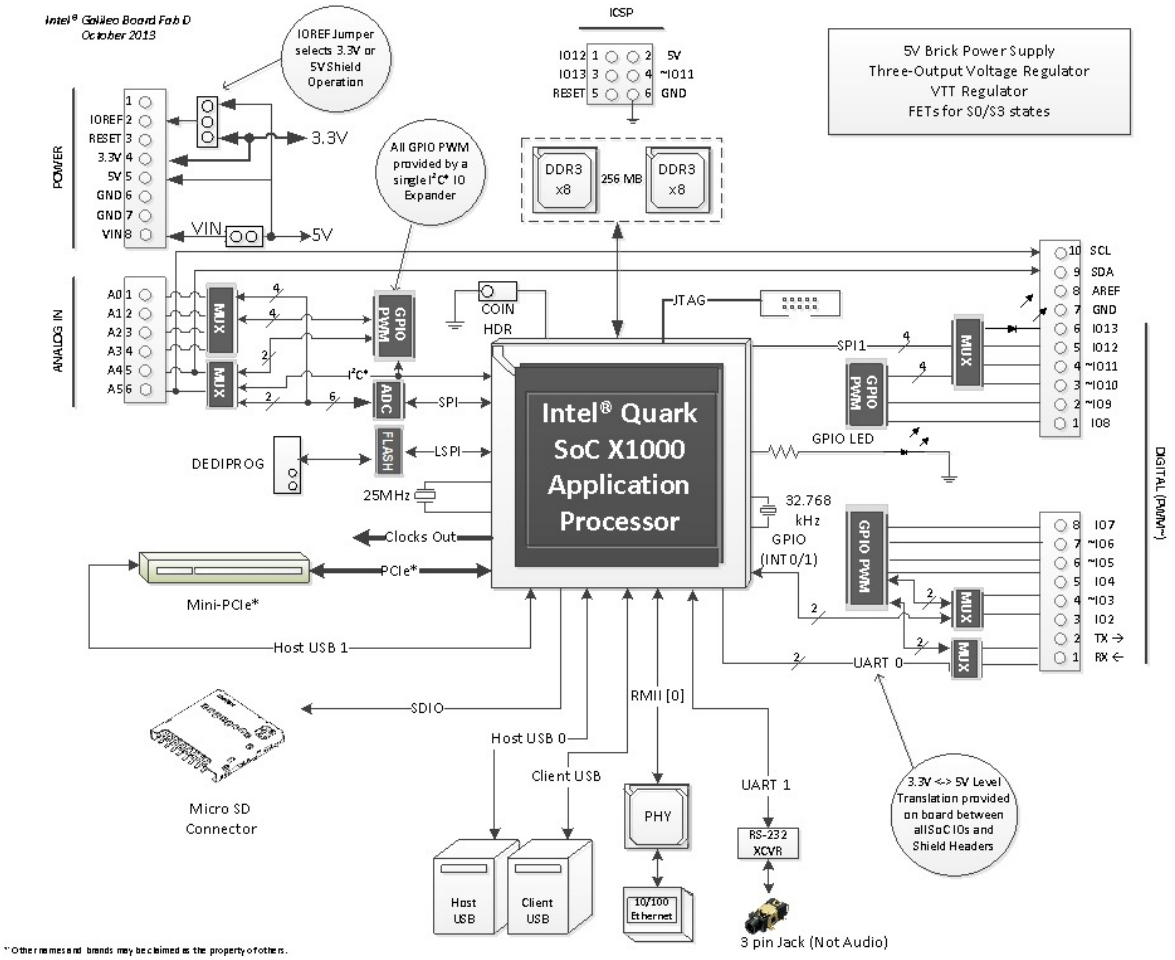


Figura 5: "Intel Galileo block diagram": muestra cómo están configurados todos los componentes dentro de un diagrama de bloque. [12]

Chip PN532 NFC/RFID módulo versión 3

Este módulo fue construido por NXP®, posee un chip PN532 muy popular en el área NFC; el fabricante ofrece un manual de usuario en su foro para ayudar a los desarrolladores, y además posee a disposición una biblioteca propia. (Ver Figura 6)

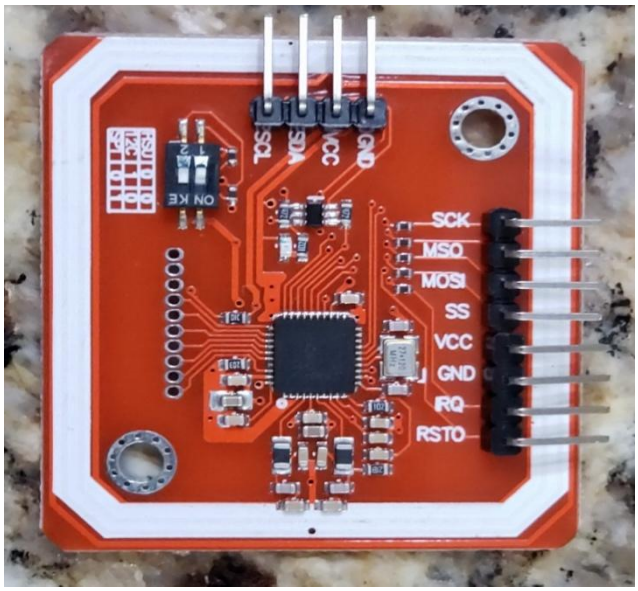


Figura 6: Chip PN532 NFC/RFID [11]

Los desarrolladores si desean pueden utilizar las interfaces de comunicación como UART (HSU), SPI o I2C, ya que este módulo facilita la conexión de esos pines. [11]

Características:

1. Soporte I2C, SPI y HSU (UART de alta velocidad)
2. Compatibilidad con el modo lector / escritor RFID
    a. Tarjetas Mifare 1k, 4k, Ultralight y DesFire
    b. Tarjetas ISO / IEC 14443-4 como CD97BX, CD light, DesFire, P5CN072 (SMX)
    c. Tarjetas Innovision Jewel como la tarjeta IRT5001
    d. Tarjeta FeliCa Tarjetas como RCS_860 y RCS_854
3. Plug and play, compatible con Arduino
4. Antena PCB incorporada, con una distancia de comunicación de 5 cm ~ 7 cm
5. Palanca de cambios a nivel, Estándar 5V TTL para I2C y UART, 3.3V TTL SPI
6. Funciona como lector / escritor RFID
7. Trabaja como una tarjeta 1443-A o una tarjeta virtual
8. Admite NFC con un teléfono Android
9. Tamaño pequeño: 43 mm * 41 mm * 4 mm

Interfaz (Ver Tabla 3):

- VCC: 3.3V ~ 5V

- I2C / UART: 3.3V ~ 24V TTL
- SPI: 3.3V TTL con resistencias de 100 ohmios en serie. Se puede conectar directamente a la interfaz de 5V de un microcontrolador como Arduino. [11]

La configuración del interruptor se muestra a continuación:

Interfaz de trabajo	SWICTH 1	SWICTH 2
HSU	OFF	OFF
I2C	ONN	OFF
SPI	OFF	ON

Tablas 3: Configuración de los swicth [13]

Funcionamiento del PN532

En la Figura 7 el PN532 incluye un procesador 80C51 con 40 KB ROM y 1 KB RAM, junto con los elementos necesarios para hacer funcionar correctamente la comunicación NFC.

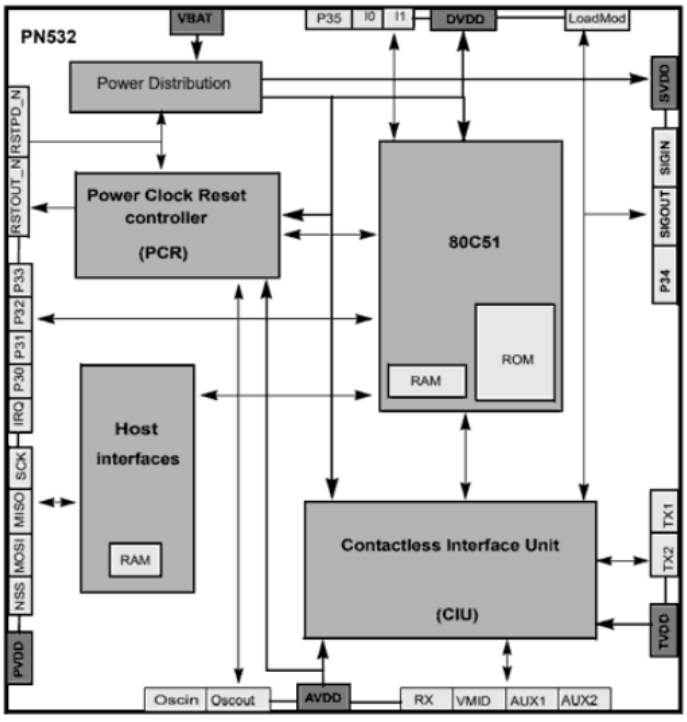


Figura 7: Funcionamiento pn532 [13]

Modos de Operación: El PN532 soporta 6 modos de operación.

- ISO/IEC 14443A/MIFARE Lector/Grabador.
- ISO/IEC 14443A/MIFARE Card MIFARE Classic 1K y MIFARE Classic 4K Card.
- ISO/IEC 14443B Lector/Grabador.
- FeliCa Lector/Grabador.
- FeliCa Card emulación.
- ISO/IEC 18092, ECMA 340 Peer-to-Peer

Las distancias típicas de actuación son de 50mm para lectura y escritura, y 100mm para emulación. La velocidad de lectura es de hasta 212 kbits/s y la de escritura de hasta 424kbts/s. Estos valores dependen, entre otros, de la antena integrada en el módulo. En los módulos Maker el alcance típico es 30mm a 50mm.

El PN532 opera a 3.3V pero admite tensiones de alimentación de 2.7V a 5.4V. Las líneas de comunicación I2C/UART funcionan de 3.3V a 24V TTL. Sin embargo, la interface SPI funciona a 3.3, pero incorpora resistencias de 100 Ohm en serie de forma que también puede ser conectado a 5V.

El consumo es de 100mA en modo Stand-By y 120mA durante la lectura y escritura. Adicionalmente dispone de dos modos de baja energía, uno modo Soft-Power-Down con un consumo de 22uA, y un modo Hard-Power-Down con un consumo de 1uA.

Comunicación por I2C

La comunicación utilizada para este sistema fue por I2C, la conexión se realiza de la siguiente forma:

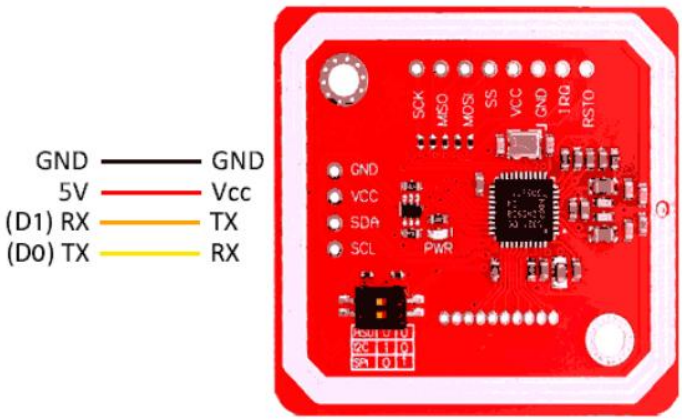


Figura 8: Conexión por I2C [13]

El PN532 es un chip NFC que podemos conectar a cualquier Sistema Embebido para leer y escribir tarjetas NFC, comunicarse con teléfonos móviles, o incluso actuar como tag NFC. Es ampliamente utilizado en todo tipo de dispositivos comerciales que implementan NFC.

Pantalla táctil Nextion HMI

Para comunicarse con el usuario y visualizar la información de SCARAA, se utilizó una pantalla táctil para aprovechar su doble función y no utilizar un teclado como elemento adicional. La referencia de la pantalla a utilizar fue una Nextion de 3.5", cuenta con una interfaz HMI y con su propio software para crear las diferentes aplicaciones, su manejo es sencillo, en donde no se utiliza código de programación complicado. Su conexión con la placa Galileo es a través de una comunicación UART.

Nextion es una interfaz máquina-humano HMI, que combina una pantalla táctil TFT con un procesador y memoria integrada, además cuenta con un software gratuito y descargable Editor Nextion, con este software se puede desarrollar rápidamente la Interfaz de Usuario mediante componentes de arrastrar y soltar (gráficos, texto, botones, controles deslizantes, etc.) e instrucciones basadas en código ASCII para codificar cómo interactúan los componentes al lado de la pantalla. Con solo 2 cables de transmisión (RX, TX), la pantalla se conecta rápidamente a MCU, a través de una fuente de alimentación de 5V. Por medio de la comunicación serie se notifica a través de eventos al MCU para que este pueda actuar, y utiliza simples instrucciones basadas en código ASCII para que el MCU pueda proporcionar fácilmente actualizaciones de progreso y estado a sus usuarios. [14]

Características:

- ✓ Resolución 480 x 320
- ✓ RGB 65K colores reales
- ✓ Pantalla TFT con panel táctil resistivo integrado
- ✓ Interfaz serial TTL de 4 pines
- ✓ Memoria flash 16M para código de aplicación de usuario y datos
- ✓ Ranura para tarjeta micro-SD a bordo para actualización de firmware
- ✓ Área visual: 73.44mm (L) × 48.96mm (W)
- ✓ Brillo ajustable: 0 ~ 180 liendres, el intervalo de ajuste es del 1%
- ✓ Fuente de alimentación recomendada de 5V500mA DC
- ✓ Consumo de energía 5V145mA
- ✓ Certificados: CE / EMC, RoHS (certificados)

Hay varias pantallas con prestaciones HMI. Algunas de ellas son: Nextion, Mikroelektrokica, 4D systems, Riverdi, HotMCU, FTDI chip, NewheavenDisplay, SmartGPU2; manejan excelentes variantes táctiles.

Nextion es una solución Human Machine Interface (HMI) que proporciona una interfaz de control y visualización entre un humano, máquina y un proceso, Es la mejor solución para reemplazar la pantalla LCD tradicional. (Ver Figura 9)

La pantalla táctil cumple el objetivo primordial de visualizar información del sistema, como datos captados por los sensores e información del cliente, la misma proporciona una interfaz amigable con el usuario.

Por la descripciones mencionada anteriormente se seleccionó la pantalla HMI de 3.2" perteneciente a la empresa NEXTION, el cual es compatible con todos los microcontroladores del mercado.

Figura 9: Pantalla Nextion 3.5

3.2.2 Software

Apache

Es un servidor web HTTP de código abierto para plataformas Unix-like (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. En sus inicios se basaba en el código de NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Actualmente es el servidor web más usado en todo el mundo, superando en 2009 los 100 millones de sitios web, el 70% del total. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation. [15]

MySQL

Es un sistema para la administración de bases de datos relacionales (RDBMS) rápido y sólido. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor de MySQL controla el acceso a los datos, para garantizar el uso simultaneo de varios usuarios para proporcional acceso a dichos datos y para asegurar de que sólo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple. Utiliza SQL (del inglés Structured Query Language, lenguaje de consulta estructurado), el lenguaje estándar para la consulta de bases de datos utilizado en todo el mundo.[15]

PHP

Acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. [15]

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutara cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.PHP es un producto de código abierto, lo que quiere decir que se puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo sin coste alguno [15]. Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, peroa su vez ofrece muchas características avanzadas para los programadores profesionales.[15]

Arduino IDE

Arduino IDE es un entorno de desarrollo, un editor de texto y compilador para programar y transferir el contenido de las instrucciones a la placa Galileo en su lenguaje máquina. El lenguaje de programación utilizado es Processing.

El programa se llama IDE, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). Este IDE debe ser instalado en nuestra PC, es un entorno muy sencillo de usar y en él escribiremos el programa que queramos que la Galileo ejecute. Una vez escrito, lo cargaremos a través del USB y la Galileo comenzará a trabajar de forma autónoma.

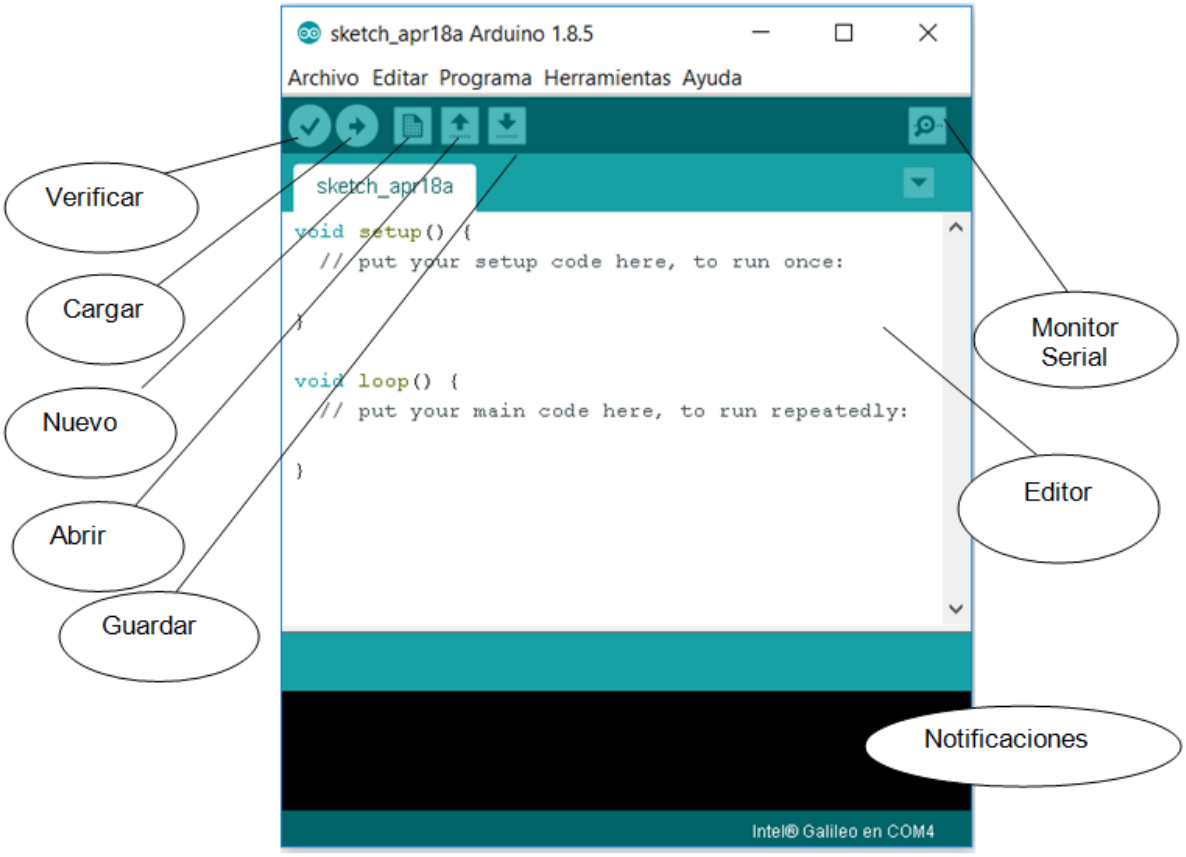


Figura 10: Software arduino IDE

Como se observa en la Figura 10 el Software Arduino IDE se compone de 3 partes principalmente:

- **Botonera** o barra de navegación:
  - *Verificar*: Se encarga de verificar la sintaxis de nuestro programa.
  - *Cargar*: Si la verificación ha sido correcta, podemos cargar el código en nuestra placa Galileo.
  - *Nuevo*: Simplemente abrimos un documento vacío (salvo funciones principales) para comenzar un nuevo programa.
  - *Abrir*: Para abrir proyectos en otros directorios o rutas.
  - *Guardar*: Simplemente guarda el programa en el directorio que especifiquemos (si es la primera vez que lo guardamos).
  - *Monitor serial*: Supongamos que necesitamos saber en algún momento qué ocurre dentro de nuestra placa Galileo, pues bien, mediante el monitor serial podemos enviar datos que se mostrarán en nuestro monitor.
- **Editor de programación**: Es la parte principal de Arduino IDE, básicamente donde se programan las líneas y líneas de código en lenguaje processing.

- **Notificaciones**: Conocido normalmente por consola, es la parte de depuración donde notifica al programador sobre errores de sintaxis, comunicación, etc. [16]

Software Nextion Editor de la pantalla HMI

Se empleó el software de Nextion para diseñar la interfaz de las aplicaciones que se utiliza para visualizar la información y así el usuario pueda interactuar con el SCARAA conociendo los datos, cambiando algunos parámetros, enviando información al SCARAA y actualizando datos.

Este software presenta un entorno grafico de programación bastante sencillo, cuenta con herramientas y controles como botones, campo de textos, barra de progreso, slider, panel de instrumentos, etc., estos a su vez tienen funciones programables que por lo general hacen posible crear cualquier diseño. A continuación en la Figura 11 se muestra el Entorno de Desarrollo Nextion Editor. En el momento de crear un proyecto nuevo, seleccionar el modelo de pantalla con el que se va a trabajar. Si seleccionamos el modelo básico, encontramos dentro del mismo varias opciones de pantallas HMI con sus características correspondientes. También vamos a especificar la orientación de la pantalla para trabajar con ella. (Ver Figura 12)

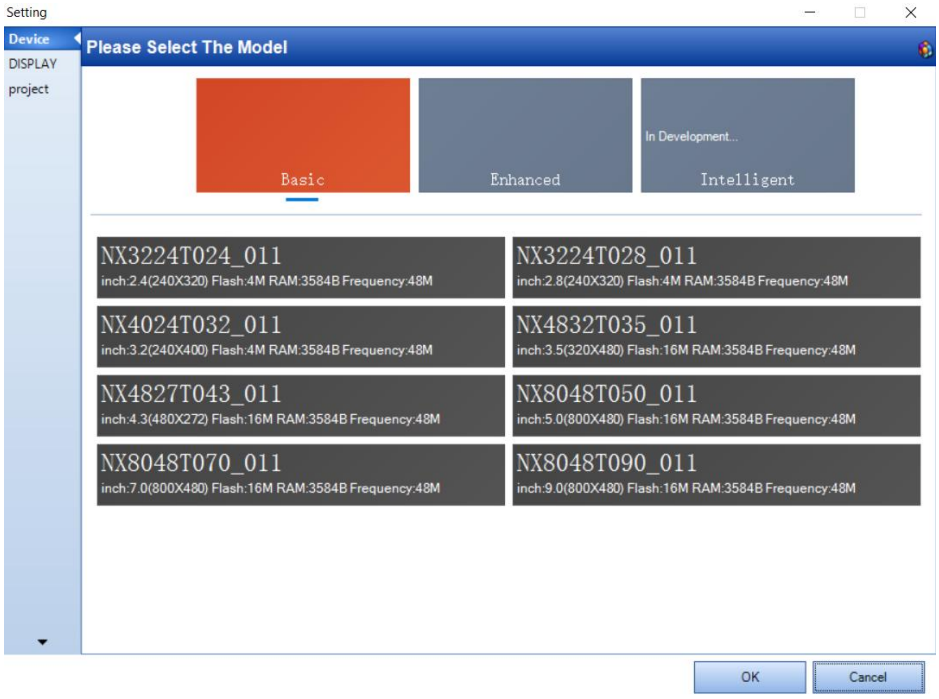


Figura 11: Elección del modelo

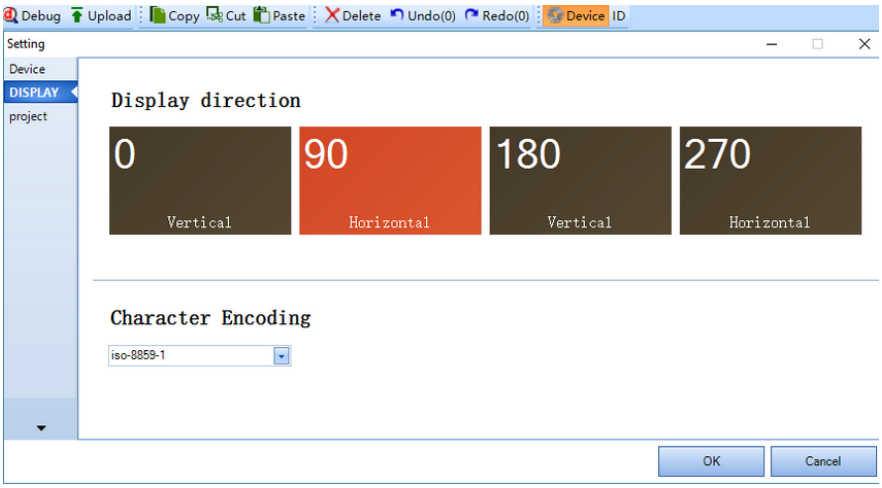


Figura 12: Orientación de la pantalla

Luego de configurar el modelo, orientación y tamaño de la pantalla HMI el editor presenta como en la Figura 13 el contexto de un nuevo proyecto.

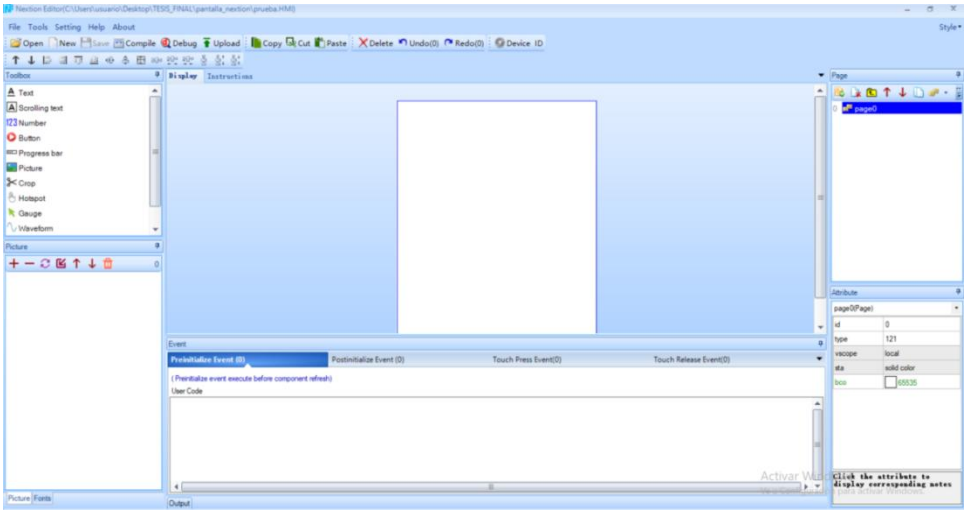


Figura 13: Entorno de programación de Nextion Editor.

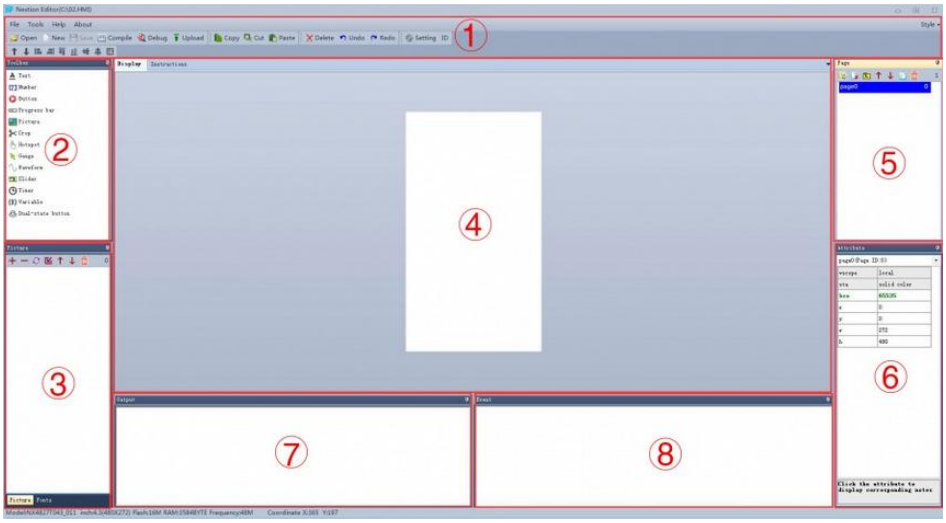


Figura 14: Componentes Entorno de programación de Nextion Editor.

En el ***área 2*** de la Figura 14 se encuentran los componentes que dan importancia, los cuáles son los elementos que podemos incorporar para darle un mayor dinamismo a nuestra pantalla. Cada uno de estos componentes dispone de una serie de atributos que aparecen en el ***área 6***, a los que se le pueden asociar mediante programación de eventos desde el ***área 8***. Esta área es muy interesante, ya que el dinamismo de nuestras pantallas dependerá de cómo realicemos la programación asociada a respuestas de diferentes eventos.

En el ***área 3*** de la Figura 14 es el bloque dónde se puede cargar nuestras imágenes y fuentes y lo más importante a tener en cuenta es que las imágenes deben de tener las mismas dimensiones de la pantalla seleccionada, ya que el editor no nos escala la imagen de ninguna manera.

También se pueden cargar imágenes como el fondo, seleccionado la opción imagen sobre la página. La única diferencia sería que no se puede mover la posición de la misma.

Compilación y simulación de un proyecto

En el ***área 1*** se encuentran 3 botones que aparecen arriba los cuales son: ***Compile, Debuq y Download.***

El botón ***Compile*** revisa la sintaxis de programación para la pantalla HMI y además controla que no se haya excedido en el tamaño de la memoria que va a ocupar. En nuestro caso, nuestra pantalla puede contener 4MB de memoria.

Si se realiza un programa complejo, es conveniente realizar un ***Debug*** de la pantalla para revisar si el comportamiento que realiza la pantalla es el deseado, ya que aunque la compilación sea satisfactoria, es posible que se haya dejado algún elemento por programar correctamente.

Carga de proyectos desde el puerto serie o USB

Finalmente si queremos subir nuestro programa, solamente tendremos que apretar el botón ***Download***. Si nuestra pantalla está conectada directamente al USB o a los pines 0 y 1 del serial de la Galileo, el código de la pantalla se cargará automáticamente.

Carga de proyectos desde la SD

Otra alternativa de carga es guardar el proyecto dentro de una tarjeta SD con un archivo en formato **.tft**. Este proceso requieres de compilar el proyecto al menos una vez y seleccionar el botón ***Open Build Folder*** contenido en el Menú del ***área 1***.

Este botón nos dirigirá a un directorio donde se crean todos los proyectos que estamos realizando y que también se pueden acceder a través de la ruta: ***%appdata% -> Roaming -> Nextion Editor -> bianyi***

Solamente tendremos que copiar el archivo dentro de la tarjeta SD e insertarla en la pantalla. El archivo tiene que estar en el directorio raíz, es decir, no tiene que estar contenido en ningún directorio. Al encender, se cargara automáticamente el archivo y se guardará nuestro programa.

CAPITULO IV

4 DESARROLLO DE LA APLICACIÓN MÓVIL

Antes de pasar a describir el desarrollo de la aplicación, tenemos que numerar los requisitos necesarios analizados a partir de las historias de usuario, que se ajusta al modelo mencionado en el Capítulo 2 sobre la metodología XP, a continuación:

4.1 Las Historias de Usuarios del SCARAA

HISTORIA DE USUARIO Nº 1	
Nombre de la Historia: Gestionar Preinscripción con la aplicación móvil	
Usuario: usuarios	
Prioridad de ejecución: Media **(Alta, Media, Baja)**	
Tiempo estimado: 3 minutos	**Iteración:** 1
Descripción: registrar datos de una inscripción	
Observaciones: para el caso de LOGIN: con poner el usuario (dni) correcto y la contraseña pueden acceder a las ofertas de los distintos eventos. Es necesario que previamente se hayan registrado en REGISTRAR con los datos que solicita la app móvil (dni, nombre, y password)	

Tablas 4: Gestionar Preinscripción con la aplicación móvil

HISTORIA DE USUARIO Nº 2	
Nombre de la Historia: Gestionar Inscripción con la pantalla touch Nextion	
Usuario: usuarios	
Prioridad de ejecución: Media **(Alta, Media, Baja)**	
Tiempo estimado: 5 minutos	**Iteración:** 1
Descripción: registrar una inscripción	
Observaciones: con poner el dni del usuario se inscriben en el único curso que se dista en ese momento.	

Tablas 5: Gestionar Inscripción con la pantalla touch Nextion

HISTORIA DE USUARIO Nº 3	
Nombre de la Historia: Acceso al sistema	
Usuario: usuarios	
Prioridad de ejecución: Alta **(Alta, Media, Baja)**	
Tiempo estimado: 3 minutos	**Iteración:** 1
Descripción: Los usuarios del sistema tendrán un nombre de usuario. Y la contraseña con la que podrán ingresar.	
Observaciones: solos los usuarios que estén registrados en la base de datos tendrán acceso al sistema.	

Tablas 6: Acceso al Sistema

HISTORIA DE USUARIO Nº 4	
Nombre de la Historia: Registro Responsable del Evento	
Usuario: usuarios docentes, usuarios alumnos	
Prioridad de ejecución: Media **(Alta, Media, Baja)**	
Tiempo estimado: 5 minutos	**Iteración:** 1
Descripción: La información requerida para el usuario de cada docente o responsable del curso/evento será extraído de la app móvil. Una vez cargada la información se guardara en la base de datos del sistema con sus respectivos usuario, permisos y funciones.	
Observaciones: Los usuarios solo podrán visualizar.	

Tablas 7: Registro Responsable del Evento

HISTORIA DE USUARIO Nº 5	
Nombre de la Historia: consultar el cupo del curso en la app móvil	
Usuario: usuarios	
Prioridad de ejecución: Media **(Alta, Media, Baja)**	
Tiempo estimado: 3 minutos	**Iteración:** 1
Descripción: El sistema deberá permitir al usuario saber el cupo disponible del evento	
Observaciones: El usuario podrá consultar el cupo de distintos eventos.	

Tablas 8: Consultar cupo en la app móvil

HISTORIA DE USUARIO Nº 6	
Nombre de la Historia: consultar el cupo del curso por la pantalla Nextion	
Usuario: usuarios	
Prioridad de ejecución: Alta **(Alta, Media, Baja)**	
Tiempo estimado: 7 minutos	**Iteración:** 1
Descripción: El sistema deberá permitir al usuario saber el cupo disponible del evento.	
Observaciones: El usuario podrá consultar el cupo del curso dictado en ese momento.	

Tablas 9: Consultar cupo en la pantalla Nextion

La presente lista describe los requerimientos de la aplicación móvil, los cuales más adelante serán especificados en prototipos de pantallas para el usuario final.

4.2 Análisis de requisitos

4.2.1 Requisitos Funcionales

- Registro de datos: El usuario deberá registrar los datos solicitados por la aplicación para tener acceso
- Ingreso a la aplicación: El usuario deberá iniciar sesión con su usuario y contraseña

- Mostrar lista de cursos: Presenta todos los cursos disponibles
- Consultar cupos disponibles del curso elegido: se podrá solicitar la inscripción siempre y cuando existiera el cupo del mismo
- Solicitar inscripción: Posteriormente a la elección del curso.

4.2.2 Requisitos No Funcionales

Los requisitos no funcionales para la aplicación, es decir, los que no especifican el comportamiento del sistema, son:

- La aplicación estará disponible el 100% del tiempo, ya que se trata de una aplicación nativa que se instalara en el dispositivo móvil.
- GUI: El diseño de la app será sencillo y ágil para el usuario
- La aplicación dependerá de una conexión a internet para él envió de información. En caso de no tener conexión a internet el usuario se podrá inscribir por la pantalla HMI.
- Portabilidad: compatibilidad de plataformas en el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0.
- Interfaz: Clara y concisa. No debe dar lugar a la confusión del usuario.
- La Aplicación trabajará con el administrador de base de datos Mysql.
- La aplicación guardará en una base de datos los registros de errores en tiempo de ejecución producidos durante todas las sesiones activas.
- El sistema de registro contará con manuales de usuarios para su entendimiento y capacitación en la herramienta.
- El servicio web de cursos, debe permitir al administrador la incorporación de nuevos cursos.
- La aplicación móvil debe alertar al usuario en el caso que las transacciones sean fallidas o exitosas.
- Para la aplicación móvil debe ser casi transparente la operación de conexión a los servicios web. Debe saber dónde se está dirigiendo (jsp, php, etc)
- La aplicación no presentará restricción en el número permitido de usuarios a registrarse, y no se darán de baja las cuentas de los usuarios que no sean utilizadas por un largo periodo de tiempo.
- Cada curso solo debe admitir un cupo de 60 asistentes al curso, detallando al momento de inscribirse el cupo actual.
- Rendimiento: el tiempo de respuesta promedio de la app no debe superar los 10 segundos.

- Seguridad: Es requisito de conexión, verificar que sea un usuario creado para la aplicación por lo que consulta directamente en la Base de Datos para acceder a la app)

Fase de Diseño

Se elaboraron esquemas breves que sirven de referencia para la ejecución. En la Figura 15 se tiene un esquema general del control de asistencia.

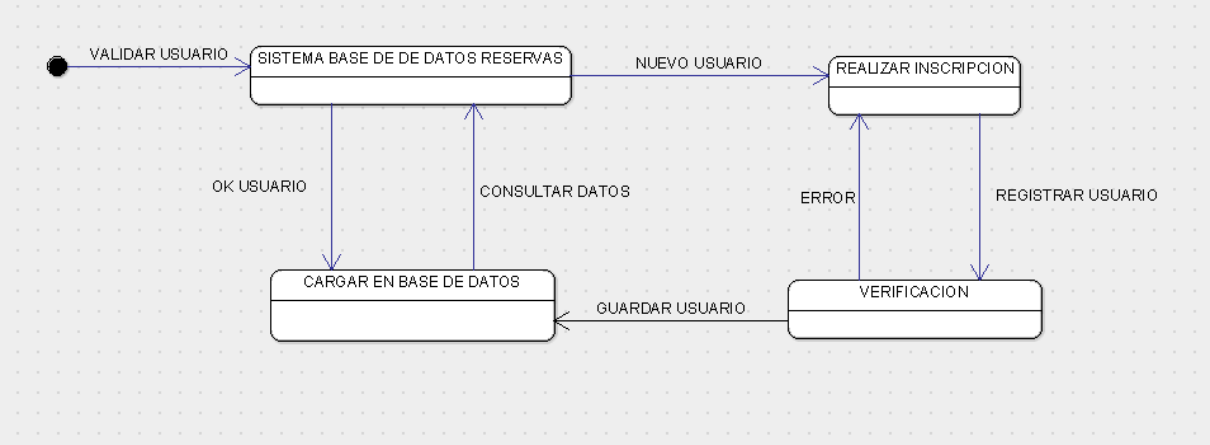


Figura 15: Diagrama UML. Prototipo del sistema de control de asistencia

Para esta fase se empleó el modelado UML ya que se puede usar para modelar distintos tipos de software.

UML ofrece varios diagramas en los cuales modelar sistemas. Mediante la herramienta ArgoUML (Versión 0.34), los siguientes diagramas, más relevantes son:

- Diagramas de Casos de Uso
- Diagramas de Clases

4.3 Casos de uso

Los casos de uso se utilizan para especificar el comportamiento deseado de un sistema o subsistema. Su especificación describe el conjunto de secuencias de acciones que se lleva a cabo en el sistema para producir un resultado para un actor. También, capturan el comportamiento deseado del sistema, sin especificar cómo se lleva a cabo dicho comportamiento.

Un actor representa un conjunto coherente de roles que los usuarios de los casos de uso representan al interactuar con el sistema (tipos de usuarios). Normalmente representan a una persona, un dispositivo hardware u otro sistema al interactuar con el nuestro, como muestra la Figura 16.

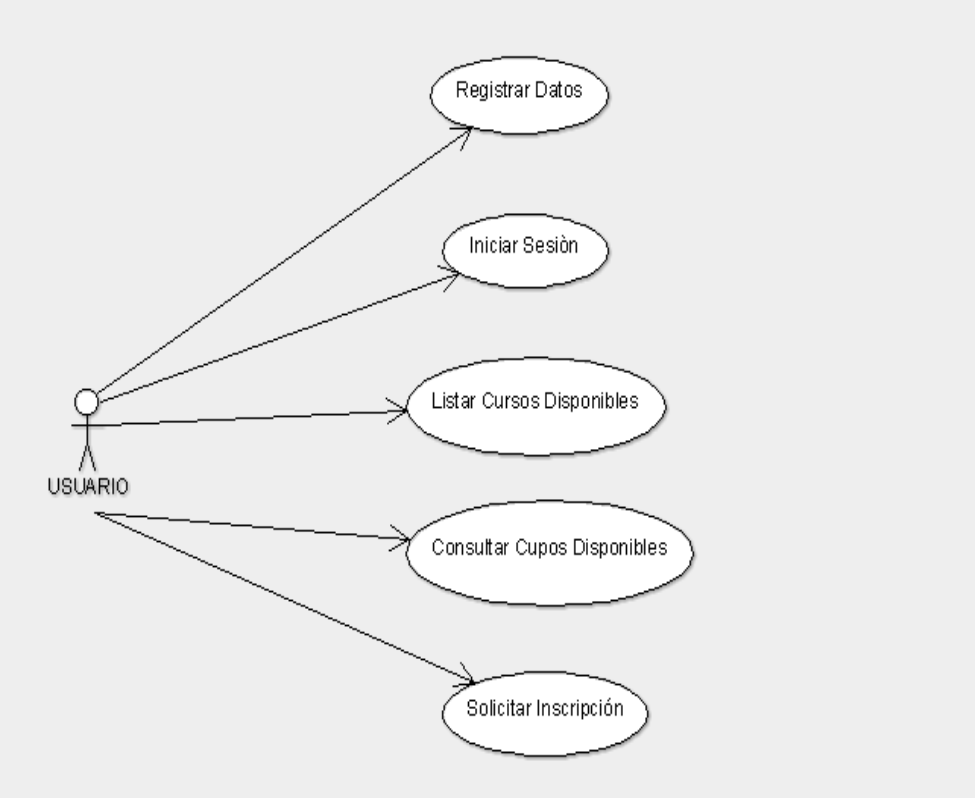


Figura 16: Diagrama de Casos de uso de la Aplicación móvil

Especificación de casos de uso (Ver Tabla 10):

“CASO DE USO GENERAL”	
Nombre	Diagrama de la Aplicación
Actor	Usuario
Descripción	-El usuario para usar la aplicación debe registrar sus datos. -Para iniciar sesión ingresar a la aplicación con un usuario y contraseña válidos. -Visualizar la información de los cursos y realizar la

	inscripción.
Precondición	-La aplicación debe estar instalada en un Smartphone con plataforma Android y tener acceso a internet. -Conexión a la base de datos satisfactoria
Actividades	-Registrar Datos: Registro de la información del usuario previo ingreso a la app -Iniciar Sesión: Consulta en la base de datos la existencia del usuario con el ingreso de usuario y contraseña para acceder a la aplicación -Visualizar y Listar cursos: Presentación de cursos disponibles con su respectivo detalle: imagen, título y fecha. -Consultar Cupo: Consulta en la base de datos la existencia de los cursos solicitados. -Solicitar inscripción: Asignar la inscripción en la relación usuario/curso de dicho curso elegido.
Flujo del Sistema	Si el usuario ingresa mal su nombre de usuario y/o clave, podrá intentar cuantas veces desee.
Post condición	Si la aplicación detecta conexión a internet envía la inscripción caso contrario se pierde el pedido.

Tablas 10: Descripción caso de uso general de la app

4.4 Diagrama de Clases

El diagrama de clases que se mostrará a continuación (Ver Figura 17) está directamente relacionado con las clases que se utilizaron en la aplicación móvil, ya que para desarrollar la misma fue necesario crear las propias tablas en la base de datos.

Es importante recalcar que las tablas y clases son las mismas que existen para la pantalla HMI Nextion, con la diferencia que se agrega la tabla: tarjetas.

En el siguiente diagrama las clases con sus respectivos atributos y como están relacionadas unas con otras.

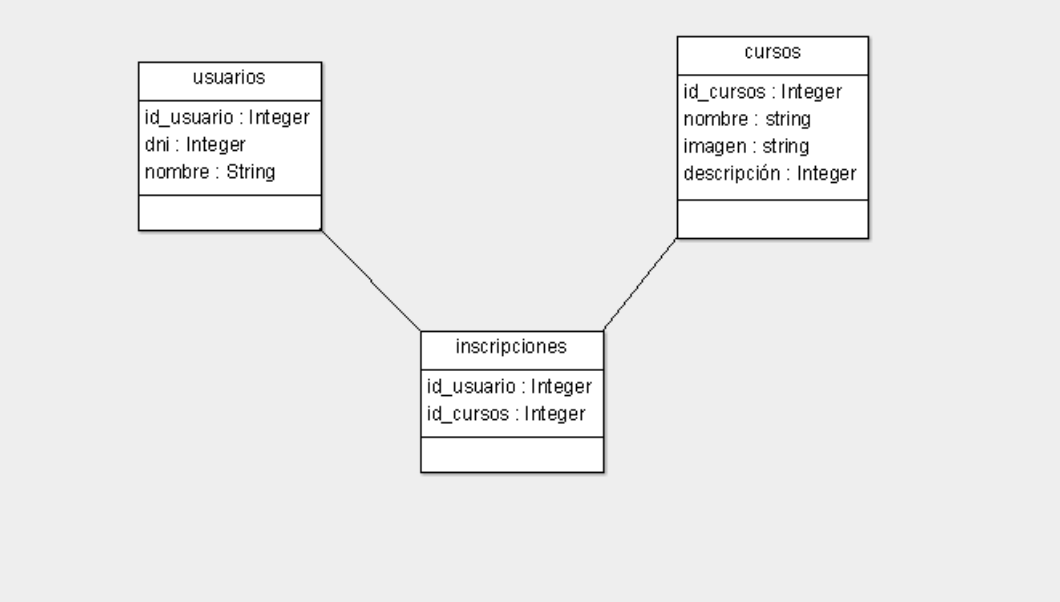


Figura 17: Diagrama de clases de la aplicación para la inscripción a la app y cursos

4.5 Análisis de Tecnologías

Para el desarrollo de la aplicación se utilizó el lenguaje de programación Android, basado en otros lenguajes de programación como ***Java*** y ***C++.***

Android Studio es el entorno de desarrollo integrado (por sus siglas *IDE*) oficial para el desarrollo de aplicaciones para *Android* y se basa en ***IntelliJ IDEA***. Además del potente editor de códigos y las herramientas para desarrolladores de *IntelliJ*, *Android Studio* ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android. Es elegido como entorno de desarrollo ya que contiene los paquetes necesarios para el desarrollo de aplicaciones móviles basados en ***Android.v***, además cuenta con soporte y actualizaciones para desarrollar aplicaciones para nuevas versiones de Android.

4.5.1 Características técnicas de Android

Android, por el hecho de haber sido creado en base al Kernel Linux, es gratuito lo que permite realizar variadas actividades con él, sin incurrir en costos de licencias. Entre las bondades que ofrece se encuentran:

- Crear aplicaciones utilizando una amplia gama de útiles bibliotecas y herramientas que pueden ser utilizadas para construir aplicaciones variadas.
- Está construido en código abierto lo que permite ser ampliado para incorporar nuevas tecnologías si los usuarios así lo desean.
- Ofrece una máquina virtual personalizada que ha sido diseñada para optimizar la memoria y los recursos de hardware en un entorno móvil.
- La plataforma Android continuará evolucionando a medida que la comunidad de desarrolladores trabajen para crear aplicaciones móviles.
- Permite ofrecer a los usuarios un amplio espectro de aplicaciones y servicios ya que cualquier persona puede desarrollarlas libremente.
- Ofrece un Framework de aplicaciones que se está habilitando para la reutilización y el reemplazo de componentes.
- La máquina virtual Dalvik, optimizada para dispositivos móviles.
- Navegador integrado, basado en el motor del proyecto abierto WebKit.
- Gráficos optimizados, suministrados por una biblioteca de gráficos 2D. Los gráficos 3D están basados en la especificación OpenGL ES 1.0, con soporte para aceleración gráfica por hardware (opcional).

4.6 Filosofía de desarrollo Android: Modelo Vista Controlador (MVC)

4.6.1 Patrones y arquitectura

Los patrones para el desarrollo utilizados en la aplicación están basados en la arquitectura de programación en capas a dos niveles. Este estilo de programación está orientado al desarrollo de componentes o clases que se encarguen del tránsito, transformación y presentación de los datos en la aplicación; de esta forma se divide el código para hacerse más limpio y legible.

Dentro de la capa de datos se encuentra la lógica que permite el acceso a los datos, es decir, conexiones o maneras de obtener información del servidor que se encuentran

codificadas en estas clases. En la capa de negocio, se manejan las transacciones entre capas, esta suele llamarse lógica de negocio, ya que en ellas se encuentran las directrices del tránsito y muchas veces la transformación de los datos.

Por último, y no menos importante se encuentra la capa de presentación, dicha capa está encargada de mostrar los datos a los usuarios, controlando acciones en las interfaces de usuario, solicitando peticiones a la capa de negocio para logar interactuar con el sistema y la base de datos.

Esta arquitectura es muy parecida a la utiliza en entornos web llamada *MVC* (Modelo – Vista – Controlador). Cabe destacar que la arquitectura que presenta *Android Studio* para programar sobre *Android* es un *MVC*. Por un lado existen los *XML* de la capa de presentación o vista y está regida por clases *java* que permiten manejar el negocio o el controlador.

MVC (Ver figura 18) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, los cuales se detallan a continuación.

Modelo: Sirve como representación específica de toda la información con la cual el sistema va a trabajar.

Vista: Presenta el modelo con el que interactúa el usuario, más conocida como interfaz.

Controlador: El controlador responde más bien a eventos, normalmente son acciones que el usuario invoca, implica cambios en el modelo y también en la vista (interfaz).

El uso de este patrón está fuertemente impulsado por Android, ya que esta plataforma utiliza esta modalidad y, al momento de crear un proyecto, por defecto se separan en tres capas.

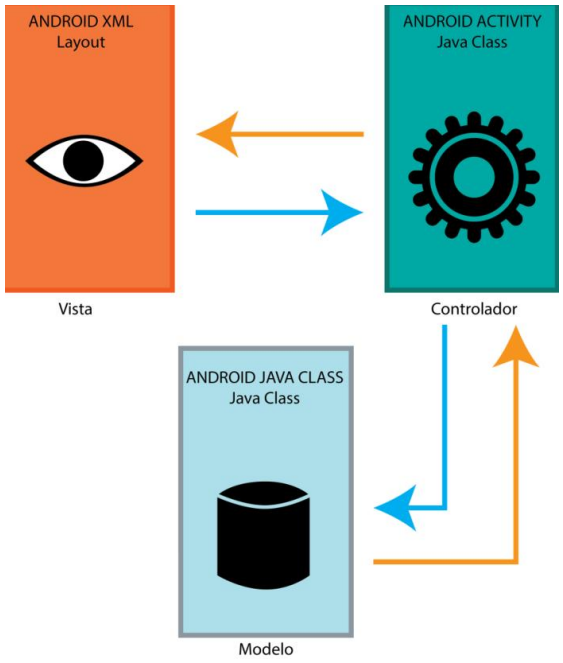


Figura 18: modelo vista controlador [18]

Android sigue el patrón de diseño modelo vista controlador (MVC), la cual separa los datos de la aplicación, la interfaz de usuario y la lógica del negocio en tres componentes distintos.

Al crear una aplicación en Android automáticamente se crearán 3 carpetas la cuales permiten seguir el orden del modelo de diseño descrito anteriormente, las carpetas son las siguientes:

Carpeta src: Es el espacio en donde se crea toda la lógica del negocio. Aquí se ubican los archivos Java que implementa a las vistas de la aplicación (Activity).

Carpeta gen: Contiene un único archivo llamado R.java. Este archivo sirve básicamente para enlazar las tareas que se hacen en XML con la programación en Java.

Android Library: Aquí se encuentran todas las librerías propias del SDK de android, dependiendo la versión elegida al crear el proyecto tendrá una versión u otra.

Carpeta assets: Aquí se incluyen los archivos varios que puede ser música, pdf, zip, rar.

Carpeta res: Dentro de esta carpeta se encuentra el archivo AndroidManifest.xml. Todas las aplicaciones deben tener este archivo y no debe ser renombrado. En él se especifican las opciones generales del programa, como el paquete principal, la actividad que deberá ejecutarse al iniciar la aplicación (y deben incluirse allí todas las actividades que se van usar), el icono a usar, los permisos, etc, además de este archivo hay tres subcarpetas las cuales guardan todo lo necesario para la interfaz, estas carpetas son:

- ***Carpeta Drawable***: Contiene todos las figuras y dibujos que se utilizarán en la aplicación.
- ***Carpeta layout***: Contiene las Activitys que presentan una interfaz gráfica, permite al usuario interactuar con la aplicación. Estas Activitys están escritas en XML.
- ***Carpeta Value***: Está contenido inicialmente el archivo strings.xml, en el que se declara las cadenas de texto que usará la aplicación.

4.7 Base de Datos

Este trabajo final utilizo una base de datos relacional MySQL, la elección de este tipo de base de datos es por las siguientes ventajas:

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece a la normalización por ser más comprensible y aplicable.
- Rendimiento: Alta velocidad de acceso y conexión (mayor que sqlite3).
- Seguridad.
- Open Source.
- La migración a otra máquina no tiene mucha dificultad, es compatible con la mayoría de sistemas operativos y su acceso y su conexión es mejor que el SQlite que proporciona Android.

Las tablas utilizadas son las siguientes:

Usuarios: como vemos en la Figura 19, aquí se almacena todos los datos necesarios para identificar a un usuario.

```
SELECT * FROM `usuarios` ORDER BY id_usuario ASC
```

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave:

Opciones

	id_usuario 1	dni	pass	nombre	serie
Editar Copiar Borrar	1	16759014	4004	Miriam Barros	D526D443
Editar Copiar Borrar	2	31814063	123456	Noelia Barrionuevo	2DD6C6DB
Editar Copiar Borrar	3	3248599	1111	Hilda Villalba	50321333
Editar Copiar Borrar	4	13141823	1234	Oscar Barrionuevo	50321333
Editar Copiar Borrar	5	42371791	55555	Sol Rios	04733263

Figura 19: Tabla MySql "usuarios"

Cursos: Esta tabla almacena el nombre de los cursos, en la misma se lista en la aplicación los cursos a los que pertenece cada usuario, ver Figura 20.

```
SELECT * FROM `cursos`
```

Perfilando [Editar en línea] [Editar] [Explicar SQ

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones

	id_cursos	nombre	imagen	descripcion
Editar Copiar Borrar	1	Curso De Programación	http://educared.fundaciontelefonica.com.pe/wp-cont...	16 de junio 2019 Curso gratuito: Introducción a l...
Editar Copiar Borrar	2	Curso SIU WICHI	http://sti.uncoma.edu.ar/images/siu.gif	Taller de práctica y capacitación para usua
Editar Copiar Borrar	3	Charla "Mineria de datos"	https://conceptodefinicion.de/wp-content/uploads/2...	17 y 18 de agosto 2019
Editar Copiar Borrar	4	Taller de Migracion Guarani 3	http://noticias.universia.com.ar/net/images/educac...	Destinado a docentes de la FTyCA
Editar Copiar Borrar	5	Taller de Robotica	https://www.createc3d.com/wp-content/uploads/2014/...	En una primera instancia se capacitará a e

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Figura 20: Tabla MySql "cursos"

Inscripciones: Esta tabla (Figura 21) se utiliza para ver la relación rápidamente entre los usuarios y sus cursos.

```
SELECT * FROM `inscripciones` ORDER BY `id` ASC
```

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

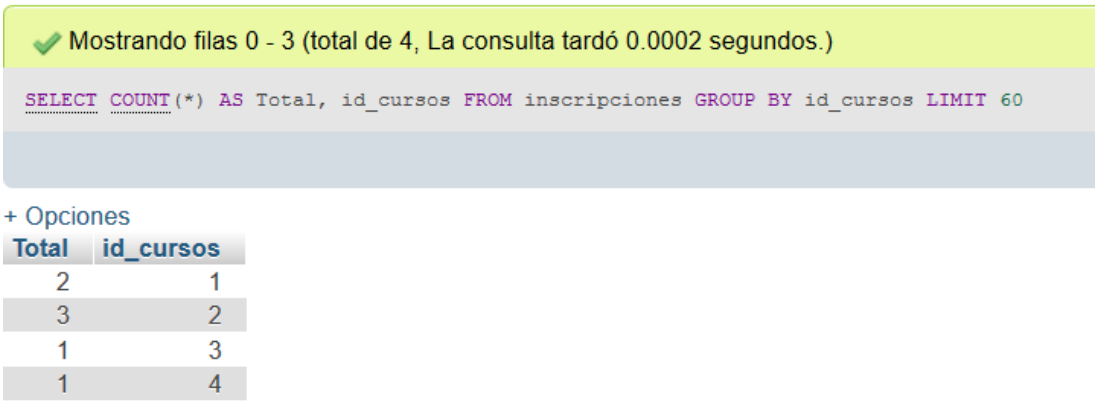
+ Opciones

	id	1	id_usuario	id_cursos
Editar Copiar Borrar	1		6	1
Editar Copiar Borrar	2		1	4
Editar Copiar Borrar	3		4	3
Editar Copiar Borrar	4		3	2
Editar Copiar Borrar	5		4	2
Editar Copiar Borrar	6		1	1
Editar Copiar Borrar	7		6	2

Seleccionar todo *Para los elementos que están marcados:* Editar

Figura 21: Tabla MySql "inscripciones"

En la Figura 22 está la consulta, que cuenta las inscripciones de cada curso. Hasta 60 cupos por curso.



Mostrando filas 0 - 3 (total de 4, La consulta tardó 0.0002 segundos.)

```
SELECT COUNT(*) AS Total, id_cursos FROM inscripciones GROUP BY id_cursos LIMIT 60
```

+ Opciones

Total	id_cursos
2	1
3	2
1	3
1	4

Figura 22: Consulta cupo actual

4.8 Comunicación son los Servicios Web

Para este proyecto se han utilizado dos servicios web.

El primero se utilizó para el Registro, Login y Consultas de los usuarios. Para gestionar la comunicación con este servicio es necesario incluir un fichero JSON en Android Studio para activar y mantener la sesión en el momento que sea necesario, con la siguiente lista de los recursos a usar:

- Hosting: Remota. Haremos pruebas que nos faciliten la implementación del servicio web.
- Lenguaje: PHP para la API y Java para Android.

- Base de datos remota: MySQL
- Formato de intercambio de datos: JSON

El segundo servicio web albergo todos los registros de los cursos, su actualización y mantenimiento. Hostinger es un servicio de hosting con soporte PHP y MySQL.

Estos servicios se han realizado para comunicar la base de datos que se encuentra en el servidor con la aplicación, creado en un servidor “vibadent”. Cada clase PHP creada es un método que ofrece un servicio en el cual se llamará a la base de datos para realizar inserciones o consultas dependiendo de las necesidades.

En la siguiente imagen se ve como es la estructura del Servicio Web. (Ver figura 23)

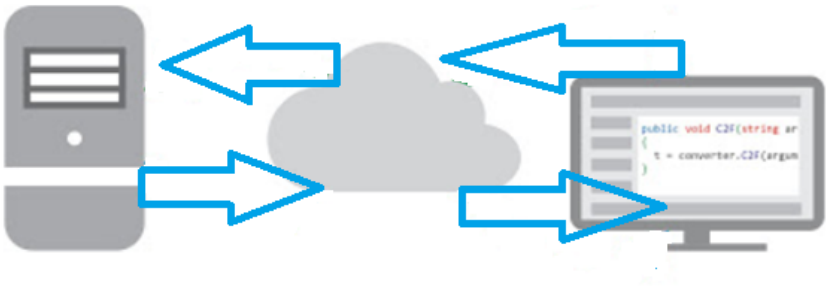


Figura 23: Estructura del Servicio Web

4.9 Diseño de la interfaz de la aplicación

La interfaz se ha diseñado con el propósito de que sea intuitiva y simple, pero a la vez potente y capaz de cubrir todos los requisitos enumerados en el apartado de requisitos. A continuación las pantallas han sido diseñadas con el propio Android Studio (Ver Figura 24).

En ellas se puede observar un diseño de maquetas de la aplicación. En primer lugar, si el usuario no se encuentra con una sesión activa, la aplicación pide un inicio de sesión, dando opción del mismo modo a registrarse en el sistema, en el caso de que no tuviera un usuario creado.

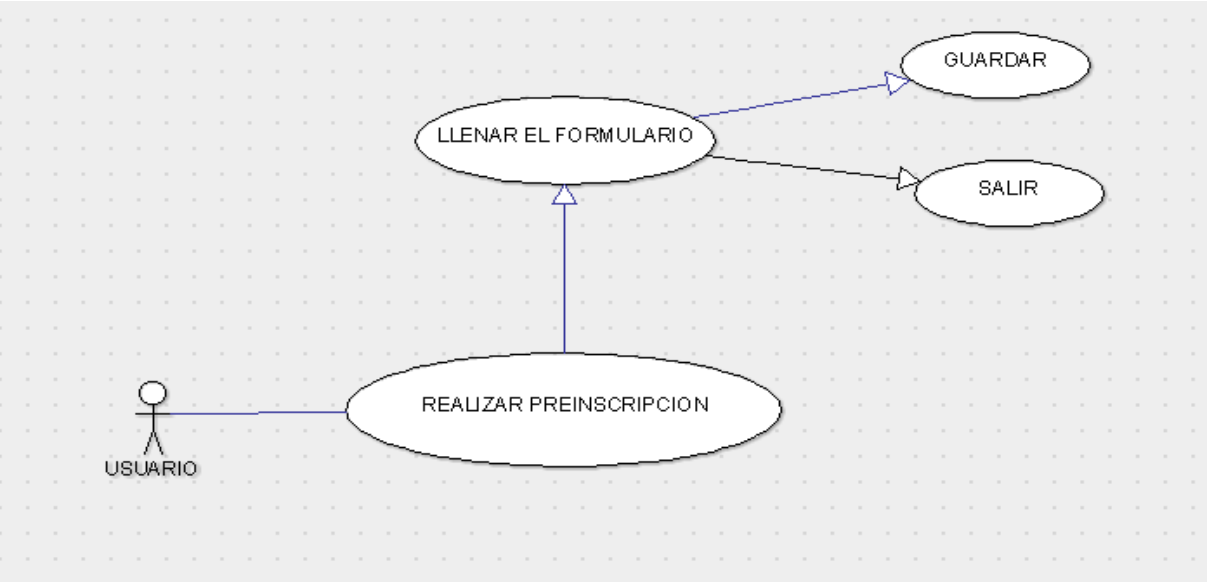


Figura 24: Prototipo de la preinscripción realizada en la aplicación móvil

En el menú de aplicaciones del teléfono móvil se visualiza el logo de la app SCARAA para realizar el registro e inicio de sesión de la pre inscripción. Ver figura 25.

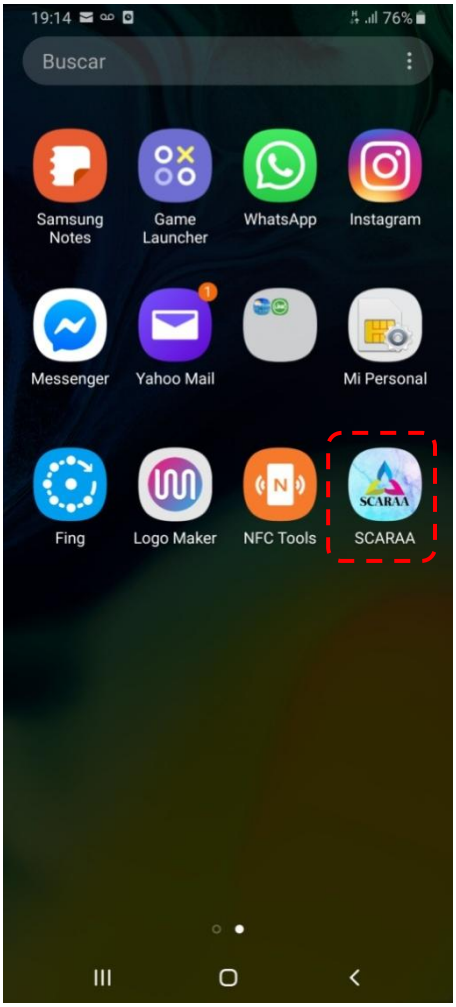


Figura 25: Interfaz de logo de la app móvil (Auditorio) instalada en el celular del usuario

Inicio de sesión

Un usuario es el actor principal en la aplicación. Su interacción con la aplicación comienza al iniciar sesión. La actividad principal de la aplicación, permite acceder con el DNI y una contraseña (Ver Figura 26). Si este no se encuentra, avanza a la actividad de registrarse en el Figura 27.

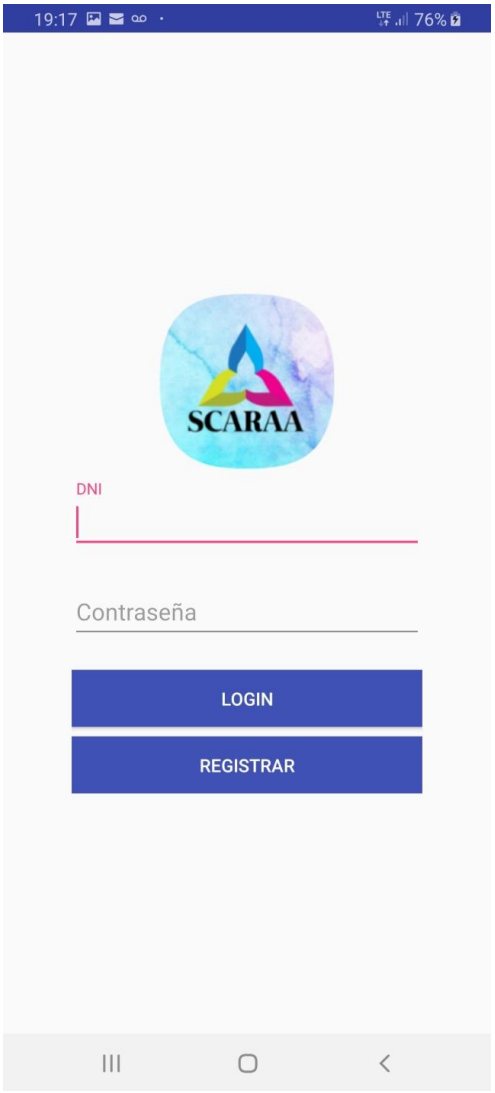


Figura 26: interfaz Iniciar Sesión

En la siguiente Figura 27, el usuario ingresa una serie de campos obligatorios, con los parámetros necesarios para poder registrarse.

Figura 27: interfaz Registro de usuarios

En la Figura 28 ingresando el Loggin del usuario y la contraseña la app realiza una Autenticación de los datos.

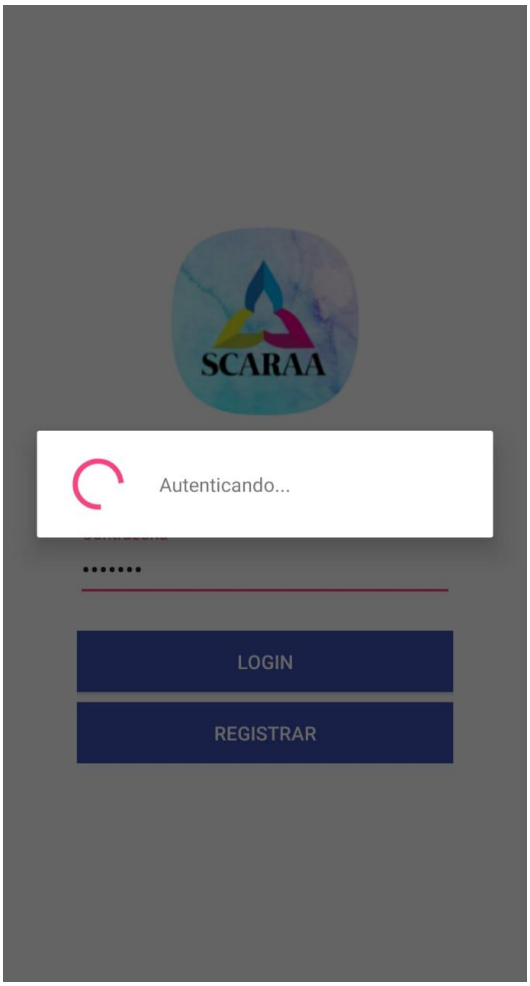


Figura 28: Interfaz autenticar datos

Si por el contrario se produce un error, se mostrar un mensaje en la pantalla del teléfono móvil como en la Figura 29.

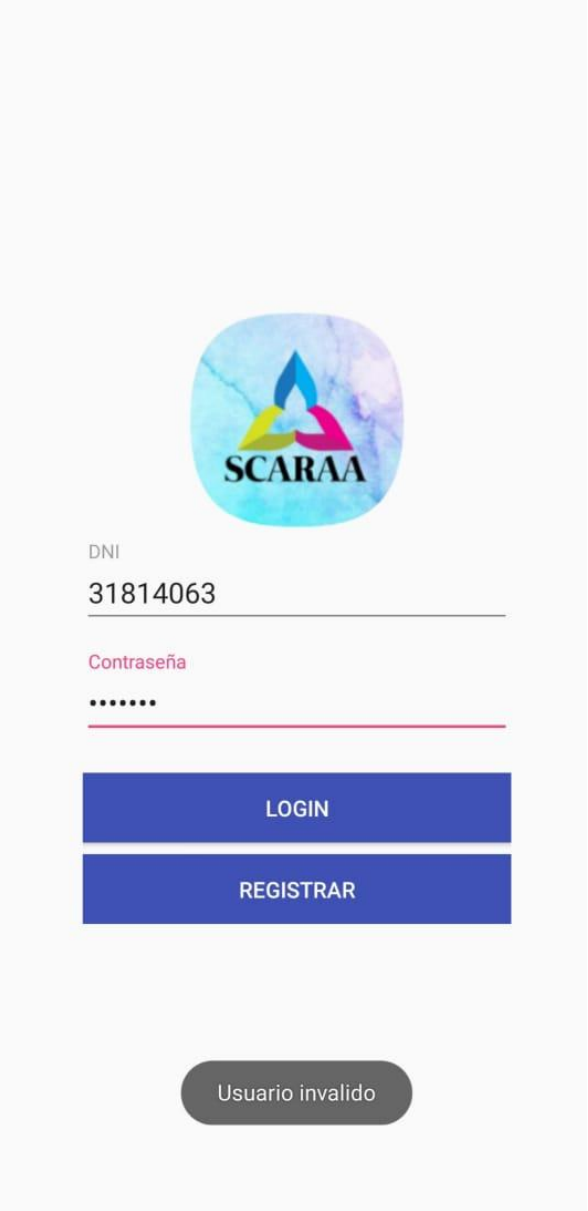


Figura 29: interfaz, usuario inválido

Si el usuario se encuentra registrado, al realizar un click en el botón de INGRESAR se ejecutara la siguiente pantalla con la lista de los cursos disponibles. (Ver figura 30).

Listado de cursos

La actividad carga de la base de datos todos los cursos disponible mostrándose en la interfaz cada uno de ellos, seleccionando en la pantalla el que se desea listar nos permite mostrar el detalle del mismo.

Figura 30: Interfaz Listado de cursos

Registro de inscripción

En la Figura 31 se detalla los datos del curso: nombre, una pequeña descripción del mismo, día y horario, disertante, lugar y alguna observación.

Figura 31: Detalle del curso seleccionado. Interfaz confirmar inscripción en el curso

CAPITULO V

5 DISEÑO DEL PROTOTIPO

5.1 Introducción

Para el presente trabajo final se ha diseñado un prototipo que cumple con las características presentadas en el diseño del sistema de Control de Acceso y Registro Automático de Asistencia, en la que se demuestra que utilizando la plataforma Intel Galileo como hardware se puede realizar de manera satisfactoria este sistema. En la Figura 32, se observa el diagrama de funcionamiento gráfico con los elementos del diseño del sistema electrónico NFC, los cuales indican la comunicación entre ellos.

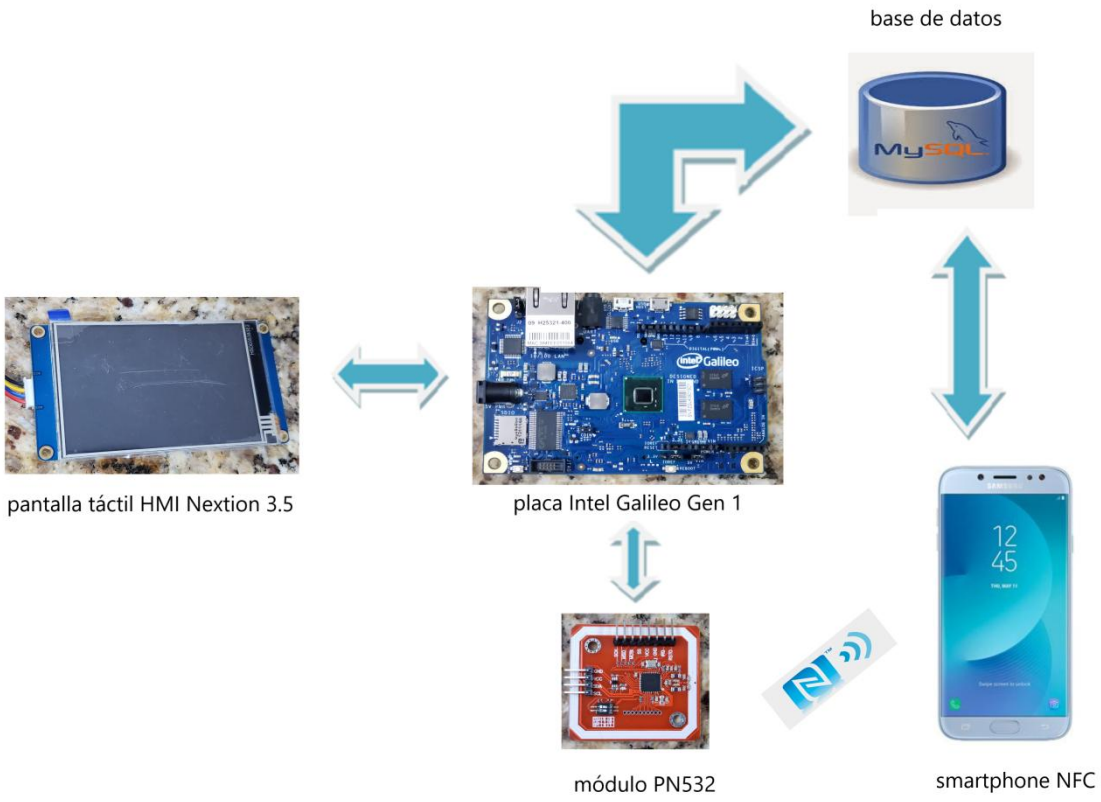


Figura 32: Diagrama gráfico del sistema SCARAA

Este prototipo cuenta con dos partes importantes: un módulo electrónico en la que los usuarios por medio de una interfaz interactiva en una pantalla táctil HMI pueden registrarse, y un módulo NFC donde registre la inscripción al curso por medio de la lectura del Smartphone. Paso previo al usar la app móvil y presentarse en el curso, con la oportunidad de asegurar el lugar con el registro de cupos, tendrá una interfaz muy parecida a la pantalla

HMI para poder controlar y manipular la información de los usuarios y los cursos y registrados en la base de datos.

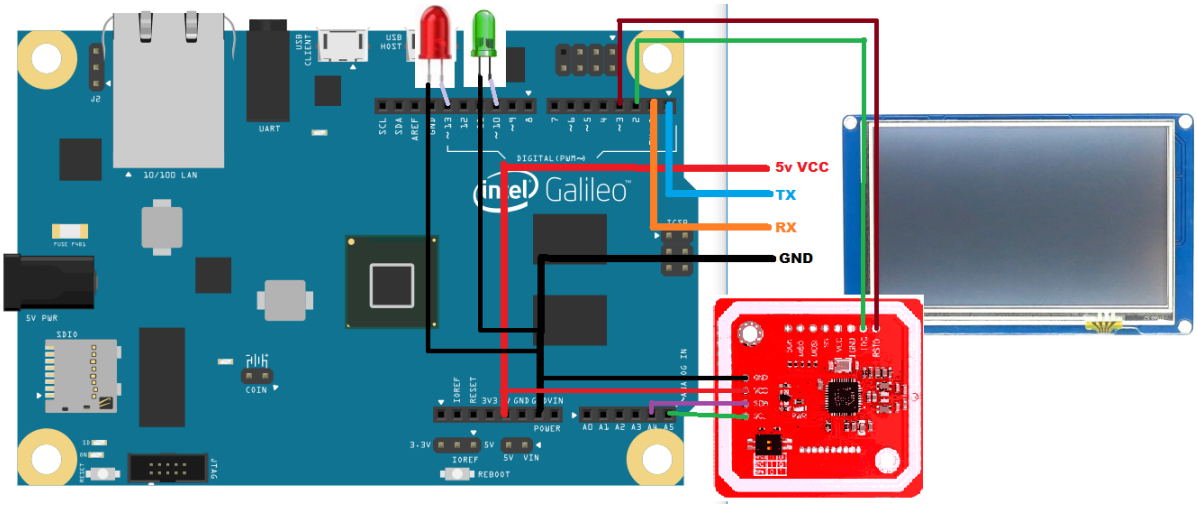


Figura 33: Esquema general del sistema de control y registro

La Figura 33, indica el proceso de conexión entre los diferentes elementos que constituyen el proyecto que son descritos a continuación:

- Intel Galileo se interconecta con el módulo NFC mediante un protocolo de comunicación serie síncrono I2C.
- Intel Galileo se conecta con la pantalla HMI para el envío y recepción de datos mediante un protocolo de comunicación asíncrono UART.

5.2 Modo de Operación del Sistema

5.2.1 Automático por el lector NFC

En el modo automático, el lector se encuentra realizando lecturas de manera periódica indefinidamente. En el momento en que un teléfono móvil o un tag entre en su rango de lectura (entre 5 a 10 cm.), éste será leído. El problema de este modo de operación por el cual no pudo ser la única solución es que algunos teléfonos no poseen la tecnología NFC. (Ver Figura 34)

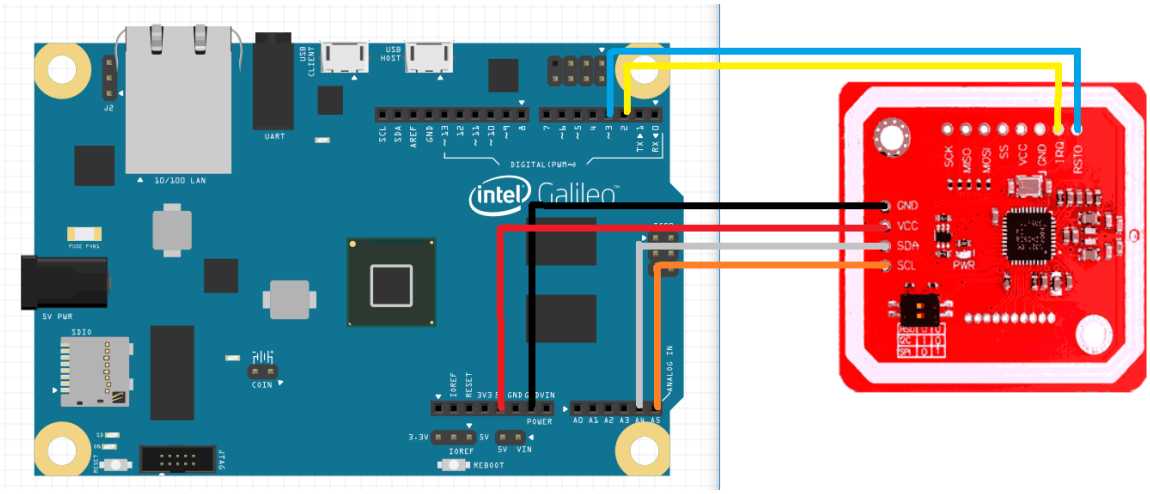


Figura 34: Circuito de Conexión de Intel Galileo gen1 con el Shield NFC/RFID pn532

5.2.2 Por medio de la pantalla HMI

En el modo “pantalla”, el usuario se registra en el curso actual y automáticamente se actualiza en la base de datos. Este esquema, permite evitar las colisiones en una red, ya que el host puede comunicarse con los lectores, uno por uno, evitando que estos transmitan al mismo tiempo. (Ver Figura 35)

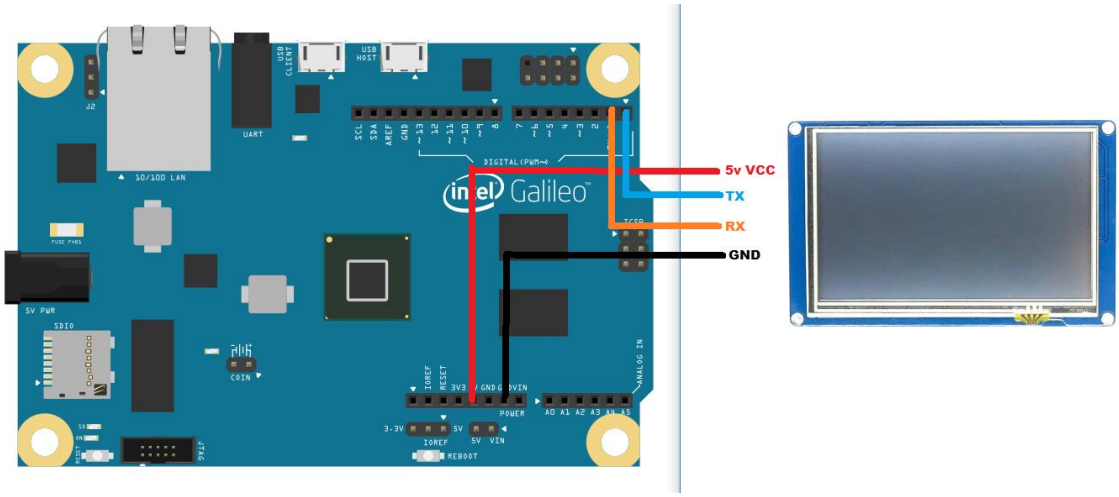


Figura 35: Circuito de Conexión de Intel Galileo gen 1 con pantalla HMI Nextion 3.5

5.3 Diagrama de bloques del sistema electrónico NFC

En la Figura 36, se observa el diagrama de bloques del sistema electrónico el cual está constituido por el Smartphone, la placa Intel Galileo gen 1, el módulo NFC y el software de gestión (app móvil).

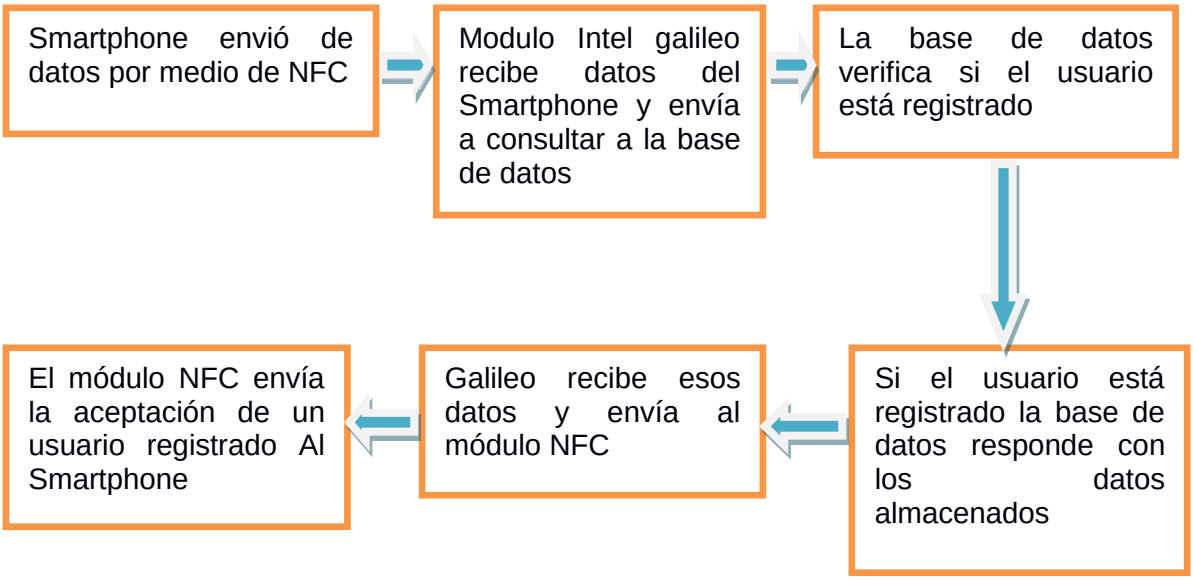


Figura 36: de bloques del circuito electrónico NFC

En la app móvil, el usuario se registra por primera vez con sus datos (nombre, usuario y contraseña) y esa información se envía a la base de datos. Si el usuario ya se registró alguna vez en la app, inicia la sesión con los datos requeridos (usuario y contraseña) para consultar en la base de datos su correcta existencia.

Al visualizar el listado de cursos disponibles, elige uno de ellos y la app le informa si existe cupo del mismo. Si lo hay el usuario puede inscribirse en él, de lo contrario le envía un mensaje haciéndole saber que no podrá ser efectuada su inscripción debido al cupo insuficiente.

El Smartphone se encargará de enviar la información, una vez registrado el usuario en la app móvil, por medio del lector NFC y la placa Intel Galileo recepta los datos y envía a la base de datos, la cual verifica si el Smartphone pertenece a un usuario registrado. Estos datos son trasmitidos hacia la placa Intel Galileo la cual emite una validación hacia el Smartphone por medio del módulo NFC.

5.4 Firmware

El dispositivo de seguimiento tiene un software de propósito específico, o firmware, grabado en la memoria flash del microcontrolador, cuyas principales funciones son:

- Comunicación y conexión Galileo + pn532 por I2C;
- Comunicación y conexión Galileo + pantalla HMI;

- Envió de información a la base de datos en mysql;
- Control de la conexión y transmisión de datos;

El firmware es imprescindible es este sistema de control y registro de asistencia. La utilización de un dispositivo electrónico que captura los datos y necesita de un nexo que permita la recopilación de los datos y su posterior almacenamiento dentro del sistema SCARAA. Es por ello que surge la necesidad de desarrollar un sistema embebido (firmware) que permita realizar este procesamiento.

Son bloques de instrucciones de programas para propósitos específicos, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de dispositivos de cualquier tipo.

Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación. Funcionalmente, el firmware es el intermediario (interfaz) entre las órdenes externas que recibe el dispositivo y su electrónica, ya que es el encargado de controlar a ésta última para ejecutar correctamente dichas órdenes externas.

Para el inicio de la programación, es necesario realizar un diagrama de flujo el cual contiene una secuencia lógica de procesos a seguir que son descritos en el siguiente subtema. Como lo muestra la Figura 37 a continuación.

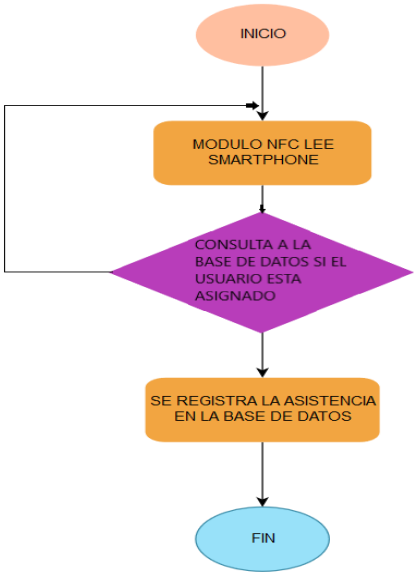


Figura 37: Flujo grama de funcionamiento interno del sistema electrónico NFC

La secuencia de pasos descritos anteriormente en el flujograma del módulo Intel® Galileo + pn532 NFC, describe el proceso de creación del código de programación el cual está disponible en el ***Anexo A (página 92).***

5.4.1 Programación de la placa embebida Intel® Galileo Gen 1

Para el desarrollo de la programación de la placa Intel Galileo es necesario tener previamente instalado el software de programación propio de Arduino el cual se presenta a continuación en la Figura 38.

El proceso de descarga del IDE de Arduino se puede realizar desde su página oficial mediante el siguiente link http: //arduino.cc/es/Main/Software donde se tendrán opciones de descarga dependiendo del tipo de PC ya sea Windows, Mac, o Linux. Una vez ejecutado e instalado el archivo Arduino.exe en modo administrador, prácticamente se procede al reconocimiento del modelo de tarjeta Arduino con su puerto serie asignado al controlador Arduino.

A continuación se realiza la elección del módulo Intel Galileo en el IDE para comenzar la programación. (Ver Figura 38)

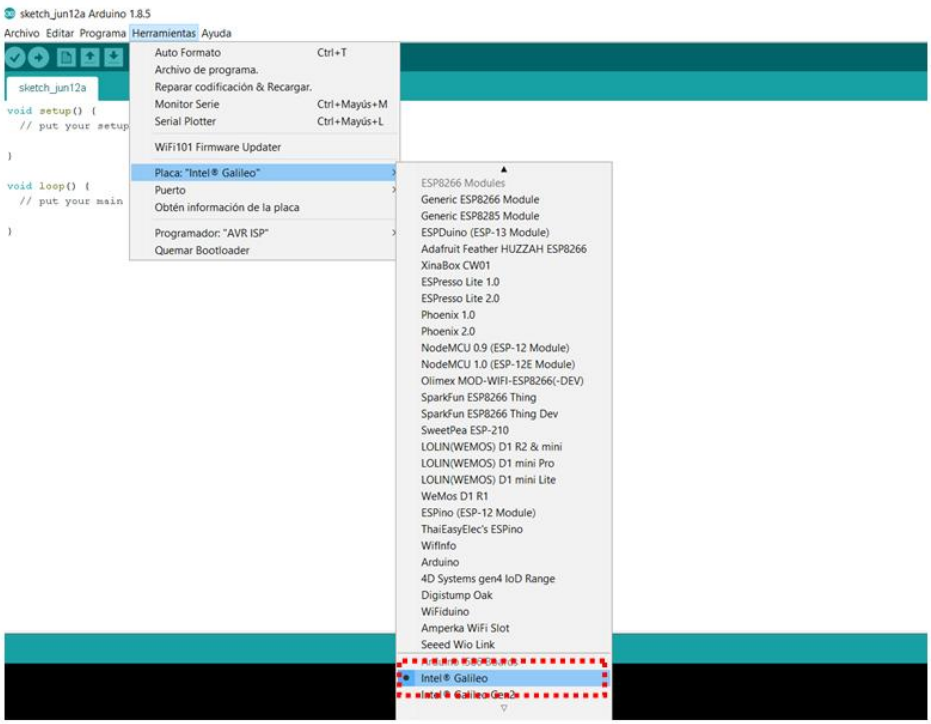


Figura 38: IDE elección de la placa Intel Galileo Gen 1

5.4.2 Diseño y Programación de la interfaz HMI

Visualización

Los datos adquiridos por el dispositivo son visualizados en una pantalla HMI y son enviados a una base de datos, para ser presentados en el servidor con el análisis pertinente de acuerdo las tablas y rangos establecidos. (Ver Figura 39).

La programación de conexión se realiza en el IDE de Arduino ***Anexo B (página 96).*** Y la interfaz gráfica se realiza en el IDE de Nextion.

El proceso de descarga del IDE de Nextion puede realizar desde su página oficial mediante el siguiente linkhttps://nextion.itead.cc.

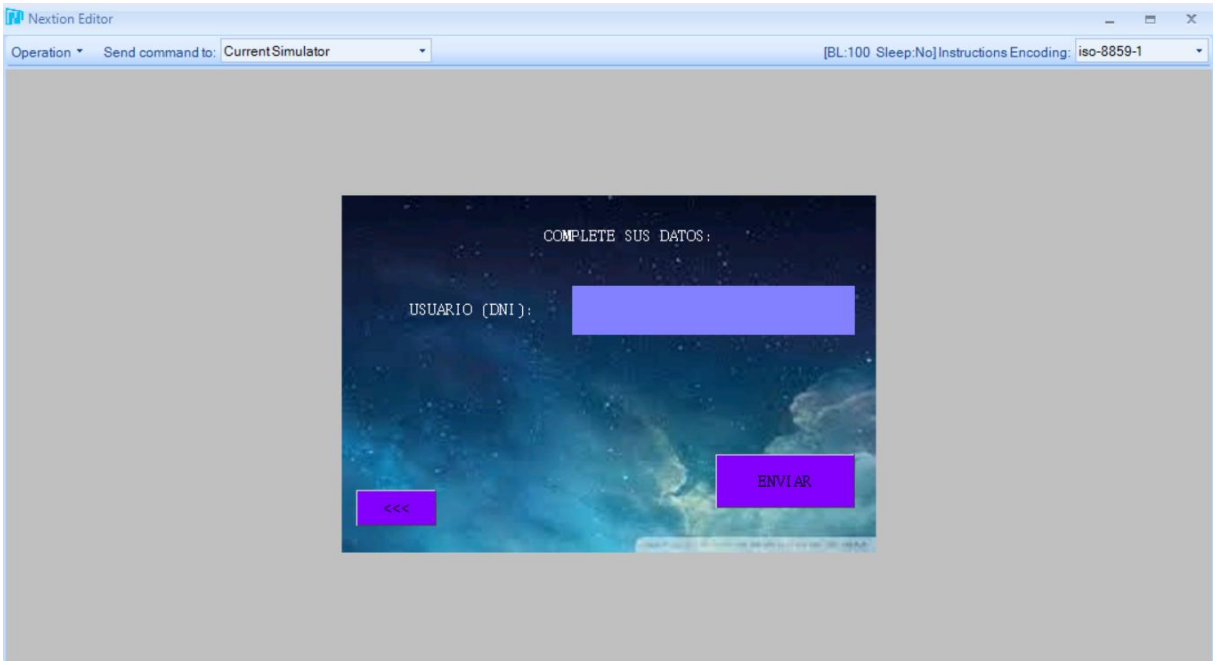


Figura 39: Pantalla HMI prototipo

CAPITULO VI

6 PRUEBAS

6.1 Introducción

La metodología XP sugiere que antes de entregar cualquier producto para su uso, es necesario realizar pruebas las cuales identifiquen posibles errores que este pueda tener. Estas pruebas son unitarias, y a corto plazo de esa manera se van cumpliendo las tareas esenciales de la app. Las pruebas tienen como objetivo principal proporcionar información objetiva e independiente sobre la calidad del producto a los usuarios finales, dicho esto, en este capítulo se detallarán las pruebas realizadas a las diferentes funcionalidades tanto de la aplicación móvil, NFC, la pantalla HMI y el servicio web.

6.2 Pruebas de funcionamiento de SCARAA

Para verificar el funcionamiento del prototipo del sistema para el control y registro de asistencia se ejecutó pruebas con 2 alternativas para el usuario.

6.2.1 Alternativa 1º

En la primera alternativa el usuario se registra en la app móvil colocando sus datos: DNI, Nombre, Contraseña, Número de UID (número de serie que identifica una tarjeta inteligente, llavero o un Smartphone).

El usuario selecciona el curso al cual desea inscribirse, y la app consulta a la base de datos la disponibilidad de cupo del mismo (Ver Figura 40).

- Si hay cupo, lo inscribe correctamente.
- Si no hay cupo, le muestra un mensaje advirtiendo.
- Si ya se inscribió, también le muestra un mensaje.

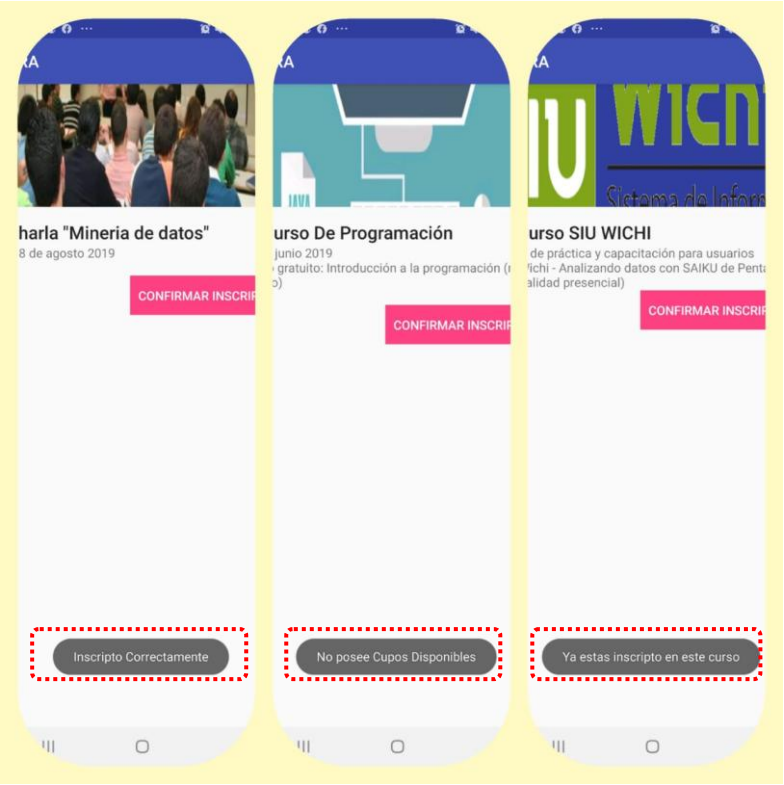


Figura 40: Mensajes de situación de disponibilidad de los cursos

Para confirmar la asistencia el usuario se presenta el día y horario del curso en el lugar pactado, se acerca al dispositivo que contiene el NFC y con su tarjeta, llavero o teléfono celular (el cual registro su UID en la base de datos), SCARAA toma este valor y consulta a la base de datos, si este número existe en algún usuario actualiza su estado que se encontraba de “pendiente”, por “aceptado” (Ver Figura 41). De esta manera queda confirmada su inscripción.

Figura 41: Estado de la inscripción en la base de datos

6.2.2 Alternativa 2º

En el caso de no realizar la inscripción a través la app móvil el usuario deberá acercarse al espacio físico donde se desarrollará el curso y colocar su DNI en la pantalla HMI. (Ver Figura 42)

Figura 42: Registro por pantalla HMI

Figura 43: Prototipo SCARAA con las dos alternativas

Ver en el ***Anexo C (página 99)*** la secuencia del armado del dispositivo.

Base de datos en la nube

El puerto Ethernet está en constante comunicación con la base de datos alojada en la nube. SCARAA se encarga de obtener los datos por medio de variables y almacenarlos en la nube. Para ello usa la conexión a internet y realiza el registro.

La información recolectada debe ser procesada, convirtiéndolos en unidades que el usuario pueda interpretar, con la ayuda del puerto Ethernet, se registra dicha información en la base de datos. Todo lo mencionado anteriormente se realiza en tiempo real, es decir, al obtener los datos o cada vez que cambie uno de ellos se realiza el registro.

6.3 Manual de usuario de la aplicación móvil "SCARAA"

Esta guía proporciona los detalles y requerimientos para el uso correcto de la aplicación móvil, esto con la finalidad de brindar al usuario una herramienta que asegure el uso correcto de la aplicación.

Figura 44: Logo de la app SCARAA "Sistema De Control De Acceso Y Registro Automático De Asistencia"

- **REQUERIMIENTOS**

  Los requerimientos mínimos para que la aplicación SCARAA funcione correctamente, son los siguientes:

  1) Sistema operativo Android (4.1 o mayor).

  2) Procesador de 400 MHz.

3) Memoria RAM de 256 Mb.

4) Chip de NFC

5) Conectividad (3Gy/oWIFI).

- **INSTALACION**

Una vez que la aplicación se ha instalado correctamente es preciso ubicar el ícono ejecutable de la aplicación, para asegurarse que se encuentra instalada correctamente (como se muestra en la Figura 45). Para hacer uso de la aplicación SCARAA, sólo es necesario tocar el icono de la aplicación para abrirla y empezar a trabajar con ella.

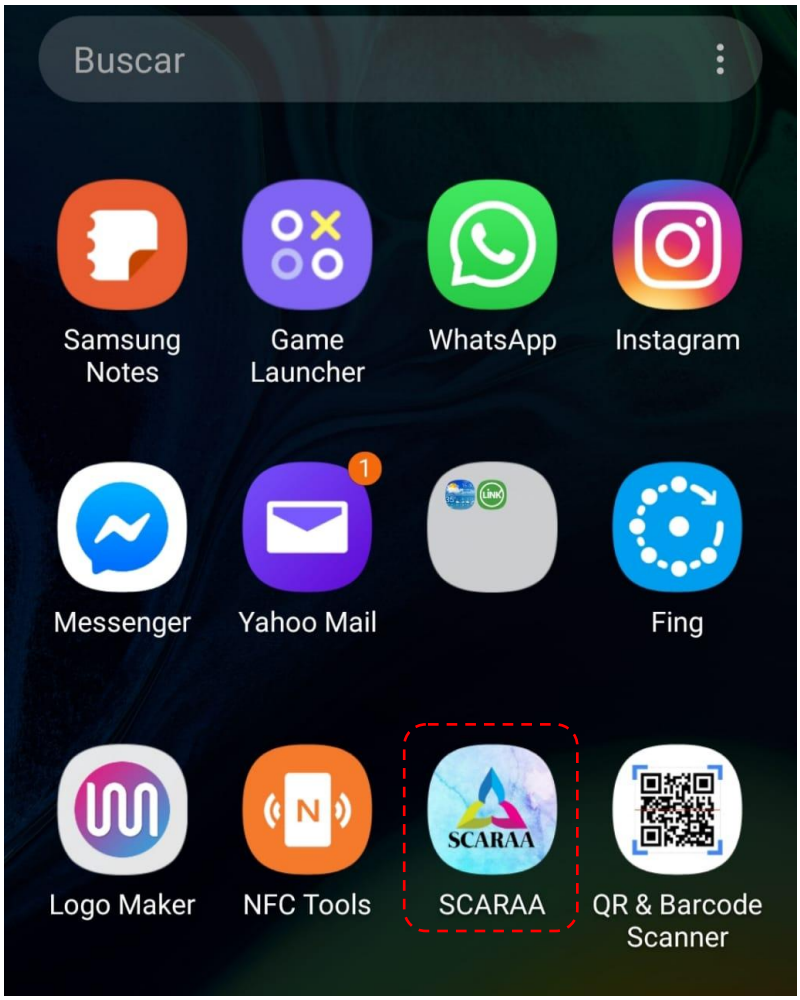


Figura 45: Pantalla en el teléfono que muestra el ícono de la aplicación SCARAA

- **CREAR UNA NUEVO USUARIO**

Para crear un nuevo usuario en SCARAA nos dirigimos al icono “REGISTRAR” de la pantalla principal de la app.

A continuación rellenamos un simple formulario que nos aparece, solicitando los siguientes datos:

DNI: Número Nacional de Identidad.

Contraseña: Introducimos una contraseña mayor a 4 caracteres y menor a 10, para poder acceder a la aplicación.

Nombre: Se trata del nombre de usuario con el que accederemos posteriormente a esta aplicación.

UID: Numero único de identificación del tag (llavero, tarjeta o teléfono con versión de Androide 9.0), usado por el usuario para registrar su asistencia en el curso.

Para terminar con el registro pulsamos el botón “CREAR CUENTA”.

- **HERRAMIENTA PARA NFC**

NFC Tools es una aplicación para leer, escribir o programar tareas sobre etiquetas NFC y otros chips RFID compatibles, se encuentra disponible para su descarga desde Play Store (Ver figura 46)

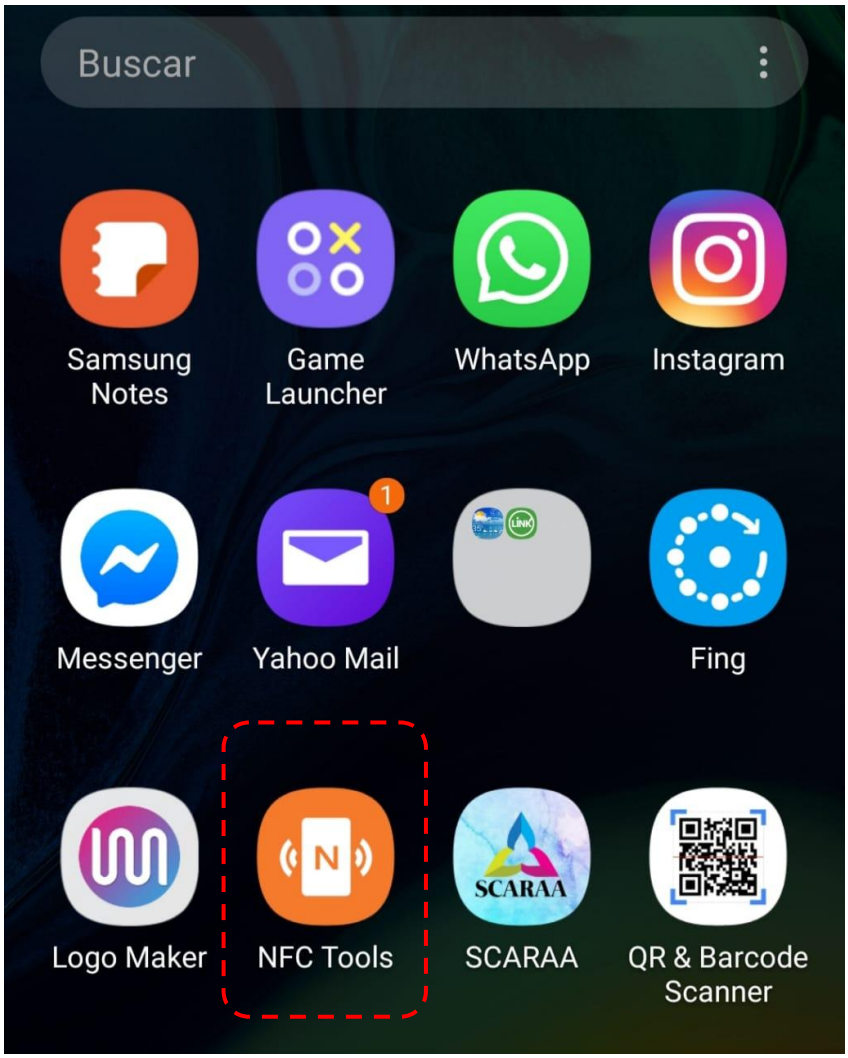


Figura 46: Logo de la app NFC Tools

Con tan sólo acercar el dispositivo a un chip NFC éste pueda leer los datos que contiene o ejecutar acciones. (Ver figura 47).

Figura 47: Interfaz de la app para acercar un dispositivo

La información que se necesita se muestra en la figura 51, como el UID del dispositivo.

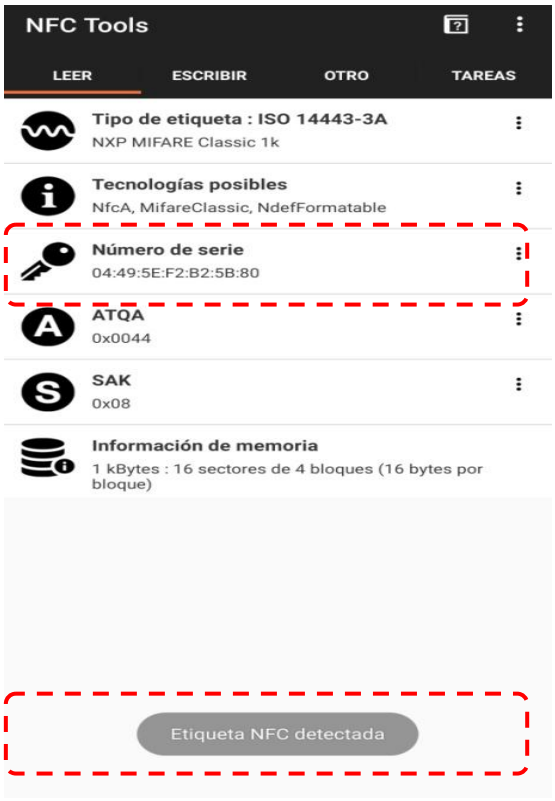


Figura 48: Interfaz de la app denotando la información de la etiqueta NFC

- **LISTADO DE CURSOS**

Para poder acceder al listado de los cursos disponibles cargados en la app, deberá ingresar el DNI, la contraseña y presionar el botón de “LOGIN”. (Ver figura 28).

En esta pantalla, se mostraran un listado de los cursos disponibles, con sus respectivos detalles. (Ver figura 49).

Figura 49: Listado de cursos

Al hacer click en el curso que desea realizar la inscripción, se podrá visualizar 3 estados posibles:

- Inscripto correcta
- Ya estas inscripto en este curso
- No posee cupos disponibles

CAPITULO VII

7 TRABAJOS FUTUROS

Durante el desarrollo de esta tesis surgieron ideas para mejorar el sistema SCARAA, a continuación se presentan algunos trabajos futuros que pueden desarrollarse como resultado de esta investigación o que, por exceder el alcance de esta tesis, no han podido ser tratados con la suficiente profundidad.

En este proyecto se puede continuar con reportes, para imprimir en PDF, pero dado que se tenían muchos requerimientos se decidió dejarlo de lado ya que se estaba tratando con muchas nuevas tecnologías las cuales podían aumentar el riesgo de fracaso del proyecto.

Dado a que los usuarios podrían entregar su Smartphone, tarjeta o tag a otro para que realice su registro por él, se podría implementar la función de sacar una fotografía al momento de realizar su marcaje la cual verifique que fue este quien lo realizo.

CAPITULO VIII

8 CONCLUSIONES

Según el objetivo principal: *Desarrollar un sistema de registro para optimizar el control de asistencia automatizado en el auditorio de la FTyCA, mediante el uso de sistemas embebidos y la tecnológica NFC*. Se obtuvo satisfactoriamente un sistema de Control de Acceso y Registro Automático de Asistencia, gracias al uso de internet de las cosas, sistemas embebidos, aplicaciones móviles, NFC, y la pantalla HMI.

El resultado más notorio que se observó al aplicar las tecnologías utilizadas durante el desarrollo del Trabajo Final, fue el uso de la tecnología NFC con los teléfonos móviles mediante una interfaz de comunicación simple, permitiendo que ambos sistemas, con características totalmente distintas, se comuniquen entre sí con el fin de dar solución a la problemática planteada.

También, cabe destacar las posibilidades que brinda la plataforma de hardware Intel Galileo. Facilitando el desarrollo del prototipo gracias a su gran conectividad, potencia de proceso, el uso de un Entorno de Desarrollo (SDK) muy sencillo. El desarrollo de este Trabajo Final no solo permitió cumplir con el objetivo principal; sino que también permitió ampliar y profundizar los conocimientos en el manejo de métodos, prácticas, tecnologías y herramientas relacionadas con el desarrollo de Servicios Web RESTfull y Sistemas Embebidos.

El proyecto desarrollado muestra el trabajo realizado utilizando un sin número de conceptos informáticos y electrónicos, logrando interrelacionar cada uno de ellos para cumplir con el objetivo. Se realizó una descripción detallada de los sistemas de control de acceso por tags con NFC (tarjetas, llaveros y Smartphone) en la actualidad, los beneficios y ventajas que se obtienen con su implementación. Además, se recopiló información sobre diferentes conceptos necesarios para el desarrollo de este proyecto, como los microcontroladores y software de programación. Se utilizó el protocolo de comunicación NFC, como protocolo ideal para los sistemas de control de acceso, gracias a sus ventajas y beneficios.

La aplicación móvil permitió dicho control interactuando con los sistemas embebidos como enlace con el sistema a controlar el registro y la disponibilidad del cupo de los cursos. Y se logró comprobar que los teléfonos con la versión de Androide 9.0 en adelante no cambian su UID, permitiendo usar dicho teléfono como una tarjeta inteligente.

La curva de aprendizaje fue difícil, y más aún la aplicación de estos conocimientos para el desarrollo del proyecto, se debió realizar una investigación exhaustiva de las

funcionalidades de estas tecnologías para evitar errores o problemas que podrían haber sido ocasionados dado el poco manejo de información respecto a los ya mencionados.

El sistema ha logrado cumplir con los objetivos establecidos al inicio del proyecto, a pesar de las dificultades. Se espera que el proyecto, actualmente un prototipo, pueda ser integrado en un futuro a la FTyCA mejorando este proceso y mitigando los problemas descritos anteriormente.

REFERENCIAS

[1] Wolf W. Computers as Components Principles of Embedded Computing System Design. 2nd ed. Burlington: Elsevier; 2008.

[2] Laboratorio de Sistemas Embebidos Facultad de Ingeniera Universidad de Buenos Aires.

[3] Tesis Mario Navarro “Integración de Sistemas Embebidos y Web Services para la Automatización de la Carga de Signos Vitales en el SIGeSa”, Julio 2.017.

[4] BARR, Michael y MASSA, Anthony. Programming Embedded Systems with C and GNU Development Tools.

[5] Santiago Vásquez, Diseño E Implementación De Un Sistema Electrónico Para El Registro De Acceso Y Envío De Información Mediante Tecnología Nfc Al Personal Administrativo Y De Soporte Técnico De La Empresa Wisp Airmaxtelecom Soluciones Tecnológicas S.A., Ecuador, 2016.

[6] Wilber Iñiguez L.- Jorge Padilla C. “*Near Field Communication*-Teoría y Aplicaciones”, Cuenca Ecuador, 2014

[7] Página oficial de NFC: https://nfc-forum.org

[8] Revista digital - https://es.popularhowto.com/nfc-data-exchange-format - 2018

[9] Eugenia Bahit , “Scrum y Extreme Programming - para programadores”, Buenos Aires, Argentina, 2012.

[10] Página de ingeniería en software- http://ingenieriadesoftware.mex.tl/52753_xp---extreme-programing.html

[11] Manual del chip NFC – “PN532 NFC RFID Module User Guide”

[12] Manoel Carlos Ramon, “Intel Galileo and Intel Galileo Gen 2,” en Intel Galileo and Intel Galileo Gen 2: Api Features and arduino projects for linux programmers.

[13] Tutoriales arduino – Luis Llamas- Leer, grabar, o emular tags NFC con Arduino y PN532 - 20 abril, 2018

[14] Página oficial de Nextion - https://nextion.itead.cc/

[15] tesis-Análisis, diseño e implementación de un sistema de control de asistencia de personal para la Unidad Desarrollo Tecnológico- G U S T A V O A L B E R T O I N O S T R O Z A U R I – Concepción abril 2016.

[16] Asociación de cursos online - https://www.programoergosum.com/cursos-online/arduino/253-curso-de-iniciacion-a-arduino/software-arduino-ide

BIBLIOGRAFÍA

- PRESSMAN, Roger. *Software Engineering. A Practitioner's Approach*, McGraw-Hill, 2001 (tr. Española de Rafael Ojeda Martín, Isabel Morales Jareño, Virgilio Yagüe Madrid, 2002).
- GALLINA, Sergio Hilario; ARANDA, Marcos Darío; FERRARO, Matías Leandro. *Sistema de tiempo real*. Catamarca, Argentina, 2015.
- SABINO, Carlos. *Como hacer una tesis*. Caracas, Panapo, 1994.
- SCARANO, Eduardo R. *Manual de Redacción de Escritos de Investigación.* Buenos Aires, Macchi, 2004.
- GALLINA, Sergio Hilario, *Circuitos Digitales: Un Recorrido desde la compuerta al VHDL.* Catamarca, Argentina, 2009.
- Apuntes del curso: Android. Facultad de Tecnología y Ciencias Aplicadas. Universidad Nacional de Catamarca, 2013.
- Apuntes de Cátedra. Metodología de la Investigación Científica. Universidad Nacional de Catamarca, 2012.
- Murphy Mark L. (2008), The Busy Coder's Guide to Android Development. [en lìnea]. Estados Unidos: CommonsWare. Disponible en: http://commonsware.com/Android/Android.2_1.sampler.pdf [2011, 26 de Octubre]
- Página oficial de Arduino - https://www.arduino.cc/en/main/software
- Página oficial de Nextion - https://nextion.itead.cc/resources/download/nextion-editor/

Índice de Figuras

Índice De Tablas

ANEXOS

1 ANEXO A

- CODIGO Módulo pn532 NFC para enviarle los UID analizados a la placa Intel Galileo.

```
#include <Ethernet.h>
#include <Wire.h>
#include <Adafruit_PN532.h>
IPAddress server(192, 168, 1, 10);
byte mac[] = {0x98, 0x4F, 0xEE, 0x01, 0x0E, 0xE6 };
char user[] = "u192354523_viba";
char password[] = "    ";
EthernetClient client;
#define PN532_IRQ   (2)
#define PN532_RESET (3)
Adafruit_PN532 nfc(PN532_IRQ, PN532_RESET);
int valor;
char* numero;
char nombre;
void setup(void) {
 Serial.begin(115200);
 nfc.begin();
 uint32_t versiondata = nfc.getFirmwareVersion();
 if (! versiondata) {
  Serial.print("PN53x no encontrado");
  while (1);
 }
 Serial.print("Found chip PN5");
 Serial.println((versiondata >> 24) & 0xFF, HEX);
 Serial.print("Firmware ver. ");
 Serial.print((versiondata >> 16) & 0xFF, DEC);
 Serial.print('.');
 Serial.println((versiondata >> 8) & 0xFF, DEC);
 nfc.setPassiveActivationRetries(0xFF);
 nfc.SAMConfig();
```

```
 Serial.println("Esperando tarjeta ISO14443A");
 if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  for (;;)
   ;}
 delay(1000);
 Serial.println("connecting...");
 if (client.connect(server, 80)) {
  Serial.println("connected");
 }
 else {
  // kf no conseguiste una conexión con el servidor:
  Serial.println("connection failed");
 }
}
void printArray(byte *buffer, byte bufferSize) {
 Serial.print("UID EN HEXA: ");
 for (byte i = 0; i < bufferSize; i++) {
  Serial.print(buffer[i] < 0x10 ? " 0" : " ");
  Serial.print(buffer[i], HEX);
 }
 client.println();
 Serial.print("UID EN DEC: ");
 for (byte i = 0; i < bufferSize; i++) {
  Serial.print(buffer[i] < 0x10 ? " 0" : " ");
  Serial.print(buffer[i], DEC);
 }
 Serial.println();
}
void array_to_string(byte array[], unsigned int len, char buffer[])
{
 for (unsigned int i = 0; i < len; i++)
 {
  byte nib1 = (array[i] >> 4) & 0x0F;
  byte nib2 = (array[i] >> 0) & 0x0F;
  buffer[i * 2 + 0] = nib1  < 0xA ? '0' + nib1  : 'A' + nib1  - 0xA;
  buffer[i * 2 + 1] = nib2  < 0xA ? '0' + nib2  : 'A' + nib2  - 0xA;
```

```
 }
 buffer[len * 2] = '\0';
}
void loop()
{
 if (client.available()) {
  char c = client.read();
  Serial.print(c);
 }
 if (!client.connected()) {
  Serial.println();
  Serial.println("disconnecting.");
  client.stop();
  for (;;)
   ;
 }
 boolean success;
 uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 };
 uint8_t uidLength;
 success = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A, &uid[0], &uidLength);
 if (success) {
  Serial.println("Tarjeta encontrada");
  Serial.print("UID Longitud: ");
  Serial.print(uidLength, DEC);
  Serial.println(" bytes");
  printArray(uid, uidLength);
  char serie[32] = "";
  array_to_string(uid, 4, serie);
  client.print("GET/auditorio/ws_v1/nfc.php?serie=");
  client.print(serie);
  client.println(" HTTP/1.0");
  client.println();
  client.println();
  Serial.println("Nro tarjeta");
  Serial.println(serie);
  Serial.println("");
  Serial.println("ARDUINO: HTTP message sent");
```

```
  delay(1000);
  if (client.available())
  {
   Serial.println("ARDUINO: HTTP message received");
   Serial.println("ARDUINO: printing received headers and script response...\n");
   while (client.available())
   {
    char c = client.read();
    Serial.print(c);
   }}
  else {
   Serial.println("ARDUINO: no response received / no response received in time");
  } }
 else {
  Serial.println("Tarjeta no encontrada"); }}
```

2 ANEXO B

- CODIGO de la pantalla Nextion HMI

```
int valor = 0;
int dato = 0;
int i = 2;
int j = 2;
int indice = 0;
byte dato2;
int Cadena[9];
byte txdato = 0xFF;
int total = 0;
int dni [4];
String DNI = "";
void setup() {
  Serial1.begin(9600);
  Serial.begin(9600);
  //pinMode (13,OUTPUT);  // Pin 13
}
void loop() {
  if (Serial1.available() > 0) {
    indice = 0;
    while (Serial1.available() > 0) {
      DNI=Serial1.readString();
     Serial.print("DNI= ");
     Serial.println(DNI);

    }

  }

  delay (500);
  delay(500);
```

```
  Serial1.write(txdato);
  Serial1.write(txdato);
  Serial1.write(txdato);

}
```

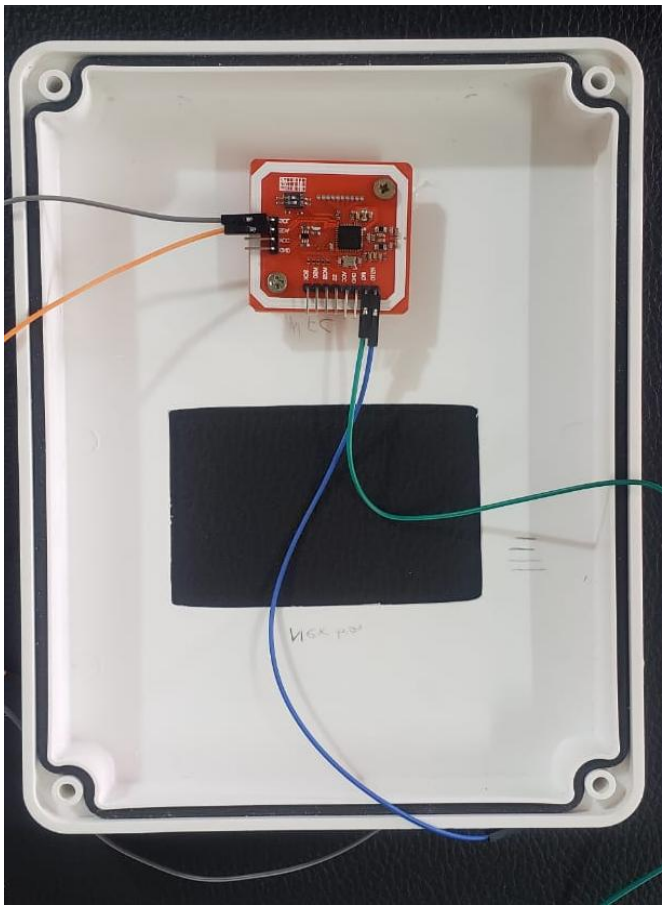
3 ANEXO C

- Armado del dispositivo

- ✓ Medición de la caja de plástico para los componentes

✓ Ubicación de la pantalla HMI en la tapa de la caja

✓ Ajustes de la placa Intel® galileo Gen 1 y la Placa PCB



✓ Control del Chip PN532 NFC/RFID módulo versión 3

✓ **Etapa Final - Prototipo de SCARAA**