

**APLICACIÓN PARA DISPOSITIVOS MÓVILES CON TECNOLOGÍA
MICROSOFT EN WINDOWS PHONE QUE PERMITA RECIBIR INFORMACIÓN
DE LOS CONTENIDOS DE UN MUSEO A PARTIR DE CÓDIGO E IMÁGENES
QR**

JOSÉ DAVID GÓMEZ CRUZ

**UNIVERSIDAD MANUELA BELTRÁN
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2017**

**APLICACIÓN PARA DISPOSITIVOS MÓVILES CON TECNOLOGÍA
MICROSOFT EN WINDOWS PHONE QUE PERMITA RECIBIR INFORMACIÓN
DE LOS CONTENIDOS DE UN MUSEO A PARTIR DE CÓDIGO E IMÁGENES
QR**

**ELABORADO POR
JOSÉ DAVID GÓMEZ CRUZ**

**ASESOR:
JAIRO ALEJANDRO BUITRAGO ROMERO**

**UNIVERSIDAD MANUELA BELTRÁN
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2017**

AUTORIDADES ACADÉMICAS

Dra. ALEJANDRA ACOSTA HENRÍQUEZ
Rectora

Dr. CARLOS ANDRÉS COLLAZOS MORALES
Vicerrector de Investigaciones (E)

Dra. ROCÍO BERNAL GARAY
Vicerrectora de Calidad

Ing. HUGO MALAVER GUZMÁN
Decano Facultad de Ingenierías

Ing. HUGO MALAVER GUZMÁN
Director del Programa de Ingeniería de Software

DEDICATORIA

A Dios por permitirme llegar hasta este punto y darme mucha salud y fortaleza para lograr mis objetivos, además de su infinita bondad, misericordia y amor.

A mi madre Elisa Cruz por apoyarme en todo momento, darme sus excelentes consejos, sus valores, su comprensión, su apoyo, por la motivación constante que me ha llevado a ser una persona de bien, pero más que nada, por su amor.

A mi familia por ayudarme a crecer profesional y laboralmente, porque aprendiendo día a día junto a ellos logre entender el sentido de las cosas que me sirven para ser una mejor persona.

A mis tutores Jairo Alejandro Buitrago, Jimmy Andrey Sánchez y a la universidad por guiarme en el conocimiento dentro del contexto del desarrollo del proyecto y de la aplicación, ejecutando los procesos de la mejor manera.

AGRADECIMIENTOS

Doy infinitas gracias a dios por darme la gran oportunidad de estudiar y por permitirme tener los medios suficientes para terminar esta carrera y no fallar en el intento, así como me dio salud, un buen trabajo y unos buenos amigos.

Le agradezco a mi familia por darme tan maravillosos consejos que me permitieron seguir adelante en mi carrera y culminar con éxito este proceso; sin ellos no hubiera logrado terminar este difícil camino que tengo en el ámbito profesional.

RESUMEN

El presente trabajo investigativo se presenta con el fin de dar solución a un problema, el cual surge desde la ausencia de innovación por parte de los museos, para que los usuarios puedan tener cada día más usuarios o visitantes. Por dicha razón se planteó llenar ese vacío de tecnología dentro del museo con códigos QR para cada elemento o exposición que se encontrara dentro del lugar.

Dicha esta falencia, se propuso la creación de un lector de códigos QR en tecnología Microsoft Windows Phone, el cual permita escanear y poder ser más interactivo de forma automática con los elementos expuestos en cada vitrina.

Esta investigación, se centra en el proceso de la creación de dicho software que permita llegar a un resultado eficiente, ayudan al museo a entrar en actividad con las tecnologías hoy utilizadas a nivel mundial, como lo son los dispositivos móviles, que abarcan el mayor porcentaje de usuarios en interacciones de aplicaciones.

ÍNDICE DE CONTENIDO

Introducción.....	1
1. Antecedentes	2
1.1. Planteamiento del Problema	2
1.2. Justificación.....	2
1.3. Delimitación del Problema.....	2
1.4. Pregunta de Investigacion	3
1.5. Objetivos	3
1.5.1. Objetivo general	3
1.5.2. Objetivos Específicos	3
2. Marco Teórico	4
2.1. Antecedentes de la Investigación.....	4
2.2. Bases Teóricas y Fundamentos Teóricos	5
2.2.1. Fundamentos del Problema de Investigación.....	5
2.2.2. Fundamentos de Ingeniería de Software.....	7
2.3. Antecedentes de la Empresa y Contexto	42
2.3.1. Códigos QR.....	43
2.3.2. Librería para Lector de Códigos QR.....	46
2.3.3. Contexto de la Aplicación	47
2.4. Marco Legal.....	47
3. Metodología.....	49
3.1. Tipo de Investigación.....	49
3.2. Diseño de Investigación	49
3.2.1. Proceso de Investigación	49
3.2.2. Áreas, Paradigmas, Técnicas y Herramientas	50
3.2.3. Cronograma	55
3.2.4. Instrumentos de Investigación.....	57
3.2.5. Presupuesto	59
3.2.6. Aplicación de la Metodología de Investigación.....	59
4. Resultados de la Investigación.....	75
4.1. Analisis de Resultados	75
5. Conclusiones.....	79
Referencias Bibliográficas.....	80

ÍNDICE DE TABLAS

Tabla 1. Tabla Comparativa de Windows Phone Versión 7.0.	41
Tabla 2. Comporacion Windows Phone Version 8.0.	42
Tabla 3. Comparacion Multiples Lectores QR.	45
Tabla 4. Encuesta.	58
Tabla 5. Presupuesto.	59
Tabla 6. Plan de Historias de Usuario.	61
Tabla 7. Iteración 1.....	61
Tabla 8. Documentación Escanear	63
Tabla 9. Documentación Obtener Información.	63
Tabla 10. Iteración 2.....	67

ÍNDICE DE FIGURAS

Figura 1. Proceso RUP.	27
Figura 2. Dispositivo Windows Phone.	36
Figura 3. Plataforma Windows Phone.	37
Figura 4. Interfaz Windows Phone.	38
Figura 5. Tienda de Windows Phone.	38
Figura 6. Capas de Aplicación S.O. Windows Phone.....	40
Figura 8. Proceso de Investigación: Fases.	50
Figura 9. Arquitectura de Software.....	53
Figura 10. Cronograma.	55
Figura 11. Diagrama de Gantt.	56
Figura 12. Diagrama UML.	62
Figura 13. Diagrama Actividad por Carril.	64
Figura 14. Diagrama de Estado.....	65
Figura 15. Diagrama de Secuencia.	66
Figura 16. Código QR.	75
Figura 17. Lector QR.....	76
Figura 18. Camara Lector QR.	76
Figura 19. Código QR Atrapado.....	77
Figura 20. Generación Link desde QR.....	77
Figura 21. Información Final Código QR.	78

INTRODUCCIÓN

En el avance que ha tenido Windows al lanzar un sistema operativo para dispositivos móviles, encontramos diversas versiones en las cuales se han encontrado grandes beneficios para los usuarios que hacen uso de estos productos que proveen un innumerable mundo de soluciones por medio de App's dirigidos a la cotidianidad de cada consumidor.

El desarrollo de nuevas aplicaciones para este tipo de dispositivos se ha convertido en un reto más exigente en cuanto a cubrir las necesidades de cada usuario, debido a que el avance que se genera día a día en la tecnología, impulsa hacia la innovación de todo tipo de aspectos que a su vez, de una forma u otra, generan problemas, como por ejemplo recuperar y procesar información que ha sido obtenida desde el dispositivo móvil.

Para resolver el procesamiento de información se sabe que primero debe ser transportada a servidores remotos que, estableciendo ciertos algoritmos para esta tarea, recupera dicha fuente y la lanza de manera veraz para que el usuario pueda observar el contenido de lo procesado; En esta acción se puede demostrar que el sistema operativo de Windows Phone es capaz de controlar los modelos en los que está construido una aplicación, gestionando en un orden específico las interacciones de usuario, que permiten relacionarse entre las interfaces de una aplicación y las secuencias correspondientes de su posterior procesamiento.

Por ende, es importante entender todo lo relacionado de la plataforma de desarrollo para Windows Phone y sus respectivas funcionalidades, pero más a fondo el procesamiento de información por códigos e imágenes QR que permitan al usuario interactuar dentro de un museo.

1. ANTECEDENTES

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad por el avance tecnológico y la globalización que experimenta el mundo, este se ha apropiado de nuevas necesidades respecto a la adquisición de información y es importante llevar este avance de información al ámbito cultural de una manera simplificada.

Con el auge los teléfonos inteligentes, los cuales permiten tener la información cada vez más rápida y veraz al alcance de la mano en el momento que el usuario lo requiera, se ha querido realizar un proyecto que busca la implementación de la herramienta lectora de códigos QR en los museos, ya que las soluciones informáticas pueden lograr satisfacer la necesidad de información en esta área apenas conocida.

Los QR (quick response), son herramientas tecnológicas que han logrado recopilar, filtrar, segmentar y personalizar información de acuerdo los intereses del usuario permitiendo obtener la información completa que en este caso es una pieza del museo que se quiera conocer más a fondo, además, de brindar un registro necesario para el museo que permite tener mayor organización y control de las exposiciones.

Implementando esta herramienta se desea acercar el arte de un museo a los usuarios que lo visitan, logrando que estos adquieran un conocimiento más profundo de manera rápida y concisa, lo que a su vez incentivara a nuevos usuarios a hacer parte de esta innovadora manera de adquisición de información en el ámbito cultural.

1.2. JUSTIFICACION

Para la realización de esta investigación nos basamos en el desarrollo de una aplicación para dispositivos móviles que permita leer códigos QR, con el objetivo de implementarlo en un museo en donde se realizara el despliegue de dicho aplicativo que mostrara la información completa de las múltiples piezas que se exponen a diario; Adicional a esto, se busca proporcionar una ayuda con respecto al control del inventario que tienen a cargo para las diferentes exposiciones que se presentan.

Demostrando la importancia que tienen estos códigos en diferentes campos de aplicación, se observa que es imprescindible el desarrollarlo desde dispositivos móviles con plataforma de Windows Phone, el cual busca un acercamiento de los diferentes usuarios a la cultura y al arte en general de los distintos museos o patrimonios nacionales.

1.3. DELIMITACIÓN DEL PROBLEMA

En el mundo del desarrollo para aplicaciones móviles, existen infinitos escenarios que permiten la interacción de acciones que virtualizan la vida cotidiana y que ayudan a realizar una serie de acciones cada vez más fácil y simple.

En este orden de ideas, las aplicaciones móviles están para ayudarnos y no para retardar procesos, como por ejemplo transmitir información de manera rápida y concisa. Así mismo existen múltiples plataformas de sistemas operativos móviles que soportan todas aquellas aplicaciones que ya generan un valor agregado a nuestra forma de pensar o realizar tareas.

Esta investigación se basa en el concepto de tecnología móvil, más específicamente en la plataforma de desarrollo de Windows Phone, la cual tiene un gran porcentaje del mercado de usuarios que consumen las aplicaciones a nivel mundial, con el fin de desarrollar un lector de códigos QR que permita interactuar entre la cámara del dispositivo y una imagen en dicho formato (QR), para que así obtenga información de un elemento específico catalogado como histórico dentro del museo.

1.4. PREGUNTA DE INVESTIGACION

¿Cómo la construcción de un lector de códigos QR puede ayudar a un museo dentro sus múltiples elementos allí expuestos?

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

- Desarrollar una aplicación para dispositivo móvil en tecnología “*Windows Phone*” para un museo, que permita obtener búsquedas de información a partir de procesamiento de códigos o imágenes QR obtenidas desde el móvil que ayude a recuperar información acerca de algún hecho histórico, objeto o personaje de influencia humana.

1.5.2. OBJETIVOS ESPECIFICOS

- Explicar los diferentes conocimientos que propone la ingeniería de software en cuanto a metodologías y patrones a seguir para aplicarlos al desarrollo de la aplicación.
- Proponer las principales características de “*Windows Phone*” y su plataforma, para así determinar la arquitectura y componentes básicos para la ejecución de la aplicación a realizar.
- Exponer los códigos QR, sus características, así como las diferentes aplicaciones existentes para lectura de los mismos y emplear los conocimientos adquiridos para tener un fundamento en el desarrollo de una aplicación orientada hacia un museo.
- Desarrollar una aplicación para dispositivos móviles que interprete información gráfica de códigos QR y realice una posterior entrega de información, entorno a un museo.

2. MARCO TEORICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Dentro de la investigación y una posterior aplicación a desarrollar, se ha considerado los antecedentes frente a la historia de los códigos QR y como estos han evolucionado y se han convertido de gran importancia, tanto para la educación como para el comercio; Podemos ver que hoy en día se encuentran por muchos lados los ya tan famosos códigos QR y sus posteriores aplicaciones, como lo es en libros, buses, productos, sitios turísticos, publicidad, e incluso hasta llegar a nuestro punto de interés, los museos.

En el artículo “Aprendizaje Móvil Aplicado en la Educación Usos prácticos ~ QR Code” del autor Bayonet Robles, Luis E, se puede observar que consiste el “QR Code”: “El “QR Code” son códigos de barras de dos dimensiones que puede contener cualquier texto alfanumérico y ofrecen a menudo direcciones URL que llevan al usuario a sitios donde puedan aprender acerca de un objeto o lugar (una práctica conocida como "mobile tagging"), representando más información que un código unidimensional de un similar tamaño.”[1]

Los orígenes se remontan a un código matriz (código de barras de 2 dimensiones), el cual se implementó por una organización japonesa en el año de 1994. Las iniciales "QR" se derivan de la palabra "Quick Response ~ respuesta rápida", donde su creador, intentaba que el contenido del código permitiera ser decodificado a alta velocidad. [1]

El uso de los datos se puede traducir por medio de un código QR donde muchos de los cuales están disponibles en las diferentes tiendas online de modo gratuito. Los usuarios simplemente introducirán los respectivos datos a ser traducidos, y el generador produce un código QR, donde se puede proporcionar de manera electrónica o en formato impreso. [1]

Hoy en día existen diferentes y variados tipos de códigos bidimensionales como por ejemplo Aztec, Code, Maxicode, BIDs, DataMatrix, etc; Se puede decir a partir de esto que los más famosos y usados son los llamados DataMatrix, implantados en el año de 1989, y posteriormente los códigos QR inventados por Toyota en el año de 1994, con el fin de identificar piezas de los coches en proceso de producción, sin embargo su uso es más desarrollado a todo tipo de servicios. [1]

De estos hechos empezaron a crearse los “QR codes” durante la historia, donde existen dos tipos diferentes: el japonés JIS X 0510 instaurado por la JIS y distribuido en enero de 1999 y a su vez la creación del estándar de la ISO, ISO/IEC 18004 de 2000 y revisado en 2006 (ISO/IEC 18004:2006); Desde 1999 la empresa que patentó estos códigos permitió usarlos libremente sin restricción alguna de derechos. El éxito de estos códigos se debe, en gran parte, precisamente a que su estandarización es abierta y su decodificación se puede realizar mediante cualquier dispositivo móvil que tenga incorporada una cámara, sin importar la calidad con que esta toma fotos; Además se caracterizan principalmente por presentar una gran velocidad de respuesta en la transmisión de algún dato en específico guardado dentro de tal código. [1]

Desde el año 2003 en Asia y más influyentemente en Japón, se comenzó a usarse los “QR codes” creando etiquetas en las que las personas podían leer información relacionada con productos, servicios, eventos, sitios de interés en internet; y todo esto solo por medio del dispositivo móvil celular y una simple aplicación de lectura e interpretación de estos códigos. [1]

Actualmente se empieza a ver más desarrollo dentro del ámbito de los códigos QR, donde intentan introducir todo tipo de información para persuadir al usuario a buscar, comprar, obtener servicios o productos; Además también puede ayudar a realizar cosas de manera automática como por ejemplo la conexión a una página web, solo con escanear el código QR respectivo y no digitar la URL, pero resulta lento y exhaustivo. [1]

2.2. BASES TEORICAS Y FUNDAMENTOS TEORICOS

2.2.1. Fundamentos del problema de investigación

La palabra Museo viene de del termino griego “μουσείον” (mouseion), el cual hace referencia a un templo o espacio que intenta explicar las nueve musas o inspiraciones las cuales son Euterpe, Calíope, Polimnia, Clío, Melpómene, Talía, Terpsícore, Erato, y Urania. Entrando en más detalle, se dice que un museo es aquella institución permanente que se encarga de servir a la sociedad de manera abierta al público, donde se adquiere, conserva, estudia y expone un patrimonio tanto material como inmaterial de la humanidad con el fin de aprender [13].

- En la actualidad existen múltiples y diferentes museos con sus respectivas subcategorías, pero entre las más comunes existen:
- Antropológico: Aquí se expone el manifiesto de la diversidad humana incluyendo piezas y contenidos que tratan sobre aquellos aspectos biológicos y sociales.
- Arqueológico: Se dedican a la divulgación de la arqueología y cuya colección procede en su mayor parte de excavaciones.
- De arquitectura: Son museos que contienen estudios dedicados a los procesos constructivos, los creadores y los edificios diseñados.
- De arte contemporánea: En este museo se encuentran obras y contenidos que poseen un orden cronológico que empieza a finales del siglo XIX hasta lo que es la actualidad.
- De artes decorativas: Tiene exposiciones de elementos que producen objetos funcionales y ornamentales como por ejemplo la orfebrería, los bordados, el vidrio la cerámica, etc.
- Ciencias naturales: Se dedican al conocimiento de la diversidad del mundo natural como la flora, la fauna, la geología, etc.
- Etnográfico: su objeto y contenido principal es tratar el folclore así como los usos y costumbres populares de una sociedad específica.
- Histórico: Son aquellos que se dedican a difundir la historia general de una ciudad o territorio específico para que se comprendan sucesos de una manera más fácil.
- Marítimo y naval: Explican o exponen temas relacionados sobre la navegación marítima y todo lo que se refiere a los océanos.

- Militar: museos cuyos objetivos están asociados al ejército o ciertos acontecimientos bélicos.
- Musical: dedican sus contenidos asociados a la música y su historia como lo es los instrumentos, los compositores, la opera o músicas populares. [13]

Hoy en día es muy común pensar en la implementación de tecnología en el ámbito de un museo, sin embargo a veces carece de esta para incentivar o motivar al público a que se apasionen por dichos recintos y sus exposiciones. Entonces, siguiendo la continua evolución en que el mundo se encuentra, se ha incorporado elementos tecnológicos para que sea intuitivo con los visitantes que observan las exposiciones encontradas. [14]

El uso de las nuevas tecnologías como por ejemplo multimedia, ayuda a que se invoquen experiencias interactivas y comunicaciones de manera bidireccional. También como factor positivo encontramos páginas web que día a día atraen visitantes a los museos. Por otro lado están todas aquellas ayudas tecnológicas que hacen accesibles la visita de estos lugares a personas con deficiencias auditivas o visuales. Por ende, uniendo las grandes ventajas que trae la tecnología se puede deducir que el museo adquiere una función de adaptación de sus elementos con el contexto social-educativo. [14]

Por supuesto cabe recordar y tener en cuenta, que los museos poseen o se integran en armonía con la tecnología para ayudar la interacción entre los elementos del sitio y no para ser el elemento principal y por el cual los visitantes acuden a observar una exposición. [14]

Para esta investigación, se hace énfasis en lo Códigos QR y como estos benefician a los museos en su cotidianidad de atraer visitantes y crear esa interacción innovadora para cada elemento expuesto. Por ello el implementarlos trae beneficios como despertar aquella curiosidad en la gente, crear educación dentro de los mismos usuarios, tener un bajo costo de la implementación y hacer una interacción diferente con el público de diferentes edades. [15]

Además la implementación de los códigos dentro del museo, puede dar ese pro de la información sobre un artista o una pieza de arte; Puede ser representada de diversas formas como por ejemplo un texto, un video, audio o hasta un juego. Aquella exposición de manera anticuada donde se tenía que investigar de más al salir del museo para entender algunos elementos se quedó en la historia, ya que la implementación de códigos ayudaría a que todo lo que se necesita de más se obtenga de manera instantánea. [15]

Para ver que tan influyente es la tecnología de cualquier tipo implementada en un museo se ve como el museo de Louvre mantiene su estilo clásico para añadir la comodidad tecnológica de Nintendo, donde sustituyo las guías de audio por muchas consolas 3DS donde se permite profundizar mucho más en las obras expuestas. Con esta ayuda el Nintendo permite encontrar todas las obras en alta definición y encuentra información adicional como lo permite el software. [16]

Por otro lado encontramos el museo judío de Berlín, donde se muestra todo lo vivido y sufrido por los judíos en el holocausto nazi, el cual creo una alianza con Acousticguide (pionero de interpretación multimedia) y con Apple permite por medio de su IPod ofrecer una guía programada en ocho lenguajes diferentes para que cada que el usuario se aproxima a una obra se reproduzca de manera automática información adicional, fotos, videos o artículos. [16]

Por último el museo americano de historia natural, ubicado en el corazón de New York es uno de los más grandes y famosos, ya que se compone por 25 edificios que interconectan 26 salas de exposiciones, laboratorios y bibliotecas. Este posee en sus exposiciones Premium una sala para películas Imax o un show planetario Hayden equipados con la última tecnología. [16]

Todo esto nos lleva a la única conclusión de que la tecnología es una gran ayuda tanto para el usuario como para el museo, porque permite un gran nivel de interacción de manera que todos los sentidos del ser humano se pueden llegar a interconectar con una experiencia única e innovadora y más si se refiere al acceso rápido a información que hoy en día vive el mundo.

2.2.2. Fundamentos de ingeniería de software

Dentro de los fundamentos de esta investigación en la ingeniería de software, pueden ser abarcados múltiples temas de conocimiento que de una forma u otra puede servir de gran aporte para la investigación de dicho tema, por lo que se considera lo siguiente:

2.2.2.1. Áreas del conocimiento de ingeniería de software

2.2.2.1.1. Requerimientos del software

Los requisitos del software son aquellos análisis, especificaciones y dichas validaciones de requisitos que un software pueda tener. Se reconoce mundialmente por proyectos de ingeniería, ya que poseen vulnerabilidades cuando las actividades se realizan mal. [37]

Los requisitos del software dan las necesidades en un producto de software que contribuye a la solución de un cierto problema del mundo real. [37]

2.2.2.1.1.1. Fundamentos de los requisitos del software

2.2.2.1.1.1.1. Definición de un requisito del software

Esta es una de las cualidades o característica innatas donde corresponde presentar una solución a un problema expuesto dentro del mundo cotidiano o real. Con respecto al software, hace referencia a todos los problemas de los que carece dicho software, por tales motivos un requisito es aquella característica que se debe exhibir para dar una solución. También se dice que los requerimientos son una mezcla de requisitos de mucha gente entre múltiples niveles dentro de un grupo de trabajo y en el campo donde el software entrara en acción. [37]

2.2.2.1.1.1.2. Producto y requisitos del proceso

Los requisitos de proceso permiten al software definir parámetros dentro del desarrollo para esencialmente constituir un producto. [37]

2.2.2.1.1.1.3. Requisitos funcionales y no funcionales

Se le conocen también como capacidades a aquellos requisitos funcionales que intentan describir las funciones básicas o más importantes que un software va a realizar. [37]

Los requisitos considerados como “no funcionales” son aquellos que, de una forma u otra, obligan la solución y se conocen como premuras o requisitos evaluadores de calidad. A su vez se pueden catalogar como esos requisitos que determinan funcionamiento, que generan el contenido, que ayudan a proporcionar mantenimiento, que dan seguridad o que son de confiabilidad.[37]

2.2.2.1.1.1.4. Características inesperadas

Como su nombre lo dice, las características inesperadas dentro de un desarrollo o un software más exactamente, son aquellos requisitos que no son trabajados bajo un único mecanismo, es decir, su cumplimiento va a obedecer porque los módulos de un software interoperan entre ellos mismos. [37]

2.2.2.1.1.1.5. Requisitos cuantificables

Todo aquel requisito que sea definido debe ser claro y conciso tanto como sea posible, de la misma forma cuantitativamente. Por esa razón es muy importante evitar requisitos imprecisos e inverificables que dependen de la opinión personal. [37]

2.2.2.1.1.1.6. Requisitos del sistema y requisitos del software

Acá se debe considerar que un sistema significa “Una mezcla de elementos para llegar a un objetivo establecido”, donde contienen el hardware, el software, un poco de soporte lógico, información, técnicas, usuarios y todos aquellos elementos que den soporte. [37]

2.2.2.1.1.2. Proceso de los requisitos

2.2.2.1.1.2.1. Modelos de proceso

- Significa aquel paso asociado en el comienzo de un proyecto y el proceso de continuación mejorada sobre un ciclo vital por el que atraviesa un software. [37]
- Iguala todos esos requisitos de un software como elementos configurables y a partir de esto, los manipula usando las mejores prácticas de administración del software en desarrollo. [37]
- Deberá ser acomodado a la estructura y al contexto del proyecto. [37]

2.2.2.1.1.2.2. Agentes de proceso

Tiene como característica fundamental que es multifacético, ya que el experto de requisitos intervendrá la dotación de tecnología lógica ajustada para este caso. Existen muchas personas asociadas además del técnico en los requisitos, donde cada uno de ellos posee un cargo dentro del desarrollo del mismo. [37]

2.2.2.1.1.2.3. Ayuda y gerencia de proceso

Introduce todos esos llamados “recursos de gerencia” de proyectos que se necesiten para ser usados en los procesos de creación de requisitos. Se obtiene el contexto para el punto de inicio y así mismo el alcance de la tecnología de dotación lógica de un proyecto. [37]

2.2.2.1.1.2.4. Calidad y mejora de proceso

Hace referencia a esa calidad de mejora por la que puede pasar un proceso de requisitos, donde su mayor intención se basa en recalcar aquel papel dominante que los requisitos resuelven de manera técnica y en términos de coste y tiempos un producto o software, pasando en todos sus puntos por la satisfacción del cliente. [37]

2.2.2.1.1.3. Captura de los requisitos

La captura de los requisitos consiste en contextualizar y realizar aquellos requisitos que serán de vital importancia para el desarrollo de un software, donde el ingeniero es el encargado en la comprensión del problema que el software

quiere solucionar. Se debe tener en cuenta que la actividad humana es esencial para determinar hasta el más mínimo detalle y además de ello, identifica a los directamente interesados y todas las relaciones que puedan coexistir entre los desarrolladores y el cliente. [37]

2.2.2.1.1.4. Análisis de requisitos

Por medio de este, se hace el proceso de análisis de requisitos con el fin de: [37]

- Descubrir y solucionar los conflictos entre los requisitos
- Detectar todos los límites que un software pueda tener y así mismo, cómo debería obrar equitativamente con su medio de acción.
- Construir esos llamados requisitos de sistema para poder de algún modo sustraer requisitos del software

Cabe destacar que el análisis de los requisitos se limita al modelado conceptual usando diferentes métodos de análisis como por ejemplo los estructurados o las técnicas de diseños. También se debe pensar en la clasificación de requisitos para ayudar a la comunicación de compensaciones y dicho proceso que se presenta de instaurarlas a su vez. [37]

2.2.2.1.1.5. Especificación de requisitos

La especificación de requisitos hace referencia a lo que típicamente se conoce como creación de un documento, ya sea físico o electrónico. Para desarrollos pesados, se realizan 3 tipos de documentos los cuales son definición de sistema, sistema de los requisitos, y requisitos del software. Para sistemas simples, el tercero de estos es el único requerido. [37]

2.2.2.1.1.6. Validación de los requisitos

Todos los documentos mencionados anteriormente, deben estar a nivel con la validación o todos aquellos procedimientos de verificación, donde el ingeniero se centra en entender el contexto de cada uno sin obtener ambigüedad en los mismos para así mismo satisfacer los estándares que permiten que sea comprensible, constante, y finito. [37]

2.2.2.1.2. Diseño de software

El diseño de software mantiene un rol muy importante dentro del desarrollo de aplicaciones porque crea y mantiene modelos que sirven como planos para construcción del mismo software. [37]

Por un lado existe el diseño arquitectónico el cual permite describir la arquitectura y ordenación de alto nivel de un sistema, donde se identifica de modo sistemático componentes e interfaces enlazados entre sí. Por otra parte está el diseño detallado el cual describe de manera autónoma e individual cada componente con el suficiente detalle para que pueda ser construido. [17]

Esta área reúne mucho conocimiento con el fin de pensar a fondo los principios como la abstracción, el acoplamiento, la cohesión, la descomposición y encapsulación para así mismo poseer diseños robustos. Por tal razón es necesario saber solucionar aspectos prácticos como la constancia de datos, sistemas distribuidos, peticiones concurrentes, administración de eventos, recuperación a fallas, etc. [17]

2.2.2.1.2.1. Fundamentos del diseño del software

El diseño también es aquella forma de corregir un problema, donde generalmente se razona un proceso que se compone por dos etapas: El diseño arquitectónico

que describe la forma en que el software se descompone y organiza aquellos componentes y la construcción de un diseño lo más detallado posible, el cual habla de cómo es el comportamiento de cada elemento y además, como produce una salida este proceso es un sistema de modelos y artefactos donde se registran todas esas decisiones fundamentales que se fueron tomando en el camino. [37]

Cada diseño de software domina las consideraciones fundamentales a los múltiples contactos y conceptos del diseño de software como por ejemplo: [37]

- Abstracción
- Acoplador y cohesión
- Descomposición y modularización
- Encapsulación del ocultar de la información
- División de la interfaz y de la puesta en práctica

2.2.2.1.2.2. Cuestiones claves en diseño del software

Para diseñar un software, se debe tener muy claro la siguiente lista de principios fundamentales: [37]

- Concurrencia
- Control y dirección de acontecimientos
- Distribución de componentes
- Manejo de errores y sus excepciones además de planes para la tolerancia de fallos
- Interacción y presentación
- Persistencia de los datos

2.2.2.1.2.3. Estructura y arquitectura del software

La arquitectura de un software da referencia a la descripción de un subsistema y de todos los componentes y relaciones que existen entre sí. Así mismo, la arquitectura propone definir estructuras internas para mayor simplicidad. [37]

2.2.2.1.2.3.1. Estructuras y puntos de vista arquitectónicos

Muy a menudo se suele llamar a estas estructuras como: “Una visión representa un aspecto parcial de una arquitectura del software que manifieste características específicas de un procedimiento de software”. Por ende, todas las visiones que son distintas son de ediciones distintas que se concentran a un diseño de software [37]

- Estructura general
- Sistemas distribuidos
- Sistemas interactivos
- Sistemas adaptables

2.2.2.1.2.3.2. Patrones del diseño

Un patrón es “una solución común a un problema común en un contexto dado”. Por ende los estilos que se consideran como arquitectónicos son patrones que de una forma u otra, describen en qué nivel de software se encuentra una organización (su arquitectura macro); o también para determinar aquellos detalles de un nivel más bajo (su arquitectura micro). [37]

2.2.2.1.2.4. Análisis y evaluación de la calidad del diseño del software

2.2.2.1.2.4.1. Cualidades de los atributos

Diferentes características son lo más importantes para lograr la forma en como el software y su diseño posee una buena calidad, varias características fundamentales como capacidad de mantenimiento, portabilidad, testeo, trazabilidad y así mismo corrección y robustez incluyendo la “aptitud del propósito”. [37]

2.2.2.1.2.4.2. Técnicas de evaluación y calidad del análisis

Existen múltiples técnicas que sirven para asegurar de manera inequívoca la calidad que tiene un diseño de software: [37]

- Estudios en el diseño del software
- Análisis estático
- Simulación y prototipado

2.2.2.1.2.4.3. Medidas

Cuando se determina o estima cuantitativamente aspectos como el tamaño, la estructura, o la calidad de un diseño de software se puede definir como medidas dentro del marco del software. [37]

2.2.2.1.2.5. Notaciones del diseño del software

Las notaciones dentro de un diseño de software hacen referencia a aquellos pasos donde se determinan el comportamiento y la estructuración organizacional de un diseño, llegando a un diseño detallado de lo que es el desarrollo de software. [37]

2.2.2.1.2.5.1. Descripción estructural

A continuación se nombran notaciones, que ayudan a describir las secuencias que componen las estructuras de un diseño de software; aquí se detallan los componentes principales y cómo se interconectan (visión estática): [37]

- Lenguajes descriptivos de la arquitectura
- Diagramas de la clase y objeto
- Diagramas de componente
- Diagramas de despliegue
- Diagramas de la Entidad-relación
- Lenguaje descriptivo de la interfaz
- Diagramas de la estructura de Jackson
- Estructura de cartas

2.2.2.1.2.5.2. Descripciones del comportamiento

Dentro de las descripciones del comportamiento de forma dinámica de un software se visualizan componentes como: [37]

- Diagramas de actividad
- Diagramas de colaboración
- Organigramas de datos
- Tablas y diagramas de decisión
- Diagramas de secuencia
- Transición de estado y diagramas de carta de estado
- Lenguajes formales de especificación
- Lenguajes del diseño de pseudo código del programa

2.2.2.1.2.5.3. Estrategias y métodos del diseño de software

Por medio de las estrategias se puede ordenar un proceso de diseño, donde por medio de métodos se puede converger un sistema de notificaciones y que se puede valorar para una descripción de procesos donde normalmente se utilizan pautas o normas a seguir. Los métodos acá usados, son de gran ayuda para transportar todo ese conocimiento y hacer base en este para que los ingenieros tengan mayor área de acción. [37]

- Estrategias generales
- Diseño (estructurado) orientado a función
- Diseño orientado a objeto
- Diseño Dato-Estructura-Centrado
- Diseño basado en componente

2.2.2.1.3. Construcción de software

Dentro de esta área se concentra el conocimiento apoyado en la creación de software útil a través de la programación de cualquier tipo, depuración de dicho código, pruebas de software y la integración de componentes finales. Los retos acá se encuentran en mermar la complejidad y cumplir criterios o estándares que ayudan a concebir el código generado, para que sea entendible y compatible por otros colaboradores, sin mencionar la optimización necesaria de cada proceso. [18]

2.2.2.1.3.1. Fundamentos de la Construcción del Software

Las características esenciales de la construcción del software se nombran en:

- Minimizar la complejidad. [37]
- Anticiparse a los cambios. [37]
- Construir para verificar. [37]
- Estándares en la construcción: Aquellos estándares que proporcionan ayuda aquellas características de la construcción de un software son: [37]
 - Métodos de comunicación
 - Programación de lenguajes
 - Plataformas
 - Herramientas

2.2.2.1.3.2. Gestión de la Construcción

2.2.2.1.3.2.1. Modelos de Construcción

Existen diversas formas para el desarrollo de software, donde se encuentran sus múltiples versiones creadas, ayudan a construir de manera más exacta que otros, sin embargo se debe definir que estos son un modelo para la construcción como actividad que se ejecuta después de cumplido un trabajo con prerequisites dando a su vez un trabajo sobre el diseño y la planificación previamente detallada. [37]

2.2.2.1.3.2.2. Planificación de la Construcción

Se debe elegir un método claro y preciso para la construcción de un software ya que puede afectar factores como prerequisites de la construcción, el orden de realización, y hasta el valor final de los objetivos planteados inicialmente. A esto se debe sumar que dependiendo el método que se elija, afectara la experiencia del proyecto para reducir la complejidad y anticipar cambios. Cada uno de estos

objetivos puede también afrontarse en los niveles de proceso, requisitos y diseño pero también estarán influenciados por la elección de un método de construcción. [37]

2.2.2.1.3.2.3. Medición de la Construcción

Se pueden calcular numerosas actividades de construcción y artefactos, incluidos el código desarrollado, el código modificado, el código reutilizado, el código destruido, la complejidad del código, las estadísticas de la inspección del código, las tasas de rectificación de errores y de identificación de errores, y los horarios. Estas medidas pueden ser útiles para intenciones de gestión de la construcción, cerciorándose de la calidad durante la construcción, optimizando los procesos de construcción. [37]

2.2.2.1.3.2.4. Diseño de la Construcción

El diseño de la construcción de un software, también se preocupa por el diseño de manera detallada y el trabajo donde extiende restricciones que pueden llegar a ser inamovibles impuesta por un problema del mundo real las cuales pueden ser atacadas por un software, solucionando una parte o todo el inconveniente. [37]

2.2.2.1.3.2.5. Lenguajes de Construcción

Los lenguajes de construcción incluyen todos los tipos de comunicación mediante los cuales un humano puede especificar una solución ejecutable para un problema de un ordenador. [37]

- lenguaje de configuración
- lenguajes de herramientas
- lenguajes de programación

2.2.2.1.3.2.6. Codificación

En la codificación se puede ver como se aplica de manera ordenada las consideraciones que pueden aplicar dentro de una verdadera actividad de construcción de código: [37]

- Técnicas para crear código fuente comprensible.
- Utilización de clases, tipos enumerados, variables, constantes predefinidas, y otras entidades similares.
- Utilización de estructuras de control.
- Tratamiento de las condiciones de error.
- Prevención de brechas en la seguridad a nivel de código (el búfer o el índice de la matriz se desborda, por ejemplo)
- Utilización de recursos por medio del uso de mecanismos de exclusión
- Organización del código fuente (en declaraciones, rutinas, clases, paquetes u otras estructuras)
- Documentación del código
- Puesta a punto del código

2.2.2.1.3.2.7. Pruebas de Construcción

Para las pruebas de construcción se debe tener en cuenta que en el momento de construir se implican dos tipos de pruebas, las cuales son desarrolladas bajo

ciertos términos y criterios el mismo ingeniero del software que implemento una solución por medio de código: [37]

- Pruebas unitarias
- Pruebas de integración

La idea en las pruebas es intentar resolver o reducir esa brecha que existe entre el tiempo en el que se generan fallos en el código y la detección de los mismos. En la mayoría de casos las pruebas de construcción se pueden realizar antes de que se escriba el código para así mismo no generar retrasos o inconvenientes en el mismo. [37]

2.2.2.1.3.2.8. Reutilización

La reutilización en la ingeniería de software permitirá, por medio de la combinación de procesos y actividades de reutilización en el ciclo de vida del software, reusar todo ese código que se puede adaptar a varios procesos. Las tareas relacionadas con la reutilización en la construcción del software durante su codificación y pruebas son: [37]

- La selección de unidades, bases de datos, procedimientos de pruebas o datos de pruebas reutilizables.
- La evaluación de la posibilidad de reutilización del código o de las pruebas.
- Comunicar la información sobre reutilización realizada en el código nuevo, los procedimientos de pruebas o los datos de pruebas

2.2.2.1.3.2.9. Calidad de la Construcción

Dependiendo de uqe se quiere realizar, se tomara las medidas necesarias para tomar técnicas, que ayudaran a su paso a conformar un gran grupo de habilidades en los ingenieros de software que producirán la construcción del mismo: [37]

- Las pruebas unitarias y las pruebas de integración
- Pruebas de Construcción
- El desarrollo de primero-haz-pruebas
- El código paso a paso
- Utilización de aserciones
- Depuración
- Revisiones Técnicas
- Análisis estático

2.2.2.1.4. Pruebas de software

Dentro de las pruebas de software hallamos diferentes tipos, como lo es por su alcance (unitario, integral, de sistema) o su objetividad (funcionalidad, confiabilidad, desempeño, regresión, aceptación, etc.). Por tal razón existen múltiples técnicas que ayudan a la simplicidad del mismo proceso, que su objetivo final es calcular la respuesta correcta del software, por lo que se logra usar tablas de decisión, análisis de fronteras, máquinas de estados e inclusive hasta la misma experiencia. [19]

Las pruebas de software ayudan a radicar un comportamiento de un programa por medio de un conjunto definido de casos de prueba, donde se seleccionan de manera apropiada para ejecuciones de manera finita, evaluando el comportamiento esperado. [37]

La definición anterior pertenece a aspectos esenciales en la identificación del “Área de Conocimiento de las Pruebas del Software”., ya que se explica:

- Dinámicamente significa que hacer pruebas siempre supone ejecutar el programa con entrada de datos (valorados [37])
- Finito: Incluso en programas sencillos, teóricamente podría haber tantas pruebas que realizar, que hacer pruebas exhaustivas podría llevar meses o años. [37]
- Seleccionados: La diferencia esencial entre las distintas técnicas de pruebas propuestas se encuentra en cómo se escoge el conjunto de pruebas. Los ingenieros informáticos deben ser conscientes de que criterios de selección distintos pueden producir grados de efectividad muy diferentes. [37]
- Esperado: Debería ser posible, aunque a veces no sea fácil, decidir si el resultado observado de la ejecución de un programa es aceptable o no, porque si no el esfuerzo de realizar las pruebas sería inútil. [37]

2.2.2.1.5. Mantenimiento del software

Se refiere a alteraciones dentro del software que involucran la prevención de fallas (de manera preventiva), así como para corregirlas (correctivo), también para optimizar el desempeño (perfectivo) o para adaptación a cambios (adaptivo). Esta área ha tomado gran fuerza debido a la adaptación de aplicaciones que están presentes en el mundo cotidiano, donde por ejemplo se vuelven puntos clave la reingeniería, el análisis de impacto, las pruebas de regresión, y el conocido outsourcing de mantenimiento. [19]

Durante los años, el desarrollo de software ha prevalecido de manera más alta que el mantenimiento del software en todo el mundo. Esto, poco a poco va cambiando ya que las organizaciones no se centran en exprimir hasta el último centavo de inversión en el software desarrollado, sino que por medio de outsourcing se ayudan para dar mantenimiento al software. [37]

2.2.2.1.5.1. El Proceso de Mantenimiento

En el mantenimiento se suministra por referencias y normas para colocar en funcionamiento los procesos dentro del mismo. La necesidad del proceso de ingeniería de software está bien documentada. Los modelos CMMI (Modelo de capacidad y madurez) se emplean a procesos de mantenimiento de software, y son similares a los procesos de los desarrolladores. [37]

- Procesos de Mantenimiento: Estos métodos ayudan a proporcionar actividades necesarias de entradas/salidas detalladas a aquellas actividades [37]
 - La Puesta en práctica de Proceso
 - El Problema y el Análisis de Modificación
 - La Puesta en práctica de Modificación
 - La Revisión/Aceptación de Mantenimiento
 - La Migración
 - El Retiro de Software
- Actividades de Mantenimiento: muchas actividades de mantenimiento son similares a aquellas de desarrollo de software. Los mantenedores ejecutan el análisis, el diseño, la codificación, pruebas, y la documentación. Ellos

deben indagar exigencias en sus actividades tal cual creadas en el desarrollo, y la documentación de modernización como el cambio de líneas de fondo. [37]

Existen un gran porcentaje de procesos, actividades, y prácticas además de ser únicas, ayudan de manera acelerada al mantenimiento de software, por ejemplo:

- Transición
- La Aceptación/Rechazo de Petición de Modificación
- Modificación y Escritorio de Ayuda de un problema
- El Análisis de Impacto
- El Apoyo de Software
- Los Acuerdos de Nivel de Servicio (SLAs) [37]

2.2.2.1.5.2. Las técnicas para el Mantenimiento

- Comprensión de Programa: para que los Programadores no gasten gran tiempo en el entendimiento de código de programas para implementar cambios, existen por ejemplo los navegadores de código. Que la documentación se encuentre clara y comprensible puede ayudar al entendimiento de dicho programa [37]
- Reingeniería: Trata de realizar de manera exhaustiva y optimizada la variación de un software para intentar reconstituirlo en una nueva forma. [37]
- Ingeniería de revés: Esta propone que en el proceso del análisis de software poder equilibrar esos componentes con sus respectivas relaciones mutuas y como último fin crear representaciones del software en altos niveles de abstracción. [37]

2.2.2.1.6. Calidad del software

En esta área de conocimiento se involucra procesos de aseguramiento de calidad, verificación, validación, revisión y auditoria, los cuales se consideran de alta categoría para la clasificación de defectos, el control estadístico de calidad, algunos modelos de predicción y análisis de tendencia. [19]

2.2.2.1.6.1. Fundamentos de Calidad de Software

En este punto se define todos esos rasgos de calidad requeridos dentro de un software que intervienen como por ejemplo métodos de medición o los criterios de aprobación para resaltar características como: [37]

- Valor y coste de la calidad [37]
- Modelos y Características de Calidad [37]
- Mejora de Calidad [37]

2.2.2.1.6.2. Procesos de Gestión de Calidad del Software

La calidad dentro del software es un efecto adverso a todas las representaciones de procesos que tengan que ver con software, productos y recursos. Esto además define procesos, requerimientos y medidas para las correspondientes salidas con su respectiva retroalimentación. [37]

Dentro de los procesos que se encuentran para tener una alta calidad del software, se compone de múltiples actividades; Aquí se puede encontrar un gran número de defectos mientras, caso opuesto, se pueden ver las virtudes de un software. Esta planificación puede ayudar a: [37]

- Definición del producto requerido en términos de sus características calidad
- Planificación de los procesos para alcanzar el producto requerido.

2.2.2.1.6.2.1. Aseguramiento de la Calidad del Software (SQA)

Todos los procesos que se presentan acá, ayudan a dar una garantía de efectos sobre el software en construcción que permite mejorar los procesos del ciclo de vida del mismo y que tienen que ser acordes a los requerimientos que se caracterizan por su planificación; Esto expresa un conjunto de actividades que debe generar confianza para la alta calidad dentro del software.

Lo anterior afirma que el problema está claramente entendido y que los requerimientos están plenamente identificados, solucionando un problema real o parte de él. [37]

2.2.2.1.6.2.2. Verificaciones y Validación

La verificación y validación del software se refiere a una aproximación disciplinada a toda la evaluación de productos del software a lo largo de todo el ciclo de vida de dicho desarrollo donde también. [37]

Por consiguiente se trata directamente la calidad de un producto software donde puede usar técnicas de prueba que pueden limitar efectos que pueden ser tratados. [37]

2.2.2.1.6.2.3. Revisiones y Auditorias

- Revisiones de gestión: La finalidad es revisar el progreso, llegando a fases como planes y programas, requerimientos confirmados y su sistema de localización; también se puede valorar la efectividad de los enfoques de gestión empleados para obtener el objetivo final. [37]
- Revisiones técnicas: El fin de una investigación técnica es valorar el producto software para determinar si es apto para su efectivo uso. [37]
- Inspecciones: El propósito de una inspección es detectar e identificar anomalías en los productos software
- Walk-throughs: El objetivo de un Walk-throughs es evaluar un producto de software. Un Walk-throughs puede ser conducido para formar una audiencia en cuanto a un producto de software.
- Auditorias: La auditoría de software intenta suministrar una verificación independiente de la aprobación de productos software y procesos a regulaciones adaptables, estándares, directrices, planes, y procedimientos.

2.2.2.1.7. Administración de la configuración

Hace énfasis al conjunto de aquellos elementos de hardware y software que constituyen un sistema, convirtiéndose desde hace unos años en una conducta que identifica la configuración en diferentes puntos del tiempo con el propósito de lograr control sobre el mismo, sin desaprovechar la integridad y rastreabilidad durante el ciclo de vida del software. [19]

Por medio de la gestión de la configuración, se aguanta el ciclo de vida del software (IEEE12207.0-96) donde se da prioridad a la gestión de proyectos, las acciones de desarrollo y mantenimiento, las actividades de garantía y a los clientes y usuarios del producto final. [37]

2.2.2.1.7.1. Gestión del proceso de la gestión de la configuración del software

La gestión de la configuración del software inspecciona o ayuda a identificar toda aquella evolución e integridad de un producto identificando elementos, negociando y controlando cambios y comprobando, guardando y originando informes de la información de configuración. Desde la perspectiva del ingeniero de software, la gestión de la configuración del software provee las actividades del desarrollo e implementación de cambios. El éxito de una ejecución de la gestión de la configuración del software demanda una planificación y gestión cuidadosas. [37]

2.2.2.1.7.2. Identificación de la Configuración del Software

Por medio de actividades, se establecen elementos que se han de examinar de manera detallada para posteriormente crear métodos de personalización para versiones y herramientas o técnicas que ayudaran a gestionar dichos elementos intervenidos. Estas actividades suministran la base para las otras actividades de la gestión de la configuración del software. [37]

2.2.2.1.7.3. Control de la Configuración del Software

Para un control optimo en la configuración del software, existe una gestión que se encarga de los cambios durante el ciclo de vida que cumple un software. Se puede decir que se generaliza todos los procesos que establece los cambios a realizar donde la autoridad solicitada puede aprobar los cambios y así mismo el soporte para la implementación. La información derivada de estas actividades es útil para medir el tráfico de cambios y ruptura y aspectos por rehacer. [37]

2.2.2.1.7.4. Registro del Estado de la Configuración del Software

El registro del estado en el software se refiere a aquella actividad que registra y así mismo proporciona información sumamente importante para gestiones efectivas: [37]

- Información del Estado de la Configuración del Software [37]
- Informes del Estado de la Configuración del Software [37]

2.2.2.1.7.5. Auditoría de la Configuración del Software

Si se habla de auditoria, se debe relacionar a una actividad que evalúa la conformidad de los productos y procesos que fueron entregados al cliente del desarrollo del software con las regulaciones necesarias y exigidas (en el caso que aplique por las leyes), así como las guías y los estándares dentro del mismo proceso. A la vez se debe contemplar un gran número de responsabilidades, papeles y personas que realizaran y verificaran que dichas características mencionadas anteriormente se cumplan. [37]

2.2.2.1.7.6. Gestión del Lanzamiento y Distribución del Software

Si se habla de lanzamiento, se debe hacer referencia a la distribución que se realiza normalmente para un dispositivos en la configuración del software, eso si fuera de la actividad de desarrollo. Se debe visualizar que esto incluye gran porcentaje de lanzamientos como sea necesarios a los clientes. Aquí un elemento fundamental para el lanzamiento y distribución es las bibliotecas de software ya que ejecuta tareas de forma automática. [37]

2.2.2.1.8. Administración de la ingeniería de software

Es lo que comúnmente se llama “Administración de proyectos”, donde se aplica las actividades administrativas -como por ejemplo la planeación, coordinación, medición, monitoreo, control y reporte- para que surja un desarrollo y posterior mantenimiento de software que se debe llevar a cabo de manera disciplinada. [19]

2.2.2.1.8.1. Iniciación y Alcance

Las actividades acá relacionadas, se centran en determinar de manera veraz y eficaz los requisitos del software por medio de métodos de inducción y la valoración de la viabilidad que existe desde distintos puntos de vista. Luego de que se ha establecido la viabilidad, se prosigue con la especificación determinante de requisitos y del cambio de los procedimientos [37]

2.2.2.1.8.2. Promulgación del Proyecto de Software

La ejecución de los planes y difusión de procesos se realizan en esta parte, donde se hace una centralización de planes, con una expectativa agresiva que cumpla con los requisitos del contratista y el lograr los objetivos. Para la promulgación es de vital importancia medir, supervisar, registrar e informar. [37]

2.2.2.1.8.3. Revisión y Evaluación

La revisión se realiza en el proceso global del beneficio de los objetivos prefijados, así mismo se indaga de la satisfacción de los requisitos del contratista. Del mismo modo se hacen evaluaciones para la efectividad del asunto global de herramientas y métodos utilizados. [37]

2.2.2.1.8.4. Cierre

Cuando un proyecto llega a su fin, es porque todos aquellos planes propuestos inicialmente y procesos se han completado en su totalidad. En esta fase, se intenta hacer un breve repaso de todos los criterios que fueron base y éxito del proyecto. Ya que se determina el cierre, se producen actividades de archivado, post mortem y de mejoras de los procesos. [37]

2.2.2.1.8.5. Medidas de la Ingeniería del Software

- Establecer y Sostener el Compromiso de Medir. [37]
- Planificar el Proceso de Medición [37]
- Realizar el Proceso de Medición [37]
- Evaluar las Mediciones. [37]

2.2.2.1.9. Proceso de ingeniería de software

Esta área se enfoca, específicamente en la definición implantación, evaluación, mejora y administración del cambio de los procesos y actividades técnicas que deben realizarse para desarrollar mantener y retirar software. [19]

2.2.2.1.9.1. Proceso de Implementación y Cambios

Acá se hace hincapié en todos los cambios organizacionales donde se describe desde la infraestructura hasta los últimos modelos y que consideraciones prácticas puede llevar el desarrollo para implementación de cambios. [37]

Aquí se describe la situación en la que las técnicas se despliegan por primera vez, y donde se cambian los procesos vigentes (por ejemplo, metiendo una herramienta o mejorando un procedimiento). [37]

2.2.2.1.9.2. Definición de Procesos

Es un procedimiento o una política aplicada de manera estándar, donde los procesos del ciclo de vida se definen por muchas razones, donde se desea incrementar calidad del producto, facilitar entendimiento y comunicación y suministrar una guía automatizada para procesos y ejecución de apoyos. [37]

2.2.2.1.9.3. Valoración del Proceso

Para valorar un proceso, se debe llevar a cabo un modelo que en algunas instancias aprecia y evalúa las capacidades del software. [37]

2.2.2.1.9.4. Medición de los Procesos y Productos

La medición en los proyectos se puede hacer con el fin de apoyar la iniciación de unas fases de implementación y cambio, donde se deberá evaluar drásticamente las consecuencias de un proceso en la implementación a un cambio. [37]

2.2.2.1.10. Instrumentos y métodos de la ingeniería de software

2.2.2.1.10.1. Las Herramientas de Ingeniería de Software

- Las herramientas de Exigencias de Software: Aquí se usan instrumentos para acceder a exigencias del software donde habitualmente se catalogan en dos categorías: modelado e instrumentos de capacidad de rastreo. [37]
- Las herramientas Diseño de Software: Acaba se cubren instrumentos que ayudan a crear y comprobar todos los diseños de software. Sin embargo los resultados de dichos instrumentos son de gran variedad en cuanto a notaciones del diseño de software. [37]
- Las Herramientas de Construcción de Software: Aquí se nombran todos los instrumentos de construcción de software para producir y traducir la representación de un programa que posteriormente es ejecutado en una máquina. [37]
- Herramientas de Pruebas de Software: Son enumerados los generadores de pruebas, frameworks de ejecución de prueba, equipos de evaluación de prueba, instrumentos de dirección de prueba, instrumentos de análisis de Funcionamiento. [37]
- Herramientas de Mantenimiento de Software: herramientas de comprensión y herramientas de reingeniería. [37]

2.2.2.1.10.2. Los Métodos de la Ingeniería de Software:

Cuando se refieren a métodos de la ingeniería de software, se habla de tres temas: la heurística donde se permite accesos informales, los métodos formales que va de la mano con las matemáticas para cualquier teoría y los métodos de prototipado donde se esclarece diversos moldes o modelos que se ajustan a una teoría.

- Métodos heurísticos: es de donde nace todo lo relacionado con métodos que son estructurados, también los métodos orientados a datos y a su vez los métodos orientados a objetos.
- Métodos Formales: se procura en esta parte relacionar el software con teorías matemáticas basándose en la ingeniería, y a su vez se sub divide en aspectos como la especificación de lenguajes, el refinamiento de código y la propiedad de verificación.

- Métodos de prototipado: Los prototipados cubre los aspectos de modelo tras modelo para llegar a un punto final, en este punto se debe pensar en estilos, objetivos, y técnicas de evaluación dentro del mismo prototipado.

2.2.2.2. Paradigmas de ingeniería de software

2.2.2.2.1. Enfoque estructurado

Se hace referencia al enfoque estructurado en el momento de usar diagramas de flujo de datos los cuales son la principal herramienta para entender un sistema o negocio y así plasmarlo en código fuente. En dicho diagrama participan procesos como métodos, flujos de datos como argumentos y archivos como bases de datos. [20]

En este enfoque un cambio mínimo dentro del código, puede llegar a alterar el resto del programa, siendo una dificultad dentro del mismo desarrollo en tiempos y recursos. Además se debe analizar este paradigma en el momento de usarlo, ya que es difícil que un fragmento de código pueda servir en otro proyecto. [20]

Hasta el día de hoy es bastante difícil encontrar este tipo de enfoque, ya que se torna más complejo y de gran esfuerzo codificar una idea por este paradigma, sin embargo fueron las bases que permitieron avanzar a la forma de desarrollar un proyecto de forma más fácil y rápida. Algunos de sus componentes son: [20]

2.2.2.2.1.1. Diagrama de flujo de datos

Es la representación gráfica dentro de un determinado sistema donde comúnmente se puede utilizar para la visualización del procesamiento de información o datos por medio de entidades o agentes externos al mismo. [20]

2.2.2.2.1.2. Diccionario de datos

El diccionario de datos se conforma por un listado de manera ordenada de todos aquellos datos compuestos que pertenecen a un sistema. Como su principal objetivo es dar precisión sobre los datos que se manejan dentro de un sistema con el fin de evitar aquellas malas interpretaciones o ambigüedades de elementos que suelen presentarse. [20]

2.2.2.2.1.3. Diseño de modulo

El diseño de modelo de datos es un lenguaje de programación que se orienta a representar una base de datos donde podemos encontrar: [20]

- Las estructuras de datos de la base
- Las restricciones de integridad
- Operaciones y manipulación de datos

2.2.2.2.2. Enfoque orientado a objetos

El enfoque orientado a objetos, permite contextualizar un objeto en entidad la cual encapsula datos (atributos) y acciones o aquellas funciones que ayudan a manipular los datos (métodos). Hablando más técnicamente un objeto es definido como una instancia o particularización de una clase. Las funciones que provee un objeto son los datos que comúnmente se llaman atributos y las acciones que permiten dicha interacción entre clases. [20]

Los cuatro pilares principales del enfoque orientado a objetos son: [20]

- La abstracción: Consiste en apartar un elemento de su contexto para dar un énfasis en la pregunta ¿Qué hace? Mas no en el ¿Cómo lo hace?, por

esto, se dice que la abstracción expresa las características esenciales de un objeto las cuales lo distinguen de los demás o dicho de otro modo separa las características esenciales de las no esenciales.

- Encapsulamiento: define el ocultamiento del estado de un objeto, es decir, que cada objeto está aislado y se puede cambiar mediante operaciones definidas para acceder a él. Esto también ayuda a proteger la estructura de un programa para que no pueda ser modificado por cualquiera que no tenga acceso autorizado a dicho objeto.
- Modularidad: es aquella propiedad que permite dividir un programa en partes más pequeñas, en las que se busca que sea lo más independiente posibles de la misma aplicación. La modularidad permite la descomposición que ayuda a la seguridad de dicho código y a la optimización del mismo para un excelente rendimiento.
- Herencia: Esta proviene en el momento exacto desde que una nueva clase es creada donde contiene atributos y métodos de una clase primaria, sin poner impedimentos de compilación por usar sus elementos.

También se debe recordar que este paradigma usa características importantes como el polimorfismo el cual se ocupa de referenciar una clase con otra sin importar la dirección a clases derivadas, la concurrencia que permite a los procesos de los sistemas ejecutarse simultáneamente y poder interactuar entre ellos sin problemas o contradicciones entre sí; por último la persistencia que consiste en inicializar objetos con atributos predeterminados o atributos por defecto. [20]

2.2.2.2.3. Arquitectura dirigida por modelos (MDA)

Este paradigma se define como “un marco de trabajo de arquitecturas para desarrollo de software con tres objetivos principales: portabilidad, interoperabilidad y reusabilidad”. Por esta razón se debe recalcar que tiene una gran habilidad para observar el ciclo que compone de manera completa un desarrollo, es decir, cubre análisis, diseño, programación, pruebas, despliegue y mantenimiento. [8]

Esta arquitectura asegura aspectos funcionales como: [8]

- Portabilidad: Aumento del re-uso de aplicaciones, obteniendo reducción de costos y recursos.
- Interoperabilidad: Esto entre plataformas para que puedan usarse métodos inflexibles para responder a aquellos estándares que complementan tecnologías entre las mismas reglas del negocio.
- Independencia de plataforma: reduciendo en gran parte temas como el costo, el tiempo y la complejidad que se evidencia en diferentes tecnologías.
- Productividad: Permite a los todos los actores con el modelo tener libertad de uso de lenguajes y conceptos, facilitando la colaboración entre equipos.
- Puntos de vista del sistema: Esta arquitectura identifica tres puntos importantes dentro de un sistema
 - Independiente de cómputo (CIM): Se orienta dentro del contexto y requerimientos de una aplicación, sin preocupación de su distribución o proceso.

- Independiente de la plataforma (PIM): Hace énfasis en aquellos atributos estratégicos dentro del sistema sin pensar en una plataforma específica, mostrando por supuesto aquellas capacidades que son abstraídas.
- Dependiente de la plataforma: En este punto se agregan todas las características y detalles relacionados con la plataforma específica.
- Transformación de modelos: Proceso que convierte un modelo a otro dentro del mismo sistema.

2.2.2.2.3.1. Elementos de MDA

La integración e interoperabilidad depende del uso inteligente de los datos en las aplicaciones, plataformas y herramientas. Por lo que los estándares principales para cuantificar en MDA (UML, MOF, XMI Y CMW) son la base para construir un esquema coherente para publicar y administrar modelos sin importar el tipo de sistema que se va a construir. [8]

- Meta Object Facility (MOF). Se refiere a un lenguaje para que determine y explique modelos de meta-modelos, donde se consideran 4 capas: Un modelo indefinido de los objetos y las múltiples relaciones; Un grupo de reglas para esquematizar un meta-modelo a interfaces separadas del lenguaje; Algo importante son las reglas, definiendo el ciclo de vida del software, la composición y su semántica. Una clara jerarquía de interfaces definen las operaciones genéricas para así mismo intentar descubrir y manejar las propiedades de los meta-modelos basados en MOF [8].
- Common Warehouse Metamodel (CWM). Aquí se determinan interfaces que podrán ser utilizadas para habilitar intercambio de Metadatos entre almacenes de los datos y la parte de negocios de una aplicación [8].
- Unified Modeling Language (UML). Básicamente es el Lenguaje de Unificado (UML) el cual apoya y sirve como notación base para la definición de CWM [8].

2.2.2.2.3.2. Proceso MDA

Un sistema bajo los parámetros de MDA puede incluir múltiples modelos que se relacionan entre sí, bajo ciertas categorías de la abstracción, con mapas que interconectan unos modelos hacia otros que incluyen: [8]

- Un CIM donde se contiene todas aquellas reglas de negocio definidas.
- Un PIM donde se defina un modelo conceptual completo con todas aquellas relaciones.
- Múltiples PSM para crear mecanismos de cumplimiento y experiencias de los sistemas.

La idea central de MDA, consiste en utilizar dos conceptos únicos en su ejecución: Modelo y Transformación. Primero se debe construir un modelo conceptual que se debe expresar en modelo UML, el cual será transformado hacia varios PSM. Se puede hacer los cambios necesarios en el modelo conceptual, con el fin de conceptualizar correctamente en PSM. [8]

2.2.2.3. Metodologías de desarrollo

Es difícil pensar en un desarrollo de software y no poseer o seguir una metodología de desarrollo para alcanzar unos criterios o estándares que ayudan a que una aplicación sea correctamente construida. Esto justamente es a lo que

ayuda una metodología, a estructurar los procesos, planificar cada detalle y controlar la ejecución de desarrollo en un sistema de información. [21]

Para definir cada metodología de desarrollo, se debe hacer memoria de cuando empezó o un poco de su historia. Específicamente en los años 60 existía una gran demanda de programas a nivel empresarial, seguidos por una libre estructura anti metódica que seguía las etapas del ciclo de vida, por lo que dichos sistemas de información resultaban pesados para un óptimo procesamiento y unas rutinas de cálculo.[21]

Por consiguiente, se buscó la manera de construir una metodología que permitiera presentar un conjunto de técnicas, tanto tradicionales como modernas, hacia el modelado de sistemas que permitieran un desarrollo alto con respecto a la calidad del software. [21]

2.2.2.3.1. Tradicionales

Aquellas metodologías que cuentan con un mayor énfasis en la planificación y control del proyecto, además de una gran especificación precisa de requisitos y modelado, se conocen bajo el nombre de metodologías tradicionales o robustas

2.2.2.3.1.1. Prototipos

Se le conoce como el “modelo de desarrollo evolutivo”, a donde se inicia con el esclarecimiento de los objetivos completos para el software, luego se identifican aquellos requisitos que son conocidos del negocio y las áreas donde se necesita más definición. Este modelo es más que todo usado para dar una vista parcial de parte del software al usuario, integrando la prueba y el error en cada interacción, dado que si no se está de acuerdo con el prototipo, simboliza que la prueba fallo y se debe hacer correcciones del error que se obtuvo hasta que el usuario final quede satisfecho. [9]

Además de esto, se debe hacer una construcción del prototipo de manera rápida y eficiente, con el fin de no utilizar muchos recursos y/o dinero en casos más críticos, ya que a partir de esto se puede iniciar un verdadero desarrollo de alta calidad. Sin embargo, cabe mencionar, que no todos los desarrollos de software son aptos para experimentar como lo hace el prototipo. [9]

Las etapas que contiene dicha metodología son contempladas en: [9]

- Recolección y refinamiento de requisitos
- Modelado, diseño rápido
- Construcción del Prototipo
- Desarrollo, evaluación del prototipo por el cliente
- Refinamiento del prototipo
- Producto de Ingeniería

Los prototipos contienen grandes ventajas en el momento de empezar a desarrollar y es importante entender sus beneficios como: [9]

- Reduce gran parte del riesgo de hacer una construcción del producto que no satisfaga ciertas necesidades por parte de los usuarios.
- Baja costos y aumenta considerablemente la posibilidad de éxito.
- Demanda la disposición de herramientas adecuadas.
- Es muy útil en el momento que el cliente conoce o establece los objetivos generales para el software pero no hace identificación de los requisitos del

sistema de manera detallada tanto de entrada o de procesamiento o de salida.

Los tipos de modelo de prototipos se pueden presentar como: [9]

- Modelo de Prototipos rápido: Es aquella metodología de diseño que crea rápidamente los diseños nuevos, los valora y decide del prototipo cuando el siguiente diseño es desarrollado por medio de un nuevo prototipo.
- Modelo de Prototipos reutilizable: Se le conoce como "Evolutionary Prototyping"; donde claramente no se deja perder el esfuerzo efectuado en la construcción del prototipo ya que por sus partes pueden ser manejados para construir un producto real. Este es mayormente usado en desarrollo de software aunque también se puede usar para determinados productos como moldes de plástico para fabricación de objetos o hasta automóviles.
- Modelo de Prototipos Modular: se le puede llamar también como Prototipado Incremental (Incremental prototyping); En este se van agregando nuevos elementos sobre el prototipo inicial en el transcurso de que el ciclo de diseño va evolucionando.
- Modelo de Prototipos Horizontal: Esta intenta cubrir una gran parte de aspectos y funcionalidades que a su vez no son operativas; Aquí es muy fácil medir el alcance del producto pero muy inconsistente en el uso real.
- Modelo de Prototipos Vertical: Se puede hacer el cubrimiento de una pequeña parte de de las funciones operativas dentro del mismo. Eso si es bastante útil para evaluar un producto por pequeñas partes
- Modelo de Prototipos de Baja-fidelidad: Se debe implementar bajo papel y lápiz la función del producto real sin llegar a mostrar el resultado final o aspecto. Es bastante eficaz para hacer test en el producto
- Modelo de Prototipos de Alta-fidelidad: Se realiza este proceso de acuerdo a lo establecido en la teoría, donde se ejecuta de la forma más parecida al diseño real, dentro de las posibilidades de aspecto, impresiones, interacción y tiempo [9].

2.2.2.3.1.2. Cascada

El famoso modelo en cascada o modelo lineal secuencial es considerado un enfoque tradicional para el ciclo de vida del desarrollo de sistemas, donde se cree que es una forma de unir un desarrollo rígido, llevando a cabo secuencia de actividades consistentes para la evaluación de requerimientos, su diseño, una implementación, la posterior integración y un bloque de pruebas. [10]

Las fases que posee esta metodología, ayudan a seguir un correcto funcionamiento de los desarrollos: [10]

- Análisis de requerimientos, donde se examina y reúne aquellas necesidades de producto y por lo general su parte final es texto.
- El diseño detalla la organización del producto de manera interna y es representado con esquemas de flujo o texto.
- La implementación por su parte hace referencia a programación. Como producto final es el código en múltiples niveles incluidos del que se genera por sistemas de compilación automática.
- La integración es aquel proceso en el cual se ensambla los fragmentos para cumplir con un producto como fase final.

También se caracteriza fuertemente por ordenar de una manera inflexible las etapas que conforman el mismo ciclo de vida de un software, ya que al comienzo de cada etapa debe esperar a que su predecesora termine de manera exitosa para dar continuidad con el proceso, por tales razones el proceso es más fácil identificar todos los costos y plazos que se deban ejecutar dentro del mismo. [10]
Para tener claro, se puede decir que la metodología en cascada es esencialmente: [10]

- El inicio y el alcance del proyecto
- La planificación del proyecto (calendario, recursos necesarios, costo)
- Definición de las necesidades del negocio y el análisis en detalle de la solución
- La creación de la solución
- Prueba que la solución funciona. La entrega de la solución a su público objetivo
- Cierre del proyecto

2.2.2.3.1.3. RUP

Se dice que un Caso de Uso como un pedazo de funcionalidad de un determinado método que suministra al usuario un costo agregado, es decir, simbolizan los requisitos funcionales de dicho sistema. Por medio de RUP, no solo se saca los requisitos del sistema sino que a su vez, guía el diseño, la implementación y la prueba mediante un proceso centrado en la arquitectura [11].

Toda la arquitectura que se encuentra dentro de un sistema, es aquella parte fundamental de la organización de piezas importantes para así tener una visión entre las partes involucradas de desarrollo y usuarios y a su vez una perspectiva del sistema completo, para obtener control del desarrollo [11].

RUP se centra en utilizar aquellos casos de uso con el fin de gobernar el proceso que se facilita en especial del establecimiento de una excelente arquitectura que no se distinga por lo fuertemente impactada en posteriores cambios dentro del mismo desarrollo y el futuro mantenimiento. [11]

Esta metodología pesada posee un proceso iterativo o incremental, donde el equilibrio correcto se encuentra entre los casos de uso y la arquitectura que suele asemejarse con la estabilidad de la forma y función del producto, lo cual se consigue con un determinado tiempo. Una estrategia para lograr esto, es tener un proceso que sea iterativo e incremental ya que ayuda a dividir el trabajo en partes más pequeñas o mini proyectos durante el desarrollo. [11]

Cada parte pequeña se puede observar como una iteración, es decir, un recorrido completo que se hace a través de los flujos de trabajo presentes, donde se obtiene un incremento que produce un estímulo de crecimiento en el objetivo o producto final. [11]

Una iteración se debe hacer a través de una cascada, donde básicamente recorre flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y pruebas), Luego realiza una planificación de la iteración, un posterior análisis y otras actividades adicionales de la iteración en específico. Para terminar se hace una composición de resultados obtenidos con respecto a las iteraciones pasadas. [11]

Como se observa, las iteraciones son el corazón o la parte fundamental de esta metodología por ello se analiza cada que se termina una con el fin de buscar posibles futuros errores o para determinar si los requisitos establecidos inicialmente han cambiado o existen nuevos para implementar. [11]

RUP se divide en ciertas partes fundamentales realizando varias iteraciones: [11]

- Las fases de inicio se focalizan al entendimiento de un problema y la tecnología que lo rodea, así como la delimitación donde se va a desarrollar.
- En la elaboración se establece una línea de arquitectura, eliminación de riesgos críticos y bases fundamentales como modelado de negocio y requisitos. Además las iteraciones se orientan al desarrollo abarcando cada vez más requerimientos y líneas de desarrollo.
- En la parte encargada de la construcción, se realiza la construcción y terminación del producto final por una serie de requisitos ya establecidos.
- La fase de transición se pretende garantizar por medio de pruebas que el producto si está apto para ser entregado y usado por una comunidad de usuario.

Dentro de la estructura del proceso de RUP, encontramos una serie de pasos que se pueden considerar por medio de 2 ejes: [11]

- Horizontal: Representa el tiempo y es usado para poner características dinámicas dentro del proceso, indicando las particularidades del ciclo de vida que lo compone, siendo expresado en termino de fases. Inicio, Elaboración, Construcción y transición.
- Vertical: Simboliza las partes del proceso que son estáticos. Representa el proceso en términos de proceso, disciplinas, flujos de trabajo, actividades, instrumentales y roles.

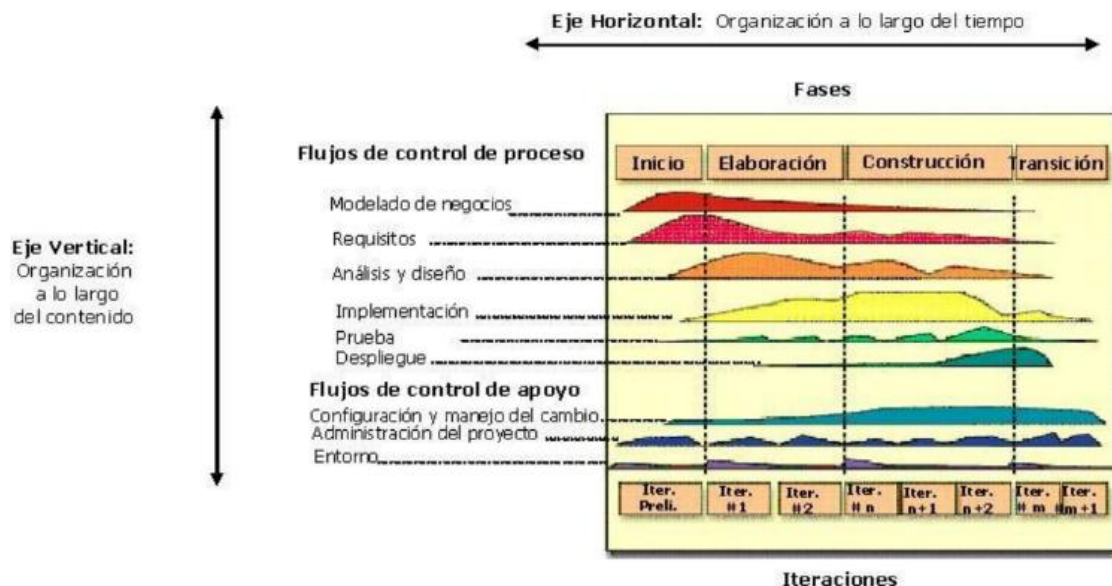


Figura 1. Proceso RUP. Disponible en: <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf>

2.2.2.3.2. Agiles

El desarrollo ágil de software se ha vuelto muy común para realizar aplicaciones de una manera más rápida y precisa, que además permita establecer una fácil cooperación entre los integrantes del equipo desarrollador y dicho software a realizar. Además toma un enfoque iterativo e incremental, el cual permite que los requisitos sean soluciones evolucionables según demanden las necesidades del proyecto. De este modo, el trabajo es realizado mediante colaboración de equipos que se auto-organizan de manera disciplinada donde toman decisiones en el corto plazo para beneficio del desarrollo del software.

Cada metodología ágil posee unos procesos similares, donde se usa: la planificación, el análisis de requisitos, el diseño, la codificación, las respectivas pruebas y un documento final.

2.2.2.3.2.1. Scrum

Esta metodología es un proceso donde se aplican un conjunto de “buenas practicas” con el fin de trabajar de manera colaborativa, o grupalmente, y obtener el mejor resultado posible para un proyecto. Su origen parte de estudios realizados en la manera como se trabaja colaborativamente de una manera productiva y rápida para cada fase. [23]

Acá se realizan entregas parciales y regulares de un producto final, donde se prioriza el beneficio que aporta el receptor del proyecto, por lo que es especialmente indicado para proyectos que están en un ambiente complejo de desarrollo, donde se busca además un resultado rápido, donde los requisitos cambian rápidamente o están poco definidos y donde la innovación, la productividad, la flexibilidad y la competitividad son altamente fundamentales. [23] También lo que busca solucionar SCRUM es resolver situaciones donde no se le hace entrega de lo que el usuario está deseando realmente, cuando se alargan dichas entregas, cuándo los costos son demasiado altos, cuando la calidad no es la pertinente con respecto a los estándares dentro del software, cuando la competencia es mejor o la moral de los equipos es baja y rotativa. [23]

El proceso de esta metodología es simple, donde se ejecuta en bloques temporales buscando en lo posible que sean cortos y fijos (iteraciones que sean de 2 semanas máximo), cada iteración proporciona un resultado completo, un aumento de producto final que sea sensible de ser facilitado con un esfuerzo muy bajo si el cliente llega a pedirlo. [23]

Las actividades que se ejecutan en dicho proceso son: [23]

- Planificación de la iteración: Desde el inicio se define la planificación de la iteración, donde posee selección de requisitos y elaboración de tareas de la iteración a realizar en un tiempo establecido.
- Ejecución de la iteración: Todos los días el equipo realiza una reunión de máximo 15 minutos, donde se inspecciona el trabajo que se está realizando por si se debe hacer adaptaciones o mejoras dentro del mismo. Adicionalmente existe un facilitador (Scrum Master) que se encarga de que el equipo cumpla con sus metas establecidas al corto plazo y no bajar la productividad.

Adicional a esto el cliente acompaña cada iteración para refinar los requerimientos y planificar de manera correcta los objetivos del proyecto y que se cumplan al terminar el desarrollo final.

- Inspección y adaptación: Esto se realiza en el último día de la iteración donde se realiza una reunión de revisión donde se presenta al cliente los requisitos completados hasta el momento y el cliente realiza adaptaciones de manera objetiva. Adicional se tiene una retrospectiva en esta parte para analizar cómo ha trabajado el equipo.

2.2.2.3.2.2. XP

La metodología XP por sus siglas (eXtreme Programing) donde se sigue un conglomerado de reglas para el desarrollo de software, basándose en diferentes ideas de cómo enfrentar un ambiente cambiante y a su vez planificar, analizar y diseñar para el corto plazo y hacer todo por pequeñas partes a la vez que se va desarrollando el software. [24]

Acá se manejan conceptos básicos para aplicarlos a su metodología como tal y manejarlos como aspectos: [24]

- Simplicidad: es la base de dicha metodología y se refiere a simplificar el diseño para agilizar el desarrollo de la aplicación y a su vez facilitar un mantenimiento. Para dicha propiedad es necesario que se maneje refactorización del código para mantenerlo simple a medida que crece.
- Comunicación: Se puede realizar de diferentes formas, pero en este caso se refiere a que el código simple comunica mucho mejor. Como ayuda puede ser autodocumentado y que sea inteligible.
- Retroalimentación: Como el cliente esta emergido en el proyecto de una forma u otra, su opinión se conoce en tiempo real, ayudando a los desarrolladores de software a centrarse en lo importante y lo que el cliente desea.
- Coraje o Valentía: Muchas prácticas requieren valentía. Por ejemplo diseñar y programa para hoy y no para mañana. Significa aquel esfuerzo que se necesita en tiempo y trabajo para implementar el resto del proyecto.
- Respeto: Los integrantes del equipo se respetan de todas las formas posibles, tanto en desarrollo de código que permita al otro integrante trabajar de forma fácil hasta respetar el trabajo de los demás.

Como características principales, posee: [24]

- El desarrollo iterativo e incremental o por pequeñas partes hasta llegar al final
- Las continuas pruebas unitarias que permiten automatizar el software
- La programación en parejas, permite tener mayor calidad del desarrollo.
- La integración del equipo con el cliente ayuda a trabajar mejor y guiado hacia un objetivo final y exitoso.
- Refactorización del código permite aumentar su legitimidad y mantenimiento, esto sin modificar su funcionamiento esencial.
- Simplicidad del código sirve para que las cosas funcionen bien y de manera rápida u óptima.

Adicional a todo esto existen unos roles que se determinan desde la primera reunión, con el fin de posicionar cada integrante de manera correcta frente a los demás y así mismo no se ejecuten tareas de manera errónea o repetida: [24]

- Programador: produce el código y lo prueba
- Cliente: Decide que se va a realizar por medio de historias de usuario y hace al final de cada iteración pruebas funcionales.
- Tester: ayuda al cliente y al equipo a realizar las pruebas pertinentes.
- Tracker: Es el encargado del seguimiento, da retroalimentación y estima aciertos de las tareas desarrolladas
- Entrenador: lleva a todos los del equipo para que hagan el proceso correctamente.
- Consultor: Es un miembro externo que ayuda al equipo con conocimiento para resolver un problema específico.
- Gestor: Es el empresario de la organización y establece la relación entre clientes y programadores.

2.2.2.3.2.2.1. Fases

- **Fase de Exploración:** Por lo regular se toma semanas o meses, dependiendo del tamaño y familiaridad que se tenga en el grupo de desarrollo, donde también se permite familiarizarse con las herramientas, tecnologías y prácticas que se utilizaran en el transcurso del proceso de creación del software. [35]
- **Fase de planeamiento:** Los desarrolladores consideran el esfuerzo y tiempo que cada historia requiere para así mismo definir el cronograma. En este punto se hace un reléase el cual no supera los dos meses de duración y se debe tener en cuenta varias iteraciones para el mismo. [35]
- **Fase de producción:** Requiere prueba y comprobación del funcionamiento del proyecto antes de que esta parte sea mostrada al cliente. Esta fase se realiza de una a tres semanas, donde si salen sugerencias se anotarán con el fin de relacionarlas posteriormente. [35]
- **Fase de mantenimiento:** Aunque es una fase de mayor esfuerzo, se centra en solucionar las tareas del cliente, donde pueden existir cambios internos para un mejor ajuste. [35]
- **Fase de muerte:** Aquí el cliente ya no posee más historias de usuario para incorporarlas al desarrollo, por lo que las necesidades del cliente han sido satisfechas en características como el rendimiento y confiabilidad la documentación final será estable, es decir, no existirán más cambios a la arquitectura. [35]

2.2.2.4. Técnicas y herramientas

2.2.2.4.1. Control de versiones y repositorios

El control de versiones es un trabajo de los múltiples cambios que suelen realizarse en las partes que conforman algún producto o la configuración interna de dicho elemento. Siendo más específico, se puede decir que es una

revisión, una versión o edición del producto será el estado en el que se encuentre el desarrollo o la modificación final. [25]

Ya que un sistema de control de versiones se puede hacer de forma muy manual, se prefiere que se usen instrumentos que faciliten dicha gestión, dando así lugar para los llamados “sistemas de control de versiones”, los cuales facilitan la administración y posibles especializaciones realizadas del producto desarrollado. [25]

Para que se pueda versionar de manera correcta un desarrollo, se debe proporcionar aspectos como: [26]

- Un método de almacenamiento de los elementos que se van a gestionar
- La posibilidad de hacer cambios sobre elementos que ya son existentes o almacenados
- Llevar un registro histórico de las acciones que se realizaron sobre cada elemento con la propiedad especial de poder volver a un estado anterior del producto versionado.

Por consiguiente, una revisión se define como aquella versión explícita de información que gestiona o procesa datos con el fin de exponerla como producto final o iteración y avance a un producto específico. Para que se pueda contrastar una revisión concreta se realizan rótulos o tags los cuales ayudan a dar una versión de cada uno de los ficheros con el nombre común de una etiqueta y se contenga un orden específico de cada parte. [26]

Por el control de versiones se puede seguir un camino diferente o una llamada “branch” o rama en inglés, que ayuda a dar una copia exacta de lo que se tiene hasta el momento la cual evoluciona de manera diferente pero bajo la misma idea de desarrollo. [25]

Los repositorios ayudan a su vez a exportar elementos o crear arboles de directorios limpios sin metadatos, usándose a menudo para la publicación de contenidos. Así mismo posee importación que copia un árbol de manera local. [26]

Un término importante es la integración o fusión, donde se pretende unir dos conjuntos de cambios sobre un mismo archivo de una revisión, lo que puede suceder cuando un usuario actualiza su archivo inicial o copia local con los cambios finales añadidos recientemente por otros usuarios. Sucede que el código ha sido ramificado por desarrolladores de forma independiente a lo que en dado momento es necesario integrar todos aquellos campos que se realizaron de manera autónoma. Por estas razones unificar o fusionar es uno de los términos más importantes, porque por medio de ellos lo podemos convertir en un solo producto y crear las diferentes versiones. [25]

Por supuesto, existen diferentes formas de colaboración dentro de un proyecto y de forma grupal: [25]

- De forma exclusiva: es la forma como se puede realizar un cambio en el repositorio y dicho sistema se encargara de bloquearlo en el momento de la actualización para que en ningún otro usuario pueda usarlo al mismo tiempo y en el momento de finalizar la modificación este queda libre para los demás.
- De forma colaborativa: Cada usuario modifica la copia local y cuando un usuario de los participantes publica, el sistema realiza los cambios de

manera automática para combinar todas las diversas modificaciones de los demás usuarios. El verdadero problema es la aparición de conflictos que se solucionarían únicamente de forma manual entre archivos duplicados, ausencia de cambios, etc.

La arquitectura es un punto fundamental dentro de los repositorios, ya que permite a los sistemas de control versionar, atendiendo a su estructura de almacenamiento de código: [25]

- Centralizado: Existe un repositorio central de todo el código consumido por los diferentes usuarios dentro del sistema, donde se facilitan tareas administrativas pero poseen problemas de flexibilidad.
- Distribuido: Cada usuario contiene su repositorio que permite intercambiar y mezclar revisiones entre ellos de lo que se lleve desarrollado. En este caso es frecuente un punto central donde normalmente está disponible un repositorio que es básicamente un centro de sincronización de los distintos repositorios de cada usuario.

2.2.2.4.2. Pruebas automáticas

Las pruebas dentro del software constituyen una parte importante de lo que es tener una aplicación final con un gran porcentaje de optimización, ya que al cliente o usuario final lo único que le importa es que funcione correctamente. Por ende aplicar pruebas de software es lo más efectivo para llegar a una alta calidad en cuanto al desarrollo y su función. [27]

Las pruebas que se realicen sin importar la perspectiva, determina lo rápido que se realiza una tarea, bajo el análisis de ciertas condiciones de trabajo. Además de todo esto, valida y verifica otros factores de la calidad del sistema como la escalabilidad, la fiabilidad y el uso de recursos. [27]

2.2.2.4.2.1. Pruebas de carga

Estas pruebas se refieren al máximo de capacidad que puede soportar un servidor web, tanto en su software como hardware, con el fin de atender un gran número de usuarios de manera simultánea, entonces la finalidad de estas pruebas son comprobar, todo componente que tenga relación para cuando este en pleno funcionamiento. [27]

Para realizar este tipo de pruebas, se debe tener en cuenta aspectos como: [27]

- Tiempo que tarda los usuarios en acceder a los datos
- La cantidad de datos y procesamiento de los mismos.
- Cantidad de archivos solicitados y el tiempo usado para transferir un dato.
- Tiempo que tarda el software para dar respuesta después de un clic el usuario
- Niveles que se presenten tras realizar interacciones entre el software.

2.2.2.4.2.2. Pruebas de integración

Las pruebas de integración permiten realizar testeos a aquellos elementos que componen un proceso. Aquí los módulos de software son combinados y probados como un grupo, y por lo general se realizan luego de las pruebas unitarias con el fin de no dejar nada al aire libre. [27]

Algunos consideran estas pruebas como una técnica para la construcción del programa mientras que a su vez se llevan pruebas para detectar múltiples errores dentro de cada iteración. [27]

Acá se encuentra dos tipos de integraciones: [27]

- Integración descendente: Se integra los módulos que posea hasta el momento el desarrollo, por medio de la jerarquía del control, Donde inicia por el control principal y termina incorporando pruebas de subordinados.
- Integración ascendente: Empieza la prueba desde los desde los niveles más bajos y luego realiza integración de todos los módulos hacia arriba.

2.2.2.4.3. Lenguajes de programación

Un lenguaje de programación es una plataforma de lenguaje el cual permite diseñar procesos que de una forma u otra son ejecutados o accionados por maquinas como las computadoras. Estos pueden usarse para crear programas de manera sistemática que controlen el comportamiento lógico y físico, si aplica, definiendo el significado de sus elementos y expresiones. Aquel proceso que prueba, depura, compila y se mantiene un código fuente de un programa se le llama programación. [28]

Mediante un lenguaje, se puede desarrollar un software o su esencia física de funcionamiento, por lo que se debe considerar elementos tan importantes como: [28]

- Variables: Las variables son aquellos espacios en memoria donde se almacena datos específicos de un conjunto de órdenes o líneas de código que conforman un programa. Por medio de estas podemos trabajar un software para que asigne y realice acciones determinadas, como por ejemplo una simple suma.
- Condicionantes: Son estructuras de código que permiten limitar o restringir que parte del programa se ejecute para que de esta forma se cumplan otras acciones o premisas. Acá encontramos el “IF” y “ELSE” que son de uso universal para realizar lo que se desee.
- Bucles: Son parecidos a los condicionantes en cuanto a función, con la gran diferencia de que se ejecutan constantemente hasta que se cumpla una condición impuesta. Acá existen el “FOR” y el “WHILE” que ayudan a hacer esa ejecución repetida.
- Funciones: Existen para evitar repetir constantemente pedazos de código que se van a reutilizar más de una vez que permite ejecutar un proceso.
- Sintaxis: La mayoría de lenguajes usan secuencias de texto que incluye palabras, números y algunos signos de puntuación únicos y necesarios para la ejecución.

Todos esos elementos escritos bajo una sintaxis específica, tiene que dar como resultado una salida o implementación, la cual de manera más técnica ayuda que se ejecute dicho programa, combinando software y hardware. Por lo que existen dos formas de implementar: [28]

- Compilación: Este proceso es la traducción de un programa escrito en un lenguaje de programación a otro lenguaje de programación, dando como salida un equivalente a lo que la maquina es capaz de traducir e interpretar. Por ende un compilador puede realizar esta operación generando muchas más líneas de código que las escritas inicialmente.
- Interpretación: Se asigna los significados a las fórmulas que no son más que cadenas de símbolos sin ningún significado.

Para esta investigación se hace énfasis a los lenguajes de programación de dispositivos móviles, donde ayudan a que la tecnología que existe hoy en día sea traducida a código y a acciones comunes y cotidianas donde se puede hacer una sin número de tareas. [28]

2.2.2.4.3.1. Plataforma para Windows Phone

La plataforma usada para desarrollo de software es Visual Studio de Microsoft, donde se basa en lenguajes de programación como C# y visual Basic .NET, ambos bajo el framework de .NET. Es dinámico utilizando como diseño la herramienta Silverlight o el conocido XAML el cual permite codificar toda la parte visual del desarrollo sin pensar tanto en la estructura principal. Para los videojuegos se conoce una herramienta llamada XNA, la cual permite generar graficas de 2D y 3D donde pueden hacer uso de Silverlight al mismo tiempo. [29]

Como herramientas de programación, existen el Visual Studio en sus diferentes versiones, las cuales permiten crear código para cualquier plataforma, Web, Móvil, de Escritorio, etc. Para desarrollo de aplicaciones de Windows Phone se necesita El Windows Phone SDK 7.1 o versión superior y el Silverlight Toolkit para expansión de herramientas. [29]

2.2.2.4.4. Sistema de base de datos

“Una base de datos es un sistema informático a modo de almacén. En este almacén se guardan grandes volúmenes de información. Por ejemplo, imaginemos que somos una compañía telefónica y deseamos tener almacenados los datos personales y los números de teléfono de todos nuestros clientes, que posiblemente sean millones de personas”. [12]

De acuerdo a esta información se sabe que se está hablando de veinte o treinta datos multiplicados por miles o unos cuantos millones de personas. Anteriormente la gestión de bases de datos se basaban en archivos planos informáticos, pero con el gigantesco crecimiento de información que se genera hoy en día, se necesita de lo que se denomina sistema de base de datos. Es por esto que se llega a la conclusión de que una base de datos es para automatizar procesos en el momento de acceder a la información y así mismo accederla de manera rápida y fácil. [12]

Entre las principales características de las bases de datos, se puede mencionar: [12]

- Independencia lógica y física de datos
- Redundancia o repetición de campos de forma mínima
- Acceso múltiple por parte de los usuarios
- Integridad en los datos
- Consultas que permiten recuperar la información como la necesite agrupar
- Seguridad en el acceso
- Back up de respaldo y posterior recuperación en los datos
- Acceso por medio de lenguajes de programación específicos

Tipos de campos

Cada sistema o base de datos lo conforma una serie de tipos de datos especiales que pueden ser muy parecidos o diferentes entre sí. Podemos encontrar: [12]

- Numérico: pueden ser enteros, decimales, etc.
- Booleanos: Poseen el estado true o false.

- Memos: Campos alfanuméricos de longitud ilimitada.
- Fechas: almacenan fechas facilitando su manejo entre datos y consultas
- Alfanuméricos: Contienen cifras y letras
- Autoincrementales: Campos numéricos que incrementan con cada registro que agregan a la base

Modelo Entidad- Relación

Este diagrama se encarga de modelar la base de datos de un sistema de información, con el fin de tener un mapa y organizar de una manera jerárquica y detallada todos sus componentes. Este modelo permite representar cualquier tipo de abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos que se determinan como entidades y relaciones, incorporando además una vista visual del diagrama en construcción. [12]

Posee conceptos importantes como: [12]

- Entidad: Es cualquier objeto o elementos que se presentan en el contexto al que haga referencia los datos; estas son las que constituyen las tablas de la base de datos donde se almacena información.
- Atributos: Son esas características o rasgos que poseen una entidad
- Relación: Vinculo que define una dependencia entre dos o más entidades
- Interrelación: Constituyen los vínculos entre entidades, pero de forma tal que representa las relaciones de forma efectiva, a través de un campo clave.
- Entidades fuertes: Es conformada por tablas principales donde esta los registros principales del sistema de información y que, por lo general, requieren de entidades o tablas auxiliares.
- Entidades débiles: Son las entidades débiles de las tablas auxiliares de una tabla principal.
- Clave: Es un campo de una entidad el cual tiene como objetivo distinguir cada registro del conjunto.
- Integridad Referencial: Es el tipo de relación que se puede producir entre tablas mediante un campo clave.

Tipos de relaciones

Es simplemente la forma de cómo se relacionan las entidades o la expresión de cómo se relacionan con otras entidades. Estas pueden ser: [12]

- Según cardinalidad: La cardinalidad se presenta por el diagrama Entidad relación, como una etiqueta que se ubica por lo general en los extremos de la línea de las entidades y que puede contener varios valores entre los que se pueden usar el 1 y el * obteniendo relaciones como:
 - Relación 1 a 1: relación uno a uno.
 - Relación 1 a *: la relación uno a varios.
 - Relación * a *: la relación de varios a varios.
- Según modalidad:
 - Optativa: Es la relación entre una tabla y varios de la tabla relacionada
 - Obligatoria: La relación entre un registro de una tabla y otro de la tabla relacionada.

2.2.2.4.5. Entorno de desarrollo

2.2.2.4.5.1. Windows Phone

El sistema operativo Windows Phone es desarrollado por Microsoft como heredero de Windows Mobile, enfocándose en el mercado de consumo personal y no empresarial. Su interfaz de usuario es totalmente renovada e innovadora que integra múltiples servicios como OneDrive, Skype y Xbox Live ya que desea competir directamente con Android de Google e iOS de Apple. [30]



Figura 2. Dispositivo Windows Phone. Disponible en <http://www.taringa.net/comunidades/winphone/7240486/Aporte-Juegos-Windows-Phone-7-Act-30-05-2014.html>

Windows Phone se lanzó el 15 de febrero de 2010 en el Mobile World Congress con el objetivo de ser una alternativa de los sistemas operativos móviles con una interfaz nueva denominada “Metro”; esta interfaz gráfica se basa en pequeños mosaicos que muestran información útil al usuario y donde se puede tener mayor navegabilidad con el mismo sistema operativo de manera ligera. [30]

Posee un concepto nuevo llamado “HUB”, donde se puede centralizar acciones y agrupar aplicaciones por actividad, como por ejemplo tener el HUB de imágenes, de contactos, de Office o de Xbox Live; En este sentido la libreta no es solo para guardar direcciones, también muestra las últimas noticias de los contactos como publicaciones, tweets y hasta sus fotos. [30]

2.2.2.4.5.2. Principales características

Entre sus componentes, se encuentra la pantalla de inicio hecha por Live Tiles, mosaicos dinámicos que en términos reales y simples son enlaces a aplicaciones u objetos individuales como puede ser contactos, páginas web o archivos

multimedia. Estos mosaicos se actualizan permanentemente, conservando información para el usuario con las llamadas, mensajes, correos electrónicos, citas, juegos o nuevas aplicaciones. [30]

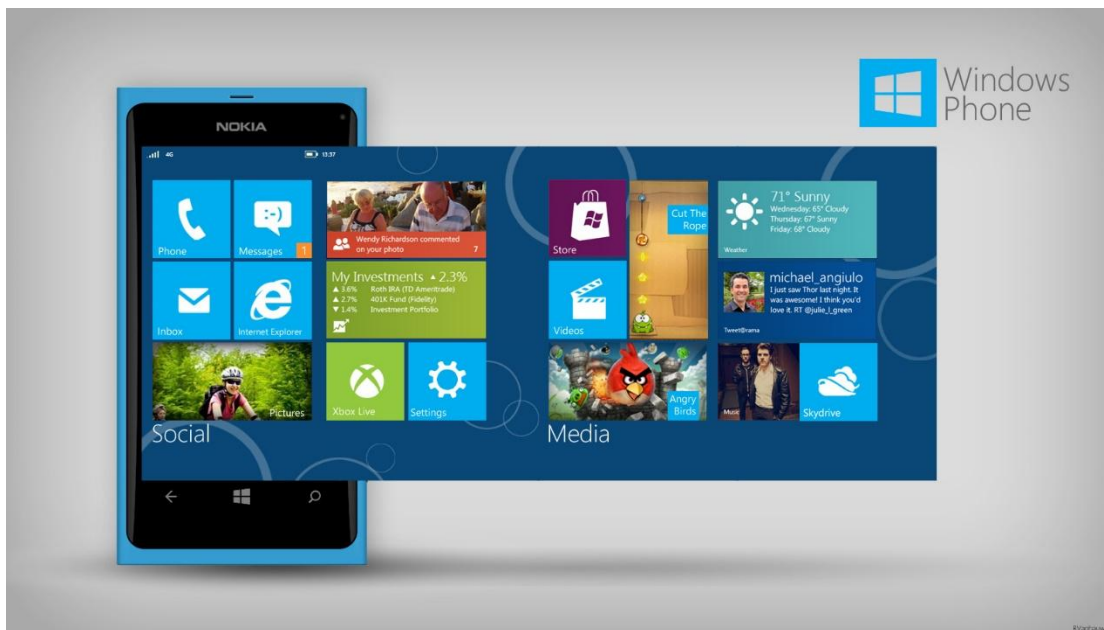


Figura 3. Plataforma Windows Phone. Disponible en: <http://antiviruspara.net/wp-content/uploads/2015/07/Windows-Phone-82.jpg>

El teclado es táctil en la pantalla, donde incluye el World Flow, revisión ortográfica, predicción de palabras y una tecla dedicada a la inserción de emoticones. Tiene un motor de búsqueda llamado Bing para contenido web además de tener su buscador por defecto, el famoso Internet Explorer. [30]



Figura 4. Interfaz Windows Phone. Disponible en: En línea:
<https://ciscoperez.wordpress.com/2011/08/03/predeterminar-teclado-numerico-en-windows-phone-7/>

Los dispositivos de Windows Phone poseen un botón de búsqueda ya sea para buscar en internet, escanear y traducir textos, leer códigos QR, búsqueda de libros, reconocimiento de canciones y ubicaciones. [30]

Su más novedoso concepto, los Hubs, clasifican acciones y agrupan las aplicaciones que correspondan a actividades determinadas. De este modo vale destacar que hay hubs de contactos, Office, fotos, podcasts y hasta juegos con la plataforma de Xbox. [30]

2.2.2.4.5.3. Tienda

Por medio de la tienda de Windows Phone que es totalmente web intenta que los usuarios descarguen y usen aplicaciones para juegos, música, videos, etc. Esto es por decirlo en términos de Play Store de Andorid o incluso el App Store de Apple. Microsoft gestiona la entrada de cada aplicación a su tienda de ventas, es decir, no permite contenido que incluya discriminación de ningún tipo, uso de drogas o material sexualmente sugestivo. [30]



Figura 5. Tienda De Windows Phone. Disponible en:
<http://faqswindowsphone.com/como-instalar-aplicaciones-en-tu-windows-phone/>

Hoy en día posee una plataforma de distribución digital no únicamente para dispositivos móviles, sino para aplicaciones con estilo Modern UI o interfaz netamente gráfica para sistemas operativos como Windows 8 o Windows 10. [30]
Se busca que los usuarios adquieran cualquier tipo de programa o aplicación para sus actividades diarias, por medio de la plataforma donde Microsoft puede escanear y buscar fallos de seguridad y programas maliciosos. [30]

Para aquellas personas que deseen desarrollar en esta tecnología a nivel mundial o subir sus proyectos/aplicaciones al Windows store, los beneficios son del 30% para Microsoft y el 70% para el proveedor de aplicaciones. Si las ventas aumentan a USD \$25,000 en ingresos, se redistribuirá los porcentajes de ganancias a un 80% para el desarrollador. [30]

Para las restricciones existentes, Microsoft regula dentro de los usuarios que suben aplicaciones para la venta en su tienda, aquellas consideradas como prohibidas por: [30]

- Contenido para adultos o cierto grado PEGI 16, ESRB Mature.
- Inciten a discriminación, odio o violencia basándose en determinado grupo social, racial, étnico, nacional, lingüística, religiosa o de otra índole.
- Contenido que estimule o incite a actividades ilícitas.
- Contenga difamación, calumnias o amenazas
- Promueva el uso irresponsable de drogas, tabaco o alcohol.
- Contenga o apoye la violencia extrema y blasfemia excesiva

2.2.2.4.5.4. Arquitectura de Windows Phone

El modelo desarrollado por Windows Phone, es una mezcla entre las arquitecturas de iPhone y Android, por ello Microsoft quería asegurar la misma experiencia para un terminal que ejecute Windows Phone que posea características mínimas. En los modelos desarrollados los usuarios obtendrán la misma sensación de uso además de que los desarrolladores sabrán que las aplicaciones funcionaran de forma idéntica en todos los dispositivos y no se encierra a un solo hardware. [30]

Por lo anterior, Microsoft desarrollo unas especificaciones iniciales para que los fabricantes de terminales que quieren incluir Windows Phone debe guiarse por lo siguiente: [30]

- Procesador Qualcomm Snapdragon S4 de Dual Core a 800 MHz o superior (hasta quad core)
- Mínimo 256/512 MB de RAM para teléfonos WVGA; mínimo 1GB RAM para teléfonos 720p, WXGA o 1080p
- Mínimo 4/8GB de memoria interna
- GPS y A-GNSS mínimo
- Soporte para micro USB 2.0
- Cámara trasera obligatoria, con autoenfoco(opcional) y mínimo VGA (flash opcional)
- WiFi 802.11b/g y Bluetooth, acelerómetro, sensores de proximidad y luz, motor de vibración
- Pantalla multitáctil capacitiva (mínimo cuatro puntos)

Con respecto a su arquitectura, Windows phone cuenta con 4 capas esenciales para las aplicaciones sean ejecutadas dentro de su sistema operativo: [30]

1. Hardware: En esta capa se representa cualquier hardware de dispositivo móvil en el que se encuentra instalado y funcionando el sistema operativo.
2. Kernel: En la capa del núcleo se encuentran los drivers, el sistema de archivos, las redes, el sistema renderizado, los gráficos, el sistema de Update o actualizaciones, entre otros.

3. Modelos: En la capa de modelos se encuentran tres modelos los cuales conciernen a: [7]
- *Modelo de aplicaciones*: Las aplicaciones se deben ejecutar bajo un fichero con extensión .xap, el cual contiene y establece los recursos válidos para que la aplicación se ejecute.
 - *Modelo UI*: hace referencia a la interfaz gráfica de usuario donde se compone de elementos y una sesión específica de interacciones que el usuario final realiza sobre una aplicación hasta llegar a implicar una o muchas aplicaciones de más.
 - *Integración con la nube*: En este punto se orienta integración o unificación del dispositivo con servicios como Exchange, Google Mail, Hotmail, Xbox Live, SkyDrive, Facebook, Twitter, LinkedIn, y Bing.[6]
4. Application Runtime: Esta capa permite y se ejecutan todas las aplicaciones del sistema operativo.[7]

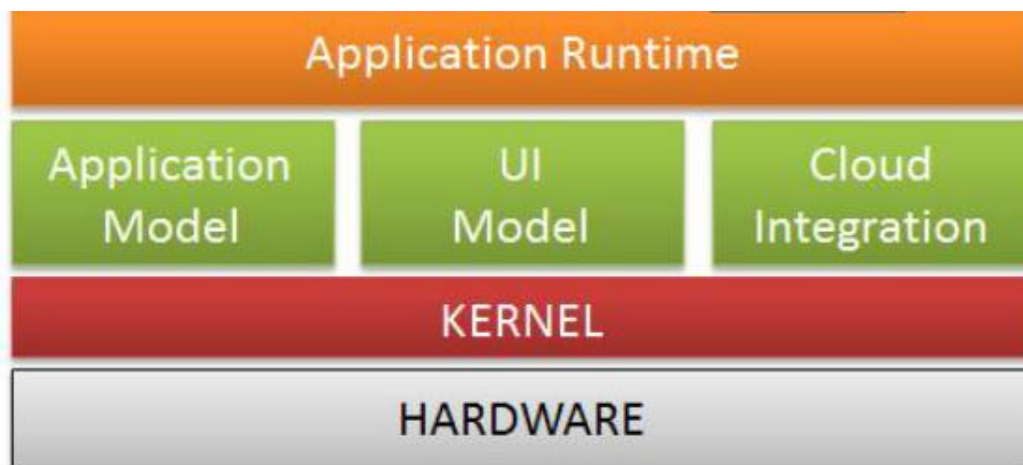


Figura 6. Capas de Aplicación S.O. Windows Phone. Disponible en: http://datateca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_13_desarrollo_de_aplicaciones.html

2.2.2.4.5.5. Versiones

Windows ha pasado por diferentes actualizaciones dentro de su creación como marca y sistema operativo para teléfonos inteligentes Windows Phone. Su descarga y actualización es similar a los demás sistemas operativos existentes en el mercado actual. [6]

Las versiones que existen son:

Windows Phone 7:

Esta versión fue la primera, género y se basa una interfaz llamada Modern UI, que luego fue implementada en Windows 8 y sus respectivas versiones. La Modern UI o llamada antes *Metro* fue desarrollada por Microsoft donde contiene una interfaz plana con colores básicos y diseños geométricos con movilidad vertical en el caso de los dispositivos móviles, la cual está optimizada para pantallas táctiles y posee iconos múltiples para explorar las diferentes funcionalidades del dispositivo. [6]

Esta versión está basado en CE (Windows Embedded Compact) el cual está desarrollado para sistemas embebidos, donde prácticamente se realizó una transición para migrar de Windows Mobile 6.0. Acá se generan aplicaciones con extensión .appx y .xap: [6]

Versión	Descripción
Windows Phone 7.0 ("Photon")	Versión Inicial
Windows Phone 7.1 ("Nodo")	Esta actualización permitía copiar y pegar, además redujo el tiempo de encendido del dispositivo, aumento rendimiento de las aplicaciones y
Windows Phone 7.5 ("Mango")	Empezó a funcionar en el 2011 donde apporto más de 500 funcionalidades como multitarea, procedimiento de búsqueda, integración con redes sociales, aplicaciones como office y Skydrive, Aumento características del posicionamiento Global (GPS) y en el navegador Internet Explorer 9.
Windows Phone 7.5.1 ("Refresh")	Dirigido a reducir características del Sistema Operativo para adecuar a sistemas operativos de bajo costo, ofrece funcionalidades nuevas pero más restricciones, por acuerdo al que se llegó con Nokia.
Windows Phone 7.8	Se estrenó en enero de 2013 donde se ofrece nuevas ventajas como la pantalla de usuario y fondos personificados para la imagen de bloqueo. A partir de acá se centró en WP8 para generar nueva versión.

Tabla 1. Tabla Comparativa de Windows Phone Versión 7.0. Disponible en: <http://dspace.epoch.edu.ec/bitstream/123456789/3540/1/18T00554.pdf>

Windows Phone 8:

Es la segunda generación de los sistemas operativos Windows Phone liberada en el 2012, donde sustituye la arquitectura basada en Windows CE con uno basado de Kernel de Windows NT, permitiéndole a los desarrolladores aportar todas las aplicaciones fácilmente entre las 2 plataformas. WP8 soporta pantallas con mejor resolución y procesadores multi-núcleo, NFC, posee control infantil donde se encuentra un área disponible y controlado para los niños, independiente del resto del móvil, también Data Sense el cual permite medir el consumo de recursos del dispositivo móvil, además de características adicionales como la cartera para guardar tus tarjetas de fidelización y crédito, etc. Como se puede visualizar en el siguiente cuadro comparativo: [6]

Versión	Descripción
Windows Phone 8 Update 1 (GDR1) ("Pórtico")	Primera versión donde permite conservar la Wifi activada aunque se bloquee el dispositivo móvil y otras

	ventajas.
Windows Phone 8 Update 2 (GDR2)	Se distribuyó a finales de 2013, contiene radio FM, soporta los protocolos CardDAV y CalDAV, mejoras HTML5 en Internet Explorer 10.
Windows Phone 8 Update 3 (GDR3)	Soporta procesadores quad-core, pantallas Full HD 1080p con una nueva fila de Live Tiles, modo de conducción, nuevas características de accesibilidad, ventajas en la opción "compartir conexión", tonos configurables, bloqueo de rotación de pantalla, cierre de aplicaciones desde la opción multitarea y ventajas en las conexiones Wi-Fi y Bluetooth.

Tabla 2. Comparación Windows Phone Version 8.0. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/3540/1/18T00554.pdf>

2.2.2.4.6. Estándar de desarrollo ISO/IEC 12207

El estándar Information Technology / Software Life Cycle Processes trata los procesos del ciclo de vida del software de la organización ISO. Su estructura comprende las necesidades de cualquiera que lo use, ya que se basa en principios fundamentales como la modularidad la cual pretende conseguir procesos con un acoplamiento muy mínimo y la máxima cohesión posible entre los procesos y por otro lado la responsabilidad la cual busca establecer una persona responsable para cada proceso que se ejecuta, dando facilidad a la aplicación de dicho estándar en el medio de personas que estén involucradas entre sí sin importar el uso final. [36]

- Procesos principales: se compone de cinco procesos los cuales brindan servicio a las partes principales del ciclo de vida del software. Proceso de adquisición, de suministro, de desarrollo, de operación [36]
- Procesos de apoyo: Estos procesos son responsabilidad de la organización donde se lleva el proceso y la cual se asegura de que el proceso si es verdadero y esta funcional. Se divide en procesos de documentación, de gestión de la configuración, de aseguramiento de la calidad, de verificación, de validación, de revisión conjunta, de auditoria y de solución de problemas. [36]
- Procesos de organización: Los emplea la organización para establecer e implementar una estructura constituida por procesos y personal asociados al ciclo de vida para mejorar continuamente. Básicamente se divide en 4 partes las cuales son proceso de gestión, de infraestructura, de mejora de proceso y de recursos humanos [36]

2.3. ANTECEDENTES DE LA EMPRESA Y CONTEXTO

Para empezar a entender el contexto de lo que es la empresa como tal, se deben definir requisitos importantes como los códigos QR y una comparación entre los mismos ya que el trasfondo principal es un lector que permita la lectura de estos importantes códigos.

2.3.1. Códigos QR

Un código QR es una versión mejorada de los códigos de barras que conocemos convencionalmente, en todos aquellos productos que consumimos o usamos en nuestra vida diaria. Estos códigos pueden contener cerca de 350 veces más la cantidad de información que se puede almacenar en un código típico de barras bidimensional. [2]

A pesar de que los códigos de barras se remontan al año 1952, donde se creó la primera patente y posteriormente evoluciono al mercado comercial con uso de lectura, fueron los que dieron inicios a todos estos códigos QR que hoy en día usamos para agilizar procesos y evitar errores humanos en la sofisticada industria en la que vivimos. [2]

Los “QR codes” son interpretables por cualquier aparato que pueda captar imágenes y que además, cuente con el software adecuado para realizar tales instrucciones; Su forma es reconocible por su patrón de cuadros oscuros y claros en tres de las esquinas del símbolo y su nombre es debido a la frase “Quick Response (Respuesta rápida)”, por su diseño para ser decodificado a alta velocidad. [2]

2.3.1.1. Cuadro comparativo

A continuación, se encuentra un cuadro comparativo de las más importantes aplicaciones, a la hora de hablar de códigos QR y procesamiento de imágenes a través de un medio digital móvil, en nuestro caso Windows Phone, además de que nos servirá como referencia para desarrollar un ámbito de la aplicación parecido, por lo que tenemos a partir de [2]:

NOMBRE	DESCRIPCION	PLATAFORMA	URL DE DESCRAGA
Beetag Reader	Permite leer códigos bidimensionales QR, datamatrix y un código bidimensional creado por la propia empresa BeetagCode, adicionalmente dispone de soporte para leer códigos de barras unidireccionales del tipo EAN-13/UPC-A.	Android – iPhone – Bada – BlackBerry – J2me – PalmOS 5.x – Symbian -- Windows Mobile 5, 6, 6.5 – Windows Phone 7.	<ul style="list-style-type: none"> http://www.betagagg.com/en/download-gr-reader/
UpCode Reader	El lector funciona muy bien con cámaras con autofocus e interacciona perfectamente con el terminal y sobre todo interpretando QR con	Up Code ofrece un lector de QR perfecto para terminales Nokia con Symbian	<ul style="list-style-type: none"> http://www.upc.fi/en/upcode/download/

	Vcards. Upcode lee además códigos datamatrix y códigos unidireccionales, características que le confieren un gran versatilidad.		
I.nigma Reader	Tiene soporte para una amplia gama de terminales y plataformas. Este lector permite compartir el código leído fácilmente vía SMS, redes sociales como Facebook o Twitter o visualizando el Qr en pantalla para que pueda ser leído nuevamente desde otro terminal.	Android – iOS – Symbian – Blackberry – Windows phone	<ul style="list-style-type: none"> • http://www.i-nigma.com/DownloadI-nigmaReader.html
QuickMark Reader	Es una aplicación de código de barras escáner móvil que permite búsqueda automática de formatos de código de barras múltiples desde su teléfono. (Quick Code, Código QR, Data Matrix, EAN 8 / 13, Código 39, Código 128) El uso de códigos de barras móviles para acceder a enlaces de la web, añadir contactos, navegar por los mapas, etc.	Apple, ASUS, AT&T, BenQ, CHT, Cingular, CMCC, Dopod, Emobile, ETEN, Fujitsu-Siemens, GIGABYTE, HP, HTC, HUAWEI, iDo, i-mate, Lenovo, LG, Lobster, Mio, Motorola, Nokia, O2, OKWAP, Orange, Panasonic, Pantech, Qtek, Samsung, Siemens, SoftBank, Sony Ericsson, Sprint, Swisscom, T-Mobile, UBiQUIO, UTStarcom y Vodafone.	<ul style="list-style-type: none"> • http://www.quickmark.com.tw/En/basic/downloadMain.asp
Kaywa Reader	Lector de códigos bidimensionales	Symbian, Java como Motorola, Samsung o Sony Erikson. Windows Mobile.	<ul style="list-style-type: none"> • http://reader.kaywa.com/getit

Lynkee Reader	Lynkee 2 le ofrece un montón de interesantes nuevas características y mejoras: navegador opcional y dedicado interno, 3 modos de escaneo, Facebook login, Generador de QR, motores de búsqueda, los llaveros, ledFlash, negativo y más rápido	IPhone – Blackberry - Sony Ericsson - HTC – Motorola - Nokia.	<ul style="list-style-type: none"> • http://m.lynkee.com/
SnapMaze Reader	Ofrece para generar códigos QR que almacenan URL de la página web que actualmente está viendo. Este es un plugin para el navegador que cambia cualquier enlace web en una página web en un código QR.	Compatible con todos los teléfonos que soporten aplicaciones Java J2ME MIDP 2.0, CLDC 1.1. y WXXX. También algunos modelos Motorola y Samsung.	<ul style="list-style-type: none"> • http://www.snapmaze.com/?q=node/7
NeoReader	NeoReader es el lector de códigos bidimensionales más rápido y efectivo del mercado y permite leer todos los tipos de códigos estándar (QR, Datamatrix, Aztec, EAN, UPC y Code 39)	Apple, Blackberry, Motorola, Nokia, Samsung y Sony Ericsson.	<ul style="list-style-type: none"> • http://www.neoreader.com/get-neoreader/wap-download

Tabla 3. Comparacion Múltiples Lectores QR. Disponible en: www.codigos-qr.com/lectores-codigos-qr

Debemos considerar, además de estas aplicaciones ya existentes y famosas entre los diferentes sistemas operativos como Android, IOS o Windows Phone, las diferentes tecnologías de dispositivo móviles que permitieron este avance en los códigos QR, así como las variadas opiniones de autores que han trabajado estos códigos para definirlos o tener un análisis detallado de su funcionamiento. [2]

Estos códigos QR son, por lo general, implantados en un teléfono celular con opciones inteligentes o el llamado Smartphone; la evolución de estos Smartphone permitieron cada vez más integrar aplicaciones para los usuarios con funciones más especializadas día a día de desarrollo, así como juegos, música, chats, redes sociales, e incluso se asemeja a un computador. Nuestro objetivo son los códigos QR, los cuales al escanearlos nos da información de publicidad, arte, ocio, marketing, servicios, sistemas o diferentes temas alojados en ellos. [2]

En la tesis de Juliana Gaviria García, titulada “Nuevas tecnologías, nuevas formas de negocios: Smartphone y códigos QR en Chile”, encontramos opiniones acerca de los códigos QR, así como las ventajas y desventajas proporcionadas por un experto en códigos QR: “Estos códigos han ayudado mucho al momento de buscar información o acceder a esta, creando una accesibilidad al mundo digital. Una de las ventajas de estos códigos, es que en primer lugar su creación es gratuita. No hay ningún costo y se puede generar muy fácil a través de internet. Por otro lado las empresas que usan esta tecnología están muy cerca de los clientes y es sencillo el seguimiento de estos códigos. Además de facilitar el acceso a las páginas webs de las empresas, su manejo es muy sencillo y cualquier persona puede acceder a la lectura de estos códigos. Otra ventaja de los códigos QR es que al ser escaneado se evitan errores de tipeo.”[3]

Sin embargo se mencionan desventajas, como el desconocimiento de esta aplicación es aún bastante grande y no es muy usada, así como el uso del código en diferentes plataformas o dispositivos no es implementada como original con el teléfono inteligente; también se debe pensar en donde se pondrá el código, que tenga buena cobertura de señal móvil, de fácil acceso, y además donde el usuario pueda encontrar lo que está viendo por medio del código QR, como un producto, un servicio o un bien. Cabe la posibilidad de que estos códigos puedan tener errores y alejar a los clientes del fin publicitario por confusiones con la estrategia de ventas. [3]

Las utilidades de estos códigos son de gran importancia al ser escaneados, encontrando información como páginas web, datos de empresas, datos de contactos, productos, compras on-line, descuentos en comparas, etc. Además se pueden encontrar en sitios como vallas publicitarias, paraderos de buses, tarjetas de negocios, revistas y todo esto ha ayudado a que muchas empresas hayan crecido y sean reconocidas por sus clientes e incluso el departamento de marketing puede manejar mejores campañas publicitarias para aumentar las ventas por medio de este código. [3]

Los usos son casi inimaginables, pero nuestra área de estudio se centra en los museos, como esta aplicación a crecido en este contexto para que las personas tengan mayor interactividad en algún tema en específico expuesto en un museo, ya sea por niños, jóvenes o adultos. [3]

2.3.2. Librería para lector de códigos QR

ZXing es una librería procesadora de imágenes multi-formato en 1D/2D y de código abierto. Actualmente es capaz de reconocer los formatos UPC-A, UPC-E, EAN-8, EAN-13, Códigos 39, 93, 128, ITF, Codabar, RSS-14 (en todas sus variantes), Matriz de datos (Data Matrix), Aztec, PDF 417 y por supuesto los populares códigos QR, muy usados desde que la tecnología móvil está presente en nuestras vidas. [38]

Esta librería fue desarrollada e integrada por Sean Owen, Daniel Switkin, Equipo de ZXing en el año 2008, inicialmente para Java donde se pudieran leer estos códigos de manera rápida y fácil aplicándolo a productos cotidianos dentro del mundo real. Con el fin de aumentar la información de un producto que contenga

códigos de este tipo, se logró llevar un avance en publicidad y exactitud de los procesos del mismo. [39]

2.3.3. Contexto de la aplicación

El desarrollo de una aplicación que sea capaz de leer códigos QR, hecha para Microsoft Windows phone y codificada en XAML de Visual Studio Express, permitirá adaptarse a los entornos de un museo, más exactamente en la mina de sal de Nemocón, donde se podrá encontrar códigos para uso informativo con el fin de volver interactivo todas las atracciones turísticas encontradas allí. Para esto, se debe tener en cuenta cosas como las que expone María Soledad Gómez Vílchez en su artículo “QR Code en museos”: “Las posibilidades y las aplicaciones que se le puede dar a los códigos QR en los museos son variadas. La facilidad de su uso, los avances tecnológicos y la difusión de la telefonía móvil y el bajo coste que supone su utilización, hace de este sistema una herramienta de utilidad para aportar información e interactuar con los usuarios de un museo. Su reducido tamaño facilita que pueda integrarse en la museografía o en las publicaciones del museo sin distorsionar la imagen general y aportando un elemento de valor. Sus aplicaciones son variadas, tanto de cara al visitante como para la organización interna del trabajo.”[4]

Se puede distribuir estos códigos en áreas de exposición, haciendo interactuante al usuario con el museo por medio de cosas útiles como mapas de orientación, realidad aumentada, itinerarios, información complementaria, juegos de pistas en el museo o geo localización de piezas donde fueron halladas. También se está dando una magnífica acogida de estos códigos para llegar a archivos de audio explicativos en MP3. En áreas de reserva podrían ser útiles en inventarios como por ejemplo el almacenaje de piezas ya que almacena mayor información de cada pieza u objeto a guardar. El famoso QR code difusión que se usa en cartelería, trípticos, folletos publicitarios, publicaciones del museo, anuncios de prensa o televisión, etc., para ampliar información que estos soportan. En la biblioteca del museo es otra alternativa para usar los códigos QR con fines de localización topográfica de libros, referencia bibliográfica de la obra, rapidez en búsqueda de los libros y enlaces a páginas web con información sobre la obra. Por ultimo podemos usar los QR code en eventos como conferencias, seminarios y todo tipo de eventos que se realicen en el museo [4].

2.4. MARCO LEGAL

Los códigos QR, por su naturaleza no poseen gran limitación de leyes en los entornos donde se hace su mayor uso; por lo general se encuentran proyectos de ley para impulsar su uso publicitario y educativo en diferentes ciudades como Buenos aires. El fin de estos códigos con su uso en diferentes partes del mundo, como se ha dado cuenta los diferentes gobiernos es referenciar y brindar y/o ampliar información audiovisual sobre actividades culturales; patrimonio histórico; sitios de interés; monumentos y esculturas en la vía públicas; plazas, parques, jardines y espacios públicos; y lugares turísticos. [33]

Un claro ejemplo lo podemos encontrar en puerto rico, donde existe un proyecto de ley impulsado el 24 de marzo de 2014 por la cámara de representantes, donde

proponen “crear y establecer la “Ley de Respuesta Inmediata Escaneando un Código (Código QR) de 2014” con los fines de ampliar la percepción pública sobre la aplicación de un Código QR o “barcode” para la difusión de información de los atractivos turísticos, patrimonios y de alto interés cultural, educativo e histórico que ofrece la Isla de Puerto Rico a través del uso de teléfonos inteligentes, fijar la Política Pública, fijar responsabilidades y obligaciones y para otros fines relacionados.”[5] Todo esto para desarrollar un sistema de acceso sistemático y estratégico que contenga recursos informativos en una gran variedad de idiomas para todos aquellos turistas que visitan este país. [31]

Así como se nombra anteriormente, en México también existen proyectos de ley para incluir estos útiles códigos de respuesta rápida a las placas metálicas de los automóviles, así como en las calcomanías y tarjetas de circulación de los automóviles, hasta incluso en los pases de las personas con el fin de ser directamente identificados y buscados rápidamente para ratificar la legalidad de todos los papeles. [32]

En conclusión, por el momento solo se pueden identificar a aquellos proyectos de ley que buscan normatividad para implementar su uso y que traiga con ello una gran variedad de beneficios económicos para las personas de un país económicamente constituido, pero habrá que esperar cuáles serán las posibles inhabilidades.

3. METODOLOGIA

Para abordar el tema del desarrollo de la aplicación en arquitectura Windows Phone, debemos realizar un estudio detallado de lo principal dentro del entorno de programación para así seguir un camino estructurado que permita cumplir con los objetivos principales del proceso de la aplicación.

Como se observa, el problema principal es la ausencia de una aplicación en un museo con lector de códigos QR para el almacenamiento de información propia del contenido histórico del lugar y sus componentes a través de dispositivos móviles; esto es un avance tecnológico de gran importancia para el medio en el que vivimos y como hoy en día interactuamos con objetos encontrados allí para obtener cada vez más información.

El desarrollo de la aplicación se realizara por lenguaje nativo de XAML, plataforma única para programar en dispositivos de Windows Phone y la cual es una poderosa herramienta para crear desde una simple aplicación, hasta un desarrollo del sistema operativo completo de esta plataforma en dispositivos móviles.

3.1. TIPO DE INVESTIGACIÓN

Esta investigación se desarrolla dentro de un marco teórico y aplicado, donde se requieren resultados específicos, en este caso una aplicación para ser usada dentro del ámbito de un museo, por medio de la plataforma Windows Phone.

La investigación teórica y aplicada nos ayudara a analizar las diferentes variables de este, de tal manera que al estudiar las metodologías para desarrollar y los diferentes enfoques que la ingeniería nos da para crear nuevas aplicaciones en Windows Phone, se encontrara una manera justificable y aplicable para producir la aplicación; además se debe pensar en tener simplicidad en el código a generar pero teniendo características definidas como el lector de códigos QR, procesamiento de imágenes, funcionalidades de cámara, etc.

Esta investigación, se fundamenta en los resultados expuestos anteriormente, en el marco teórico, donde se define una multitud de términos para de ahí mismo sacar los posibles instrumentos o las bases que pueden ser usadas como elementos de ejecución del proyecto, ya sea en fases o periodos de tiempo de acuerdo a una metodología.

3.2. DISEÑO DE INVESTIGACIÓN

En esta investigación se intentara abordar los múltiples conocimientos de manera teórico practica en los que se basa la ingeniería de software para el desarrollo de una aplicación móvil con el cual, desde un punto de vista tecnológico, permitirá realizar un objetivo final, "Construir un lector de códigos QR".

3.2.1. Proceso de investigación

Por medio del proceso de investigación, se obtendrá un desarrollo simple y efectivo que ayudara a tener rapidez y unas bases sólidas en el crecimiento de la aplicación a construir; por ende el proceso se convierte en un medio simple de efectividad al localizar la información necesaria para el proyecto y transformarla en un determinado tiempo, con el fin de lograr un objetivo final.

A continuación, se muestra la forma del proceso investigativo que se va a seguir con el fin de involucrarse en el proyecto establecido y obtener un resultado exitoso:

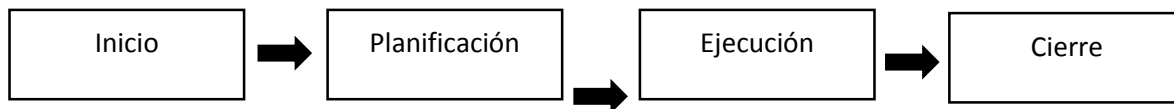


Figura 7. Proceso de Investigación: Fases.

- **Fase 1: Inicio:** En esta fase se investigó toda la parte relacionado con historia, funcionamiento de los códigos QR y posibles áreas del conocimiento para uso y aplicación de dicho marco teórico dentro del desarrollo de la aplicación.
- **Fase 2: Planificación:** Se planifico y decidió las mejores áreas, los estándares, el paradigma con el que se basó y las técnicas y herramientas del desarrollo de lo que se usó dentro del aplicativo final; en las siguientes secciones se explica con más detalle las teóricas específicas a usar además de aspectos básicos como los recursos a usar y lo relacionado con planteamiento del problema.
- **Fase 3: Ejecución:** En la ejecución se usara la metodología XP donde tendrá una serie de iteraciones para el posterior desarrollo de la aplicación y un producto final
- **Etapa 4: Cierre:** Se da cierre al proyecto concluyendo los beneficios que se logró sacar de la aplicación de toda le teoría expuesta en el desarrollo de la aplicación.

3.2.2. Áreas, Paradigmas, Técnicas y Herramientas

3.2.2.1. Áreas del Conocimiento de la Ingeniería de Software

Para aplicar lo investigado en las secciones anteriores, se contara con el tiempo de requerimientos del software, que se denominaría a las solicitudes que realiza el usuario en el momento de saber qué es lo que desea para la posterior creación. Por tal motivo el objetivo principal será entregar las funciones requeridas que desee el cliente satisfaciendo las necesidades y una o unas funcionalidades preestablecidas.

Acá, podemos decir que se determina un papel importante dentro de la ingeniería de software y sus áreas de conocimiento, ya que se empieza el diseño, la construcción y las pruebas del software a realizar, por supuesto siguiendo lineamientos como la planificación, la estructura, el análisis, las notaciones y el seguimiento de pruebas.

Seguido de esto, se da mantenimiento a dicho proyecto o aplicación ya construida donde se evalúan diferentes variables de mejora y es aquí donde entra en relación la calidad de dicho software para que así mismo ayude a una fácil construcción con buenos resultados.

Esto sin la administración de la ingeniería de software no sería posible, porque se usó un alcance, una planificación, la comunicación, la revisión y un determinado cierre dentro del marco teórico de lo que se aplica a este desarrollo.

Hay que resaltar que los instrumentos de la ingeniería usados acá son por medio de métodos como iteraciones que cada que se realiza una se obtiene más evolución en el proyecto y todo esto por medio de las herramientas que se desarrollara la aplicación como Visual Studio.

3.2.2.2. Paradigma de Desarrollo de Software

Para esta investigación y su posterior aplicación se debe usar el enfoque orientado a objetos, ya que permite una facilidad de realización del código y reusabilidad del mismo para obtener un proceso exitoso y de manera ordenada con respecto a todos aquellos métodos que se usan dentro de la misma aplicación.

Ya que la aplicación a desarrollar se realizara de forma iterativa, se interrelaciona con los fundamentos básicos del enfoque orientado a objetos, donde el inicio es la comunicación con el usuario para obtener el dominio del problema e identificar las clases básicas que se van a desarrollar para el lector de códigos QR.

Por ende es de vital importancia tener una planificación y análisis del proyecto en cuanto a sus riesgos y estructura básica para consolidar un buen producto final. Para llevar a cabo dichas teorías se tendrá en cuenta:

- Identificación de clases candidatas como parte de la estructura que conformara el lector donde se encuentra una clase principal en la cual esta instanciada el código fuente de lectura de QR's.
- Se usó la biblioteca externa Zxing la cual permite procesar imágenes en multiformato 1D/2D que se encuentra bajo licencia gratuita para su uso.
- El desarrollo de clases se realizara de acuerdo al número de menús que se deseen integra en la aplicación.
- Se añadirán todas las clases construidas en cada iteración que permita generar nuevo código.

3.2.2.3. Metodología de Desarrollo de Software

Para un diseño óptimo de la aplicación y de manera rápida y eficiente, se perseguirá los procedimientos que se impone mediante la metodología para la construcción de un software de manera ágil; de este modo se tomara como base la metodología de programación extrema o XP por sus siglas en inglés (eXtreme Programming).

La programación extrema permitirá desarrollar el proyecto poniendo más énfasis en la adaptabilidad que en la documentación y previsibilidad, por ello se deben tener en cuenta los valores que se manejan en esta creación de aplicación.

Dentro del XP se manejan valores como la simplicidad que es importante en el momento de creación del código y la comunicación entre el cliente y el equipo de desarrollo donde juega un papel importante de intercambiar ideas y código para mayor facilidad de aprendizaje; Por supuesto a todo esto, se debe aclarar que si hay comunicación debe haber una retroalimentación, donde se permite conocer en tiempo real las opiniones del usuario final para que en cada entregable de la aplicación, pueda ser mejorado y llevado a un estado óptimo.

Para la aplicación en Windows Phone con desarrollo de lector de códigos QR y procesador de imágenes para información del museo, tendremos que tener en cuenta características fundamentales de esta metodología que permitirá seguir:

- Desarrollo iterativo e incremental: Se iniciara con el desarrollo fundamental de la estructura de la aplicación, para que ella pueda tomar, a través de su

cámara integrada, los códigos QR y procesar información relevante a ese código generado de manera preestablecida con el objeto del museo. Durante cada desarrollo se ira unificando la interfaz principal, el lector de códigos y usos de cámaras y enlaces de contenido con integración a internet por medio del desarrollo iterativo o avanzando por pasos.

- Pruebas unitarias y continuas: La aplicación para el museo será probada en cada módulo o iteración para buscar posibles errores de código y desarrollo, así se podrá mejorar el aplicativo de manera frecuente y automatizada en cada proceso. También se aplicaran pruebas de integración asegurando el correcto funcionamiento de subsistemas así como pruebas de regresión observando detalladamente en posibles errores de líneas de código o bugs, etc.
- Integración del equipo desarrollador con el cliente: se mantendrá informado y conectado con el cliente para que este opine y corrija de manera inmediata el desarrollo de la aplicación y sus funcionalidades, dando así una posible corrección de errores para añadir nuevas funcionalidades en la aplicación final.
- Refactorización del código: El código se debe desarrollar de manera legible y entendible para un recomendable y posterior mantenimiento en el momento de la ejecución de la aplicación. Se reusara todo aquel código para mayor facilidad y mejor entendimiento humano.
- Simplicidad de código: una objetividad en el proyecto es la simplicidad del código generado para la aplicación a desarrollar. Entre más simple sea el código ayudara a que en un futuro se pueda agregar funcionalidades mejores, pero con trabajo moderado y de manera rápida y así brindar mantenimiento fácil de realizar.

Para el desarrollo de roles, se debe tener en cuenta que para este proyecto será desarrollado por 1 solo integrante, por lo que se hace referencia a:

- Programador: Producir el código del sistema.
- Cliente: Expresa la idea para así mismo la implementación de lo que pidió en estancia final.
- Tester: Ejecutara pruebas y difunde los resultados en el equipo, además de ser responsable de herramientas de soporte.
- Consultor: ayuda al equipo a resolver problemas del proyecto. Normalmente es externo al equipo (Tutor).

3.2.2.4. Arquitectura de Software

Por medio de la siguiente ilustración se puede observar cómo será organizada y detallada la aplicación para su posterior funcionamiento, un lector de códigos QR:

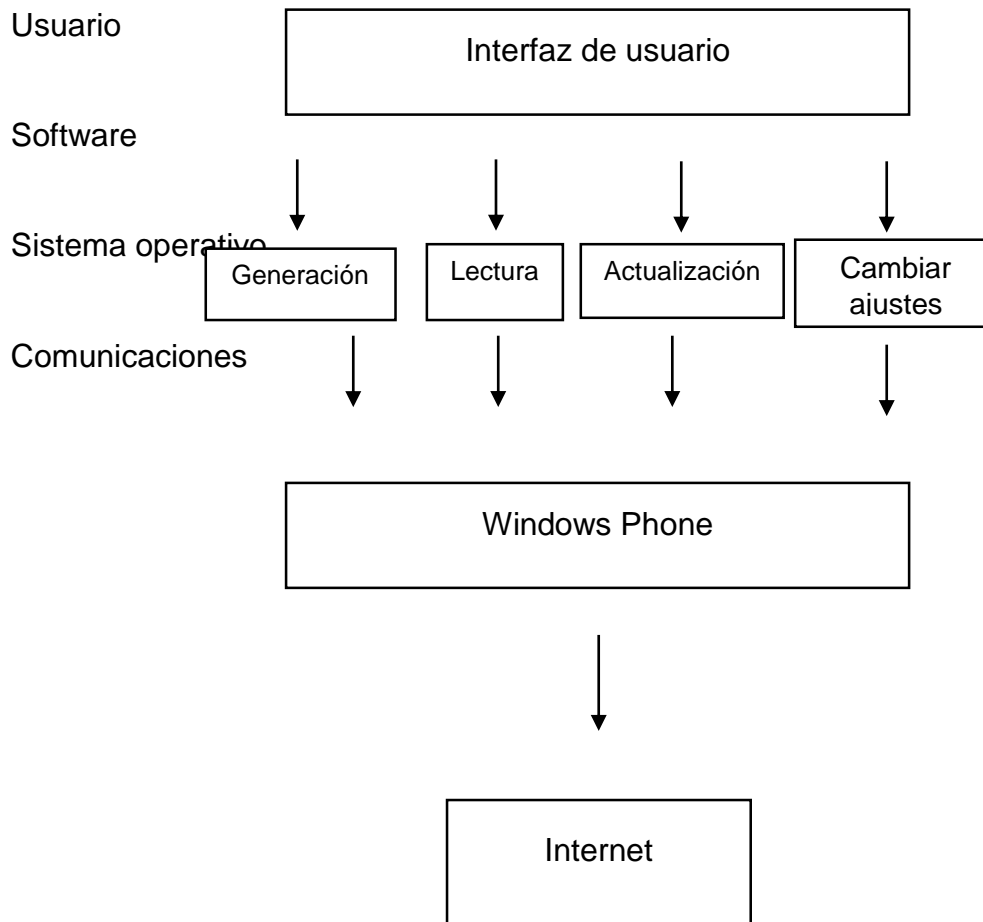


Figura 8. Arquitectura de Software

- **Interfaz de usuario:** Su funcionalidad básicamente es otorgar al usuario una comunicación con el sistema para que acceda a todos los módulos hasta en la parte más baja.
- **Generación:** Si se tienen datos desde la capa del usuario, se puede incluir un mensaje en un código QR para compartir con otras aplicaciones del dispositivo
- **Lectura:** Se encarga de extraer un mensaje de un código QR
- **Actualización:** Se informan las nuevas cosas que posee la versión de la aplicación
- **Cambiar ajustes:** Se incluyen ajustes por defecto de la aplicación
- **Sistema operativo Windows Phone:** Todo lo anterior se debe comunicar con el sistema operativo.
- **Comunicación externa:** El lector debe tener la posibilidad de comunicación con otras aplicaciones

3.2.2.5. Estándares de Desarrollo

Existen muchos estándares a seguir para el desarrollo de software o de una aplicación móvil en Windows Phone, por ende decir que se usaran todas aquellas

es prácticamente increíble aunque se busca un punto de equilibrio o perfección con respecto a la calidad del desarrollo del aplicativo.

Pero en este punto, no podemos preguntar ¿por qué estándares de desarrollo?, Pues es de vital importancia tener en cuenta que el desarrollo de software debe hacerse o guiarse bajo unos parámetros que de una forma u otra conllevan a calidad en el mismo y a su vez se ejecuta de forma exacta las tareas con respecto a los lineamientos del equipo.

Por dichas razones, se ha tomado el estándar “ISO/IEC 12207 Information Technology / Software Life Cycle Processes” [36] el cual hace referencia a los procesos del ciclo de vida del software de la organización, donde se establecen actividades que se aplican a la definición de requisitos, su adquisición, la configuración del sistema y la finalización de su uso. Además su objetivo principal es proporcionar una estructura común para que todo el equipo involucrado, desde el desarrollador hasta el cliente, usen un lenguaje común o procesos bien definidos dentro del desarrollo del lector de códigos QR.

3.2.2.6. Herramientas de Desarrollo

Para esta investigación se hace uso del visual Studio, como ya se había definido anteriormente, este entorno de desarrollo permitirá integrar una solución óptima y confiable para el sistema operativa al que va dirigido dicho proyecto, por supuesto en un lenguaje de programación XAML.

Nosotros como desarrolladores, tenemos la gran ventaja de crear múltiples aplicaciones en cualquier entorno por medio de esta herramienta que al final del proyecto ayudara a construir una aplicación sencilla y fácil que leerá códigos QR para recuperar información de manera inmediata.

Todo esto orientado a la creación de aplicaciones móviles por medio del visual Studio para Windows Phone el cual ayuda a integrar y probar por medio de un emulador la construcción del aplicativo a obtener como meta final.

La versión de Visual, el nombre de los complementos y el número de versión del entorno de desarrollo software para Windows Phone (llamado Windows SDK, "Software Development Kit") serán los siguientes:

- Version de Visual Studio: Express 2012 for Windows phone.
- Complementos: Windows Development Tools (WDT).
- Windows SDK: 8.0
- Framework: 3.0 o superior
- Librería ZXing para leer códigos QR

3.2.3. Cronograma

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Agregar nueva columna
1		Fase Inicio	60 días	lun 01/02/16	vie 22/04/16		
2		Investigación Marco Teórico: Antecedentes de la Investigación	7 días	lun 01/02/16	mar 09/02/16		
3		Fundamentos del problema de investigación	7 días	mié 10/02/16	jue 18/02/16	2	
4		Fundamentos ingeniería de software: áreas del conocimiento	7 días	vie 19/02/16	lun 29/02/16	3	
5		Fundamentos ingeniería de software: Paradigmas	7 días	mar 01/03/16	mié 09/03/16	4	
6		Fundamentos ingeniería de software: Metodologías de desarrollo	7 días	jue 10/03/16	vie 18/03/16	5	
7		Fundamentos ingeniería de software: Técnicas y herramientas	7 días	lun 21/03/16	mar 29/03/16	6	
8		Antecedentes: Códigos QR	7 días	mié 30/03/16	jue 07/04/16	7	
9		Antecedentes: Contexto de la aplicación	6 días	vie 08/04/16	vie 15/04/16	8	
10		Marco legal	5 días	lun 18/04/16	vie 22/04/16	9	
11		Fase Planificación	61 días	sáb 23/04/16	vie 15/07/16	1	
12		Selección de las áreas del conocimiento del software	12 días	lun 25/04/16	mar 10/05/16		
14		Metodología de desarrollo a usar	8 días	lun 23/05/16	mié 01/06/16	13	
15		Desarrollo de arquitectura de software	8 días	jue 02/06/16	lun 13/06/16	14	
16		Estandar de desarrollo a usar	8 días	mar 14/06/16	jue 23/06/16	15	
17		Herramientas seleccionadas	8 días	vie 24/06/16	mar 05/07/16	16	
18		Instrumentos de investigación	8 días	mié 06/07/16	vie 15/07/16	17	
19		Fase Ejecución	63 días	lun 18/07/16	mié 12/10/16	11	
20		Fase Inicio: Reunión de inicio	1 día	lun 18/07/16	lun 18/07/16		
21		Fase Inicio: Estimación de costos	2 días	mar 19/07/16	mié 20/07/16	20	
22		Fase Inicio: Asignación de recursos	2 días	jue 21/07/16	vie 22/07/16	21	
23		Fase Planificación: Análisis alcance del proyecto	10 días	lun 25/07/16	vie 05/08/16	22	
24		Fase Planificación: Realización de estimaciones	10 días	lun 08/08/16	vie 19/08/16	23	
25		Fase Ejecución: Historias de Usuario	5 días	lun 22/08/16	vie 26/08/16	24	
26		Fase Ejecución: Planificación de recursos	3 días	lun 29/08/16	mié 31/08/16	25	
27		Fase Ejecución - Iteración 1: Desarrollo Requerimientos	7 días	jue 01/09/16	vie 09/09/16	26	
28		Fase Ejecución - Iteración 1: Desarrollo de diagrama de flujo	5 días	lun 12/09/16	vie 16/09/16	27	
29		Fase Ejecución - Iteración 2: Codificar en C# para Windows Phone	10 días	lun 19/09/16	vie 30/09/16	28	
30		Fase Ejecución - Iteración 2: Pruebas funcionales	4 días	lun 03/10/16	jue 06/10/16	29	
31		Fase Ejecución: Reunión de cierre del proyecto	1 día	vie 07/10/16	vie 07/10/16	30	
32		Fase de cierre: Aceptación del proyecto	1 día	lun 10/10/16	lun 10/10/16	31	
33		Fase Cierre: Entrega de aplicación y documentos finales	2 días	mar 11/10/16	mié 12/10/16	32	
34		Fase Cierre	5 días	lun 17/10/16	vie 21/10/16		
35		Reunión inicial de entrega	1 día	lun 17/10/16	lun 17/10/16		
36		Revisión de historias y	3 días	mar 18/10/16	jue 20/10/16	35	
37		Reunión final: entrega total	1 día	vie 21/10/16	vie 21/10/16	36	

Figura 9. Cronograma.

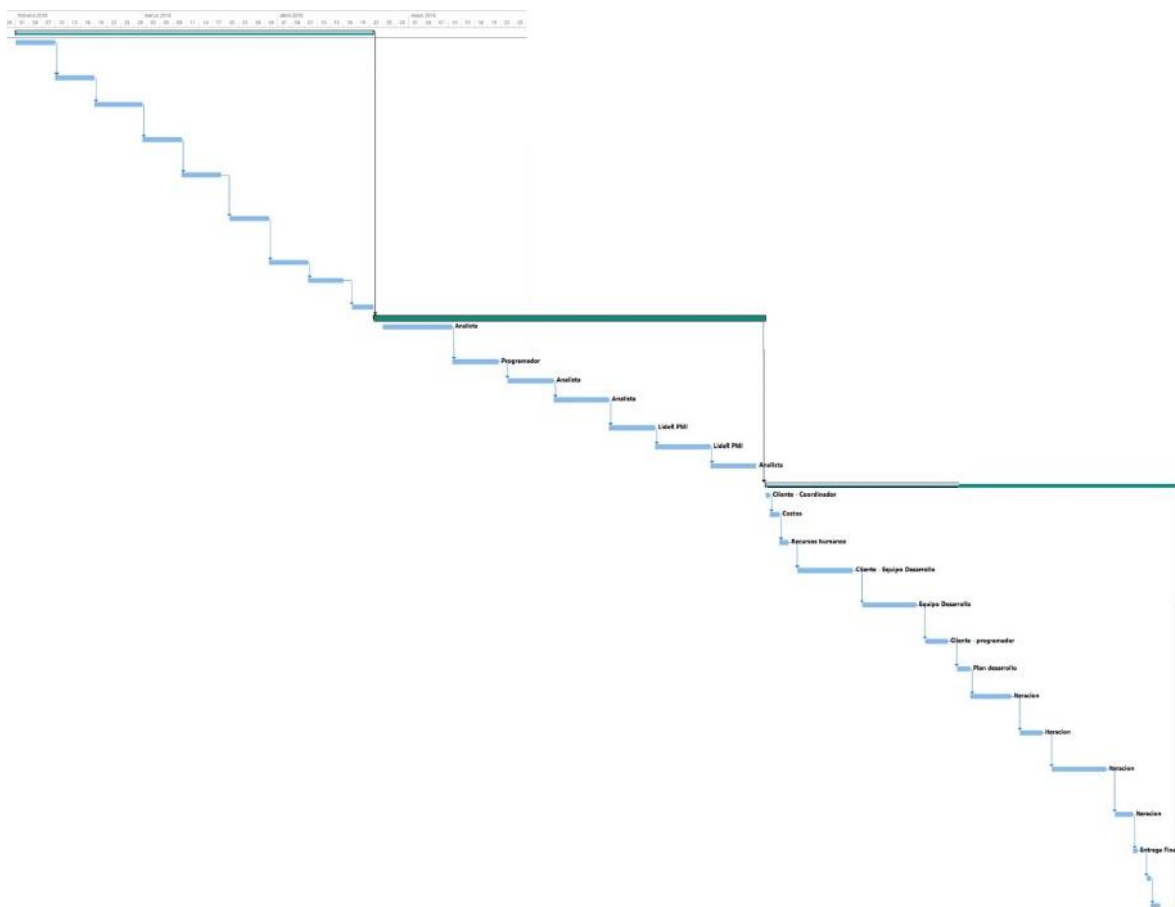


Figura 10. Diagrama de Gantt.

3.2.4. Instrumentos de Investigación

Las técnicas de recolección de datos se basan en realizar la estructuración de encuestas para conocer la opinión de los usuarios que han logrado experimentar y probar la aplicación lectora de códigos QR (post desarrollo) y así mismo validar su funcionamiento, para tener en cuenta el aporte personal y su beneficio a la información recaudada para la mejora de usabilidad, aspecto, dinamismo y originalidad.

Primero, a los usuarios se les mostrara de manera detallada y funcional la aplicación desarrollada y el mantenimiento de la misma para que se hagan una idea de porque la estructura de la encuesta y así mismo evaluar los diferentes aspectos de opinión.

Encuesta	
Fecha de Aplicación	01/03/2017
Tipo de encuesta	Online
Tipo de preguntas	Cerradas – De intención
Nivel de medición	Ordinal
Objetivos	Opinión publica
Datos más relevantes que se desean obtener	Sondeo para posibles mejoras
Datos Complementarios	
Quien realiza la entrevista	David Gómez
Datos de población encuesta	Mayores de 18 años
Relación con el proyecto	Creación de lector de códigos QR
URL de la encuesta	https://docs.google.com/forms/d/e/1FAIpQLSfWi0DkNbVkiZQBBWhvVB_D2Hd83RTgilYeQIQXjfyVSnc3XQ/viewform?c=0&w=1
<p>Estimado(a) señor(a). Mi nombre es José David Gómez Cruz, se está realizando un estudio para posterior a eso desarrollar una aplicación lectora de códigos QR ajustándose a necesidades de los usuarios en un museo, donde se genera un código QR extendiendo información de un elemento allí expuesto y la aplicación permite visualizar todo el contenido por medio de la web. Esto se está realizando en el de Nemocón Colombia, más exactamente en la mina de sal de este municipio y toda la información que usted nos brinda será de carácter estrictamente confidencial.</p>	
Preguntas de filtro	
A1	¿Cuál es su nombre?
A2	¿Cuál es su edad?
A3	¿Cuál es su nivel de escolaridad?
Preguntas de uso de códigos QR	
B1	<p>¿Cuántas veces ha visitado usted un lector de códigos QR en los últimos 30 días?</p> <p>A. 2-3 B. 4-5 C. 6 ó más veces. D. Es la primera vez que visito la Aplicación</p>
B2	<p>¿Cree que volverá a usar</p> <p>A. Sí</p>

	la aplicación en los próximos 30 días?	B. No C. No estoy seguro
B3	¿Por qué motivo/s ha visitado usted la aplicación?	A. Para obtener más información B. Por curiosidad C. A modo de aprendizaje D. Otro (por favor, especifique) _____
B4	¿Cuál es su nivel de satisfacción general con el lector de códigos QR?	A. Completamente satisfecho B. Satisfecho C. Insatisfecho D. Completamente insatisfecho
B5	¿Cómo le gustaría combinar la tecnología con un museo?	
B6	¿Cree que la tecnología aumenta la visita en los museos?	

Tabla 4. Encuesta.

Cabe resaltar que esta encuesta se realiza de manera subjetiva, cuyos resultados dan un valor agregado al trabajo que se realiza más no un aporte necesario a la aplicación.

3.2.5. Presupuesto

Componente	Mes 1	Mes 2	Mes 3	Cantidad	Costo Unitario	Costo total
		MANO DE OBRA				
Analista de Requerimientos y diseño	10H	10H	10H	30H	30000	900000
Analista desarrollador	5H	5H	5H	15H	50000	750000
Analista de calidad y Tester	2H	4H	6H	12H	30000	360000
		HARDWARE				
Uso de computador para desarrollo	30H	60H	100H	190H	4000	760000
Uso computador de pruebas	10H	10H	20H	40H	4000	160000
		SOFTWARE				
Visual Studio Express for Windows Phone	0	0	0	1	0	0
		SERVICIOS				
Energía	10KW/H	10KW/H	10KW/H	30KW/H	1000	30000
Internet	1 PLAN	1 PLAN	1 PLAN	3	50000	150000
Subtotal						3110000
Otros gastos						200000
Total						3310000

Tabla 5. Presupuesto.

3.2.6. Aplicación de la metodología de Investigación

Para esta aplicación se introdujo el desarrollo bajo las fases de investigación donde se siguió de manera exacta el cronograma creado en la sección (3.2.3) y los respectivos pasos y parámetros estudiados anteriormente:

3.2.6.1. Fase de Investigación: Inicio

Se realizó la investigación pertinente a todo lo relacionado con temas de ingeniería de software y códigos QR además de un contexto de donde se ejecutó la aplicación con el fin de aplicar toda esa teoría a un modelo de desarrollo para la construcción del lector de códigos QR. Para esto se demoró aproximadamente 60 días, ya que la teoría requerida para hacer algo es bastante extensa.

3.2.6.2. Fase de Investigación: Planificación

Se determinó las bases de la ingeniería de software y sus áreas de conocimiento, ya que se empieza el diseño, la construcción y las pruebas del software que se

hicieron, por supuesto siguiendo lineamientos como la planificación, la estructura, el análisis, las notaciones y el seguimiento de pruebas. Además de eso el paradigma que se usó es la programación orientada a objetos, las herramientas fue Visual Studio para Windows Phone y el SDK 8.0 de Windows Phone y la metodología XP (eXtremme Programming) con sus diferentes fases que este incluye en el momento de realizar una aplicación.

La duración de esta fase fue de 61 días, ya que se evaluó y determinó de manera exhaustiva tanto las herramientas a usar como la forma de realizar el desarrollo con lo anteriormente mencionado.

3.2.6.3. Fase de investigación: Ejecución metodología XP

3.2.6.3.1. Fase XP: Exploración

Esta fase no demora más de una semana y es donde se intentó resolver, al tiempo con el cliente, cual es el objetivo principal del proyecto a realizar, que en este caso es el desarrollo de una aplicación móvil en tecnología Windows Phone la cual permitirá leer códigos QR, además de analizar los escenarios que definen la aplicación como lo es en un museo con un lector de códigos QR; Así mismo se estima costos del proyecto (véase sección 3.2.5.) para una valoración por parte del cliente de los recursos a utilizar y además, casi al mismo tiempo, se asigna recursos para programar y recoger toda la información de la aplicación.

Como primera medida se hizo con el cliente una reunión donde, en la medida de lo posible, tener una breve explicación del requerimiento de lo que se busca hacer para proyectar un tiempo estimado de finalización de la actividad.

Historias de usuario:

- **H1:** Obtener la arquitectura necesaria para el desarrollo de la aplicación lectora de códigos QR.
- **H2:** Se desea realizar una aplicación lectora de códigos QR que permita obtener información de un museo como por ejemplo videos, textos, imágenes, etc., ampliando dicha información para tener un mayor enriquecimiento de los elementos expuestos.

3.2.6.3.2. Fase XP: Planificación

- **Análisis del alcance del proyecto:** Se hizo un análisis del alcance del proyecto donde se definió que se realizara el desarrollo única y exclusivamente en el ámbito de tecnología móvil, para la plataforma de Windows Phone para construir una aplicación lectora de códigos QR que permita traer información acerca de un elemento expuesto en un museo. Esto tendrá como entrega final un producto funcional el cual ayuda a mejorar la experiencia visual del usuario dentro del recinto.

- **Realización de estimaciones:** Para este proyecto se tiene estimado usar 3 personas para el posterior desarrollo como lo es el Analista de requerimientos, el programador y el Tester los cuales desempeñaran sus funciones respectivas para aportar al proyecto de manera rápida y precisa. Los costos son explicados en la tabla 4 de la sección 3.2.5. Presupuesto.

En la planificación del desarrollo para la aplicación lectora de códigos QR se estudia la anterior historia de usuario para darle tiempo de ejecución y recursos necesarios dentro del mismo proceso con el fin de llegar a cumplir las expectativas y necesidades del cliente:

Plan de publicaciones (Release planning):

Numero	Historia de usuario	Fecha de publicación	Tiempo de desarrollo (estimado)	Número de personas	Pruebas
H1	Realizar la arquitectura en la que va a funcionar el lector de códigos QR	20/08/2016	1 mes	1	No aplica
H2	Desarrollar la aplicación funcional para que lea e interprete códigos QR	30/09/2016	1 mes	1	Funcionales
H3					

Tabla 6. Plan de Historias De Usuario.

3.2.6.3.3. Fase XP: Iteraciones

Para las iteraciones se tendrá en cuenta nada más que dos (2), ya que el desarrollo de dicha aplicación es bastante sencillo y las pruebas se pueden generar de manera fácil para terminar el objetivo final de construir una aplicación para Windows Phone.

Iteración 1:

Descripción	Historias/Tareas	Resultado	Fecha de entrega
Historias realizadas	H1	Finalizada	30/09/2016
Tareas de la historia H1	Diseñar requerimientos	Realizadas	30/09/2016
Tareas de la historia H1	Diseñar diagramas que muestren el funcionamiento de la aplicación	Realizadas	30/09/2016

Tabla 7. Iteración 1.

Requerimientos

Funcionales:

- **RF1.** El usuario Ingresa en la aplicación.
- **RF2.** El usuario Escanea el código.
- **RF3.** El usuario consulta la información obtenida del código.

No funcionales

- **RNF1** La aplicación se desarrollara para arquitectura Microsoft Windows phone en el lenguaje de programación XAML.
- **RNF2** La interfaz de usuario será integrada o unida con la cámara del dispositivo móvil.
- **RNF3** La aplicación necesitara uso de internet con datos móviles o WI-FI.

Diagrama de Interacción

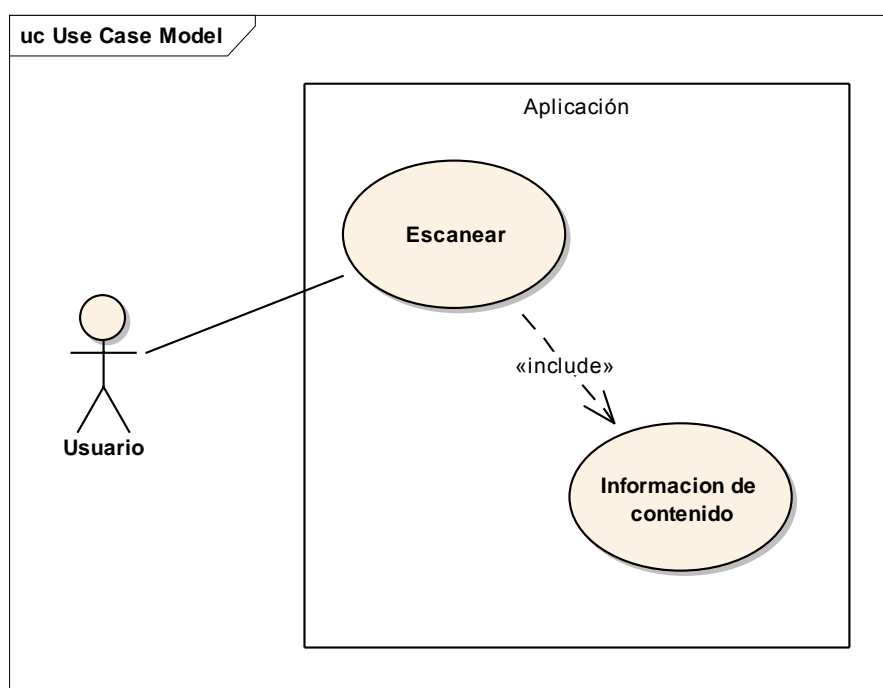


Figura 11. Diagrama UML.

Documentación:

a. 001 – Escanear

Nombre	Escanear. Cód.: 001	
Fecha y Versión	12/05/2015 – V_001	
Requerimientos	RF1. RF2. RNF1. RNF2	
Actor(es)	Usuario	
Descripción	Esta opción permite Escanear un código o imagen QR establecida en el museo para consultar mayor información.	
Precondiciones	Instalar aplicación.	
Flujo Normal	Usuario 1 Ingresa a la aplicación. 3 Escoge la opción de	Aplicación 2 Permite ingreso del usuario y muestra las opciones a realizar

	escanear un código nuevo	4 Valida opción. 5 Escanea código en el medio o museo.
Flujo Alternativo	Usuario 1 3 6 Vuelve a la elección de opciones	Aplicación 2 4 Valida opción. 5 Error en la cámara.
Post Condiciones	Mira información del código o consultar nuevo código otra vez.	

Tabla 8. Documentación Escanear

b. 002 – Obtener información

Nombre	Obtener información. Cód.: 002	
Fecha y Versión	12/05/2015 – V_001	
Requerimientos	RF1. RF2. RF3. RNF3	
Actor(es)	Usuario	
Descripción	Esta opción permite obtener información de un código o imagen QR establecida en el museo.	
Precondiciones	Escanear Código.	
Flujo Normal	Usuario 1 Ingresa a la aplicación. 3 Escoge la opción de escanear un código nuevo	Aplicación 2 Permite ingreso del usuario y muestra las opciones a realizar 4 Valida opción. 5 Escanea código en el medio o museo. 4. Recupera y muestra información del código
Flujo Alternativo	Usuario 1 3 6 Vuelve a la elección de opciones	Aplicación 2 4 Valida opción. 5 Error en la cámara. 7. Error en la recuperación de información.
Post Condiciones	Mira información del código o consultar nuevo código otra vez.	

Tabla 9. Documentación Obtener Información.

Diagrama de actividad por carril de Escanear Código QR

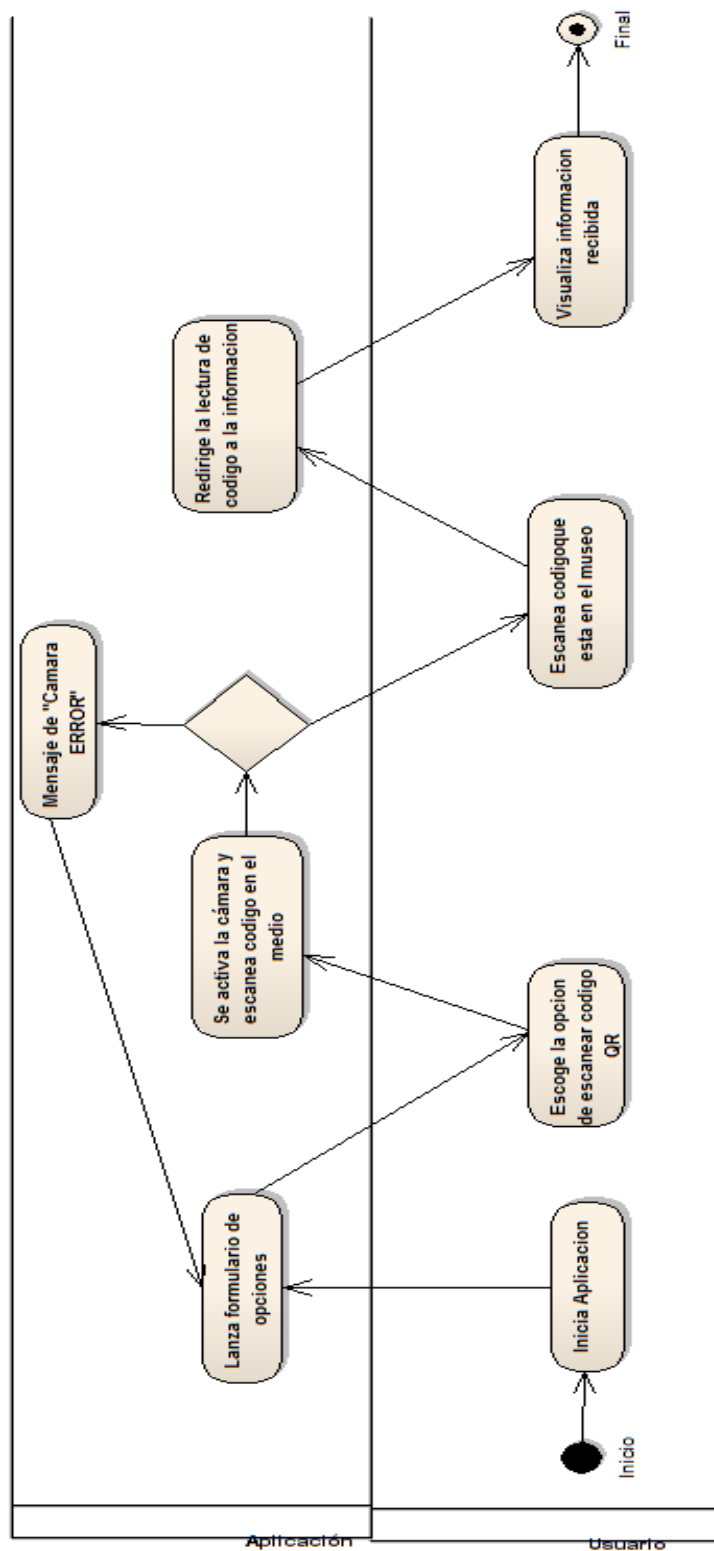


Figura 12. Diagrama Actividad por Carril.

Diagrama de estado de Escanear Código

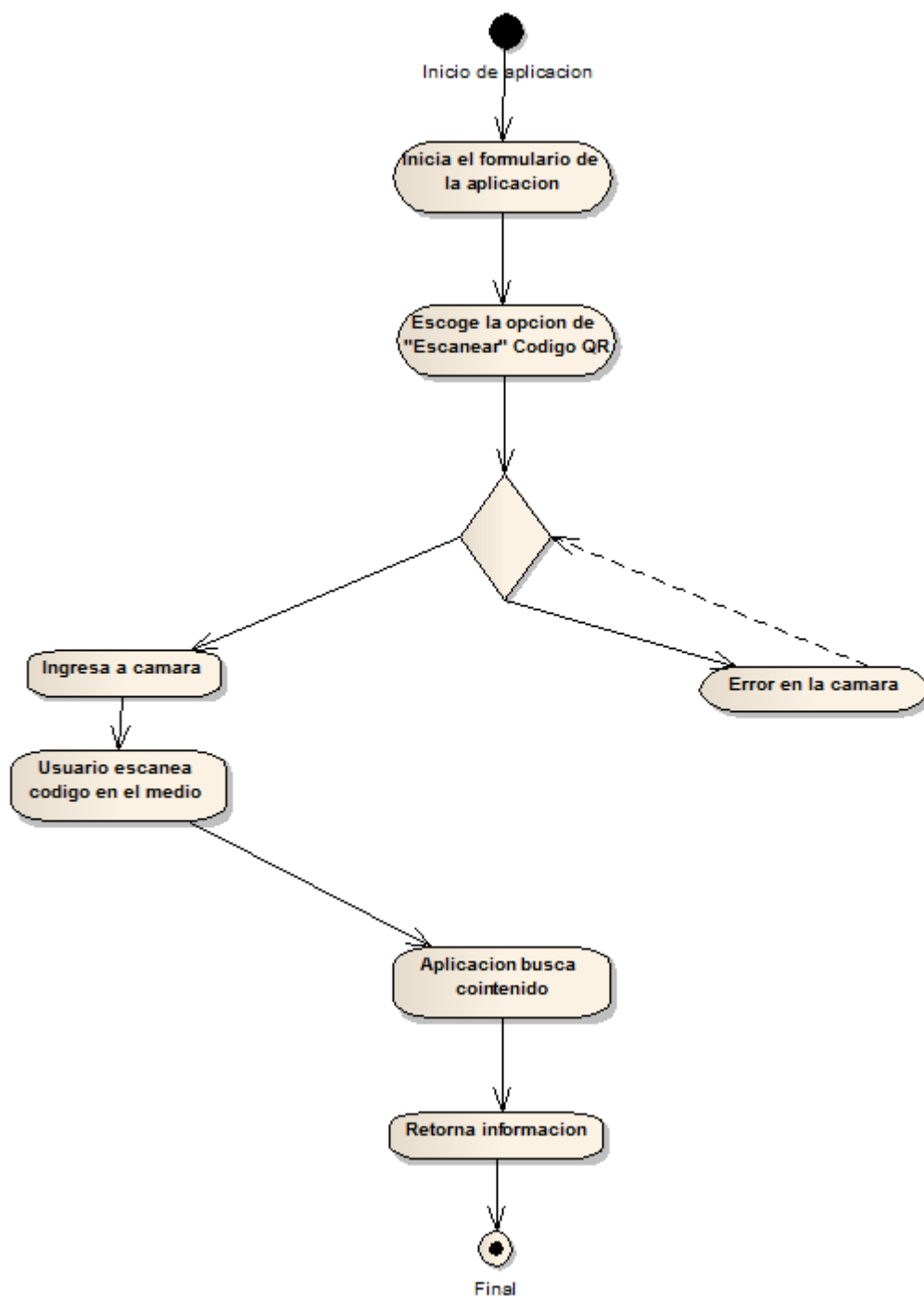


Figura 13. Diagrama de Estado.

Diagrama de secuencia de Escáner código

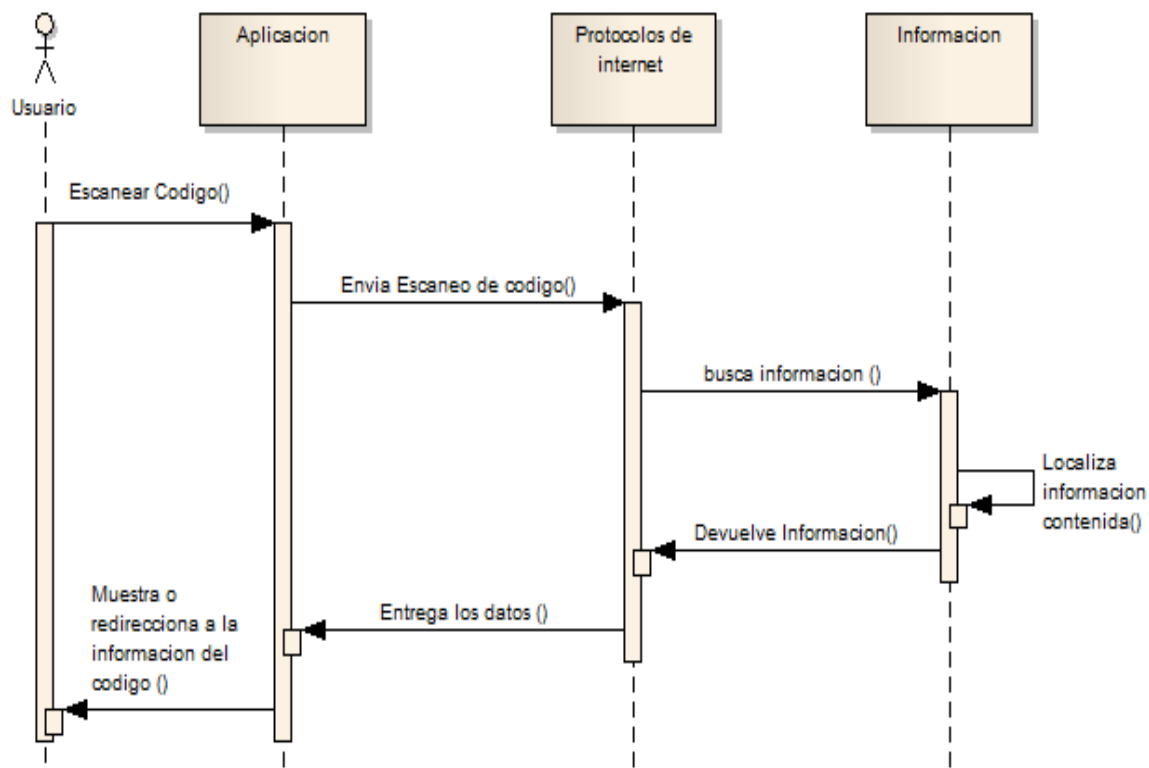


Figura 14. Diagrama de Secuencia.

Iteración 2:

Descripción	Historias/Tareas	Resultado	Fecha de entrega
Historias realizadas	H2	Finalizada	05/11/2016
Tareas de la historia H1	Codificar en lenguaje C# para Windows Phone	Realizadas	05/11/2016
Tareas de la historia H1	Pruebas funcionales del código	Realizadas	05/11/2016

Tabla 10. Iteración 2.

Clase PhotoCameraLuminanceSource

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ZXing;

namespace Tesis
{
    public class PhotoCameraLuminanceSource: LuminanceSource
    {
        public byte[] PreviewBuferY { get; private set; }

        public PhotoCameraLuminanceSource(int Width, int Heigh): base(Width,
Heigh)
        {
            PreviewBuferY = new byte[Width * Heigh];
        }

        public override byte[] Matrix
        {
            get { return (byte[])(Array)PreviewBuferY; }
        }

        public override byte[] getRow(int y, byte[] row)
        {
            if (row == null || row.Length < Width)
            {
                row = new byte[Width];
            }
        }
    }
}
```

```

        for (int i = 0; i < Height; i++)
        {
            row[i] = (byte)PreviewBuferY[i * Width + y];
        }
        return row;
    }
}
}

```

MainPage.xaml

```

<phone:PhoneApplicationPage
    x:Class="Tesis.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

```

<!--LayoutRoot es la cuadrícula raíz donde se coloca todo el contenido de la página-->

```

<Grid x:Name="LayoutRoot" Background="Transparent">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="**"/>
    </Grid.RowDefinitions>

```

<!-- NOTA PARA LA TRADUCCIÓN:

Para traducir las cadenas mostradas, copia sus valores a las claves con el nombre

correspondiente en el archivo de recursos del idioma neutro (AppResources.resx) de la aplicación y

reemplaza el valor de texto codificado de forma rígida entre las comillas de los atributos

con la cláusula de enlace cuya ruta de acceso apunte a ese nombre de cadena.

Por ejemplo:

```

Text="{Binding Path=LocalizedResources.ApplicationTitle,
Source={StaticResource LocalizedStrings}}"

```

Este enlace apunta al recurso de cadena de la plantilla denominado "ApplicationTitle".

Al agregar idiomas admitidos en la pestaña Propiedades del proyecto se creará un nuevo archivo resx por idioma que puede contener los valores traducidos de las

cadenas de la interfaz de usuario. El enlace de estos ejemplos hará que el valor de los atributos

se tome del archivo .resx correspondiente al valor de CurrentUICulture de la aplicación en tiempo de ejecución.

-->

<!--TitlePanel contiene el nombre de la aplicación y el título de la página-->

```
<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
  <TextBlock Text="Universidad Manuela Beltran" Style="{StaticResource
PhoneTextNormalStyle}" Margin="12,0"/>
  <TextBlock x:Name="te" Text="Lector QR" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
</StackPanel>
```

<!--ContentPanel. Colocar aquí el contenido adicional-->

```
<Grid x:Name="ContentPanel" Background="CornflowerBlue" Grid.Row="1"
Margin="12,0,12,0">
  <Button x:Name="ret" Content="Escanear" Background="Black"
Margin="30,44,122,459" Click="Button_Click"/>
</Grid>
```

<!--Quitar la marca de comentarios para ver una cuadrícula de alineación que ayuda a comprobar que los controles están alineados en los límites normales. La imagen tiene un margen superior de -32px para tener en cuenta la bandeja del sistema. Establécelo en 0 (o quite el margen) si la bandeja del sistema está oculta.

Antes de enviarla, quita este código XAML y la propia imagen.-->

```
<!--<Image Source="/Assets/AlignmentGrid.png" VerticalAlignment="Top"
Height="800" Width="480" Margin="0,-32,0,0" Grid.Row="0" Grid.RowSpan="2"
IsHitTestVisible="False" />-->
</Grid>
```

```
</phone:PhoneApplicationPage>
```

MainPage.xaml.cs

```
using System;
using System.Collections.Generic;
```

```

using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using Tesis.Resources;

namespace Tesis
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            NavigationService.Navigate(new Uri("/Escaner.xaml", UriKind.Relative));
        }
    }
}

```

Escaner.xaml

```

<phone:PhoneApplicationPage
    x:Class="Tesis.Escaner"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot es la cuadrícula raíz donde se coloca todo el contenido de la
    página-->
    <Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>

```

```

        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contiene el nombre de la aplicación y el título de la página-->
    <StackPanel Grid.Row="0" Margin="12,17,0,28"/>

    <!--ContentPanel. Colocar aquí el contenido adicional-->
    <Grid x:Name="ContentPanel" Background="Red" Grid.Row="1"
Margin="10,0,12,10">
        <Rectangle Margin="0,0,0,62">
            <Rectangle.Fill>
                <VideoBrush x:Name="videoBrush">
                    <VideoBrush.RelativeTransform>
                        <CompositeTransform x:Name="compositeTransform"
CenterX="0.5" CenterY="0.5">

                            </CompositeTransform>
                        </VideoBrush.RelativeTransform>
                    </VideoBrush>
                </Rectangle.Fill>
            </Rectangle>
            <ListBox Margin="10,10,10,62" x:Name="_matchetList" FontSize="30"
FontWeight="ExtraBold"></ListBox>
            <Grid x:Name="ContentPanel2" Grid.Row="1" Margin="10,656,12,10"/>
        </Grid>
    </Grid>

</phone:PhoneApplicationPage>

```

Escaner.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;
using Microsoft.Phone.Controls;
using Microsoft.Phone.Shell;
using Tesis.Resources;
using Microsoft.Devices;
using System.Windows.Threading;
using ZXing;
using ZXing.QrCode;
using ZXing.Common;

```



```

namespace Tesis
{
    public partial class Escaner : PhoneApplicationPage
    {
        private PhotoCamera photoCamera;
        private readonly DispatcherTimer dispatcherTimer = new DispatcherTimer();
        private QRCodeReader codeReader;
        private HyperlinkButton Link = new HyperlinkButton();

        public Escaner()
        {
            InitializeComponent();
        }

        protected override void OnNavigatedTo(NavigationEventArgs e)
        {
            photoCamera = new PhotoCamera(CameraType.Primary);
            photoCamera.Initialized += (s, a) =>
            {
                codeReader = new QRCodeReader();
                Dispatcher.BeginInvoke(() => { compositeTransform.Rotation =
photoCamera.Orientation; });
            };

            videoBrush.SetSource(photoCamera);

            dispatcherTimer.Interval = TimeSpan.FromMilliseconds(250);
            dispatcherTimer.Tick += (o, arg) => ScanPreviewBuffer();
            dispatcherTimer.Start();

            base.OnNavigatedTo(e);
        }

        protected override void OnNavigatedFrom(NavigationEventArgs e)
        {
            dispatcherTimer.Stop();

            base.OnNavigatedFrom(e);
        }

        private void ScanPreviewBuffer()
        {
            try
            {
                int width = Convert.ToInt32(photoCamera.PreviewResolution.Width);
            }
        }
    }
}

```

```

        int height = Convert.ToInt32(photoCamera.PreviewResolution.Height);
        PhotoCameraLuminanceSource _luminance = new
PhotoCameraLuminanceSource(width, height);
        photoCamera.GetPreviewBufferY(_luminance.PreviewBuferY);
        var binBitmap = new BinaryBitmap(new HybridBinarizer(_luminance));

        var result = codeReader.decode(binBitmap);

        if (result != null)
        {
            MessageBox.Show("Codigo Atrapado");
            Link.Content = "Abrir Link";
            Link.NavigateUri = new Uri(""+result);
            Link.TargetName = "_blank";
            ContentPanel2.Children.Add(Link);
            //Link.Click += new RoutedEventHandler(URLink_Click);
        }
    }
    catch {}
}
}
}

```

3.2.6.3.4. Fase XP: Producción

En la fase de producción se implementó aquellos cambios de más en los cuales el usuario agrega o modifica las historias de usuario para realizar un mejor desarrollo o de alta calidad en el objetivo final. Para la aplicación lectora de códigos QR no es necesario crear nuevas historias o incluir nuevas características al desarrollo ya que se el requerimiento inicial del cliente fue bastante claro y sencillo, por lo tanto no aplica para esta fase.

3.2.6.3.5. Fase XP: Mantenimiento

Como se puede observar, dentro de la misma simplicidad del desarrollo, esta fase no aplica por que no se mantiene el funcionamiento, sino que una vez realizado se entrega de manera definitiva.

3.2.6.3.6. Fase XP: Muerte del proyecto

Una vez no existen más historias de usuario para realizar entonces se hace entrega final del desarrollo y se da muerte al proyecto con una reunión final entregando todos los detalles de lo que se hizo y un manual técnico.

3.2.6.4. Fase de investigación: Cierre

Para el cierre del proyecto se buscó la aceptación del producto final con las evidencias de código fuente y funcionamiento de la aplicación. En la entrega final se hizo la revisión de las historias de usuario que hayan sido realizadas de las

cuales se mencionan anteriormente. Se da cumplimiento a cada una con la evidencia del desarrollo en la ejecución de cada iteración.

Se procede a facturar el proyecto por medio de los presupuestos que se realizó y se hace una liberación en el proyecto con respecto a los recursos utilizados o personal usado para dichas actividades.

4. RESULTADOS DE LA INVESTIGACION

4.1. ANALISIS DE RESULTADOS

Para el desarrollo del aplicativo llamado CODE QR desarrollado en Windows Phone, se hace uso e hizo hincapié en la metodología XP la cual funciona por diferentes fases donde se permite desarrollar de la manera más fácil y sin complicarse con estructuras de diseño o programación, evidenciado anteriormente.

Como paso final se construyó la aplicación y se deja evidencia del código para la posterior revisión como resultados de la aplicación mencionado en las iteraciones anteriormente:

Lector Gráficamente:



Figura 15. Código QR.

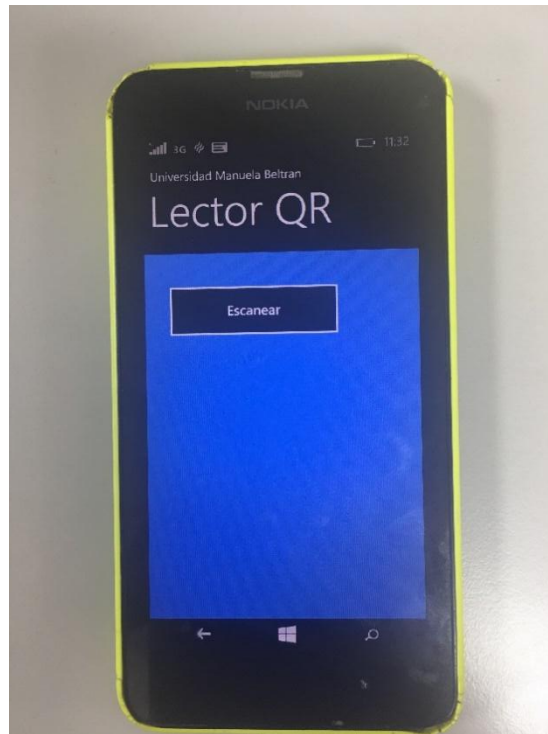


Figura 16. Lector QR

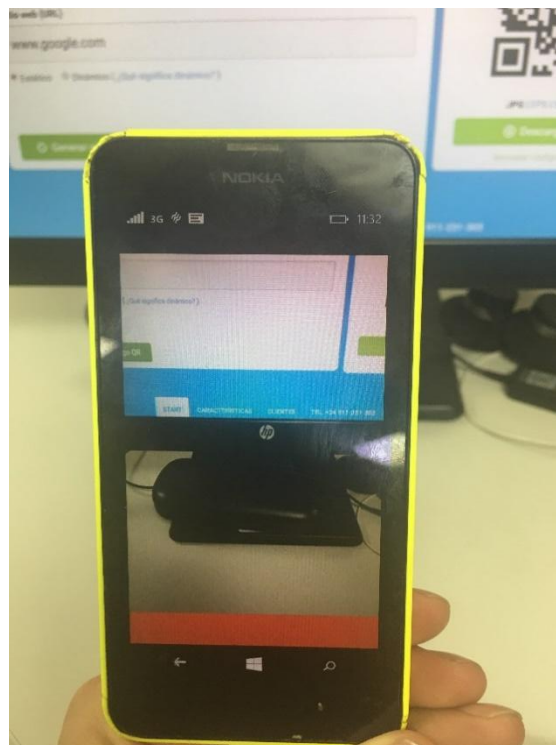


Figura 17. Camara Lector QR.

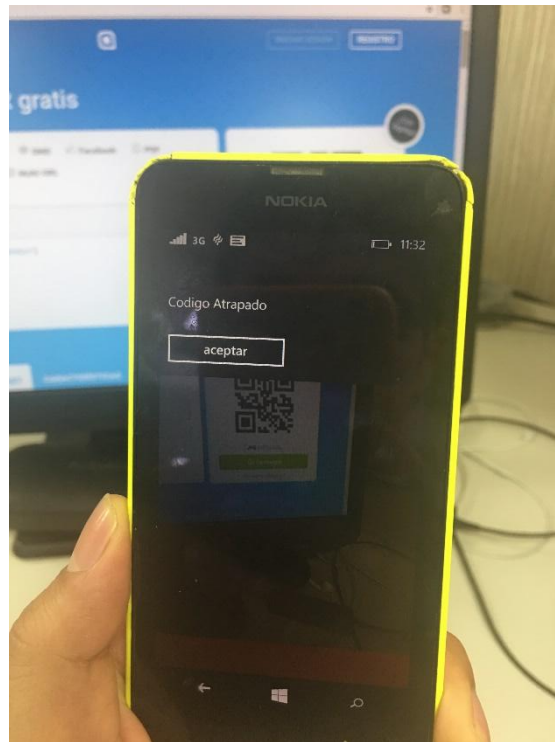


Figura 18. Código QR Atrapado.

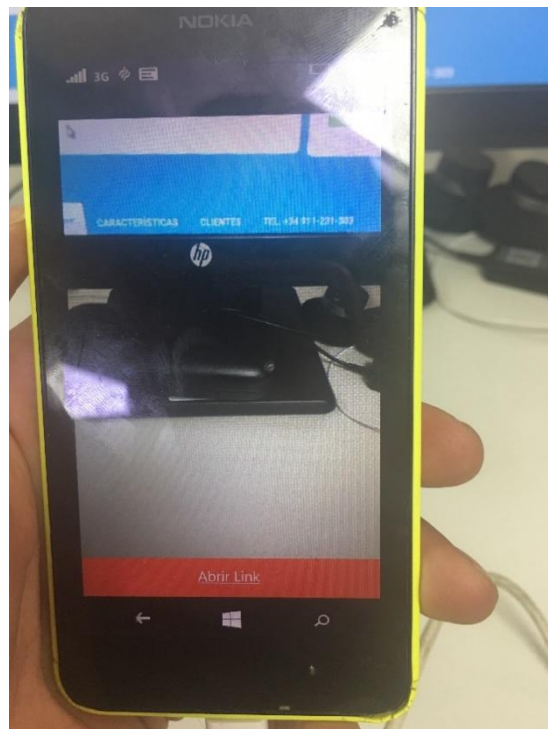


Figura 19. Generación link Desde QR.

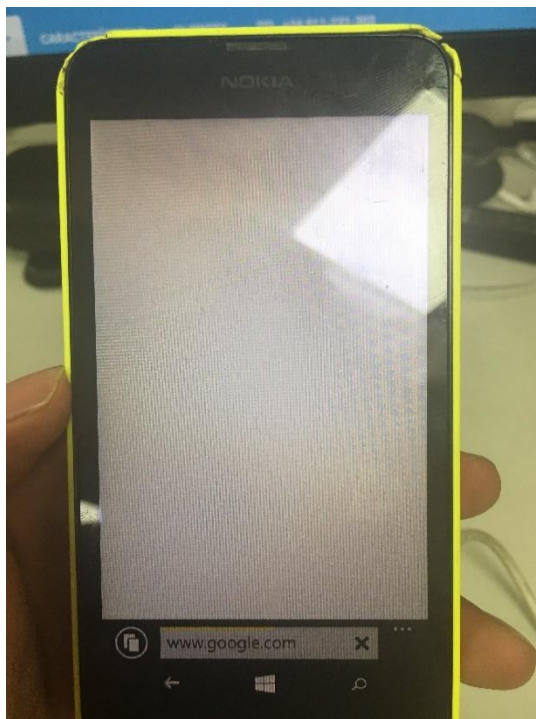


Figura 20. Información Final Código QR.

El lector de códigos QR analizara y recuperara información de links expuestos en un museo para la ampliación de la información de elementos expuestos en dicho recinto. Por ende se puede dar como un resultado satisfactorio el desarrollo de la aplicación y como se aplicó una serie de fases con su respectiva metodología para el desarrollo del mismo proyecto, suponiendo que se está vendiendo el producto a un cliente real.

5. CONCLUSIONES

Del anterior trabajo se concluye:

- Se explicó y entendió de manera clara y precisa todos aquellos conocimientos de la ingeniería de software que son más relevantes en el momento de hacer un desarrollo de una aplicación en cuanto a áreas, paradigmas, metodologías, y técnicas.
- Se listó las diferentes características de Windows Phone así como su entorno de desarrollo en el cual se pueden crear aplicaciones y a su vez se tuvo éxito en el proceso de absorción de dicho conocimiento para la aplicación realizada.
- Se entendió de manera sencilla y simple la tecnología de códigos QR y sus diferentes propósitos, donde fue aplicado al ámbito de un museo para el posterior desarrollo de una aplicación.
- Se desarrolló por medio de una metodología la aplicación en plataforma Windows Phone la cual permite escanear un código y retornar información para darle al usuario una experiencia más enriquecedora acerca de los elementos de un museo.

REFERENCIAS BIBLIOGRAFICAS

- [1] Luis E. Bayonet Robles. *Aprendizaje Móvil Aplicado en la Educación Usos prácticos ~ QR Code* [online]. México: Universidad autónoma de México. 9-oct-2010. Disponible en: <http://repositorial.cuaed.unam.mx:8080/jspui/handle/123456789/1209>
- [2] Denso Wave Incorporated. *Lectores de códigos QR* [online]. 2011. Disponible en: <http://www.codigos-qr.com/lectores-codigos-qr>
- [3] Juliana Gaviria García. *Nuevas tecnologías, nuevas formas de negocios: Smartphones y códigos QR en Chile* [online]. Chile: Universidad del rosario y universidad del desarrollo. 2012. Disponible en: <http://repository.urosario.edu.co/bitstream/handle/10336/3776/1020744780-2012.pdf?sequence=8&isAllowed=y>
- [4] María Soledad Gómez Vílchez. *QR Code en museos* [online]. 2010. Disponible en: <http://mediamusea.files.wordpress.com/2010/10/qr-code-en-museos.pdf>
- [5] Matos García, Bianchi Angleró, Pérez Ortiz, Santiago Guzmán y Silva Delgado. *Proyecto de ley, Estado Asociado de Puerto Rico* [online]. 2014. Disponible en: https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CBsQFjAA&url=http%3A%2F%2Fwww.camaraderepresentantes.org%2Ffiles%2Fpdf%2F%257B2A9BDA30-C8FC-44D2-90FA-AFB856F0F08C%257D.docx&ei=_Zw9VPqZKNHpggTnvYCIDA&usg=AFQjCNGzYPhsuorbl4EzFufzPxzvebCkLg&sig2=tkyq3tbC_RG1ihen0MKJKw&bvm=bv.77412846,d.eXY
- [6] Lucy Johanna Honores Chuchuca, Jenny Gabriela Vizuite Salazar. *Estudio estadístico comparativo entre sensores Android y Windows phone aplicado en la detección de movimientos telúricos* [online]. Ecuador: Escuela superior politécnica de Chimborazo. 2014. Disponible en: <http://dspace.esepoch.edu.ec/bitstream/123456789/3540/1/18T00554.pdf>
- [7] UNAD. *Lección 13 Desarrollo de Aplicaciones. Colombia: Universidad Nacional Abierta y a Distancia* [online]. 2013. Disponible en: http://datateca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_13_desarrollo_de_aplicaciones.html
- [8] Mike Armas. *Model Driven Architecture (MDA). MSDN Library: artículos técnicos* [online]. 2012. Disponible en: <https://msdn.microsoft.com/es-es/library/jj130728.aspx>
- [9] EcuRed. *Modelo de prototipos* [online]. 09/11/2016. Disponible en: https://www.ecured.cu/Modelo_de_Prototipos
- [10] Blogspot. *Metodología en cascada* [online]. 16/04/2013. Disponible en: <http://metodologiaencascada.blogspot.com.co/>
- [11] *Rational Unified Process (RUP)* [online]. Disponible en: <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf>
- [12] apr.com. *¿Qué es una base de datos y cuáles son los principales tipos? ejemplos: Mysql, Sqlserver, Oracle, Postgresql, Informix... (dv00204a)* [online]. Disponible en: http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=500

- [13] Croma Cultura. *¿Qué es un museo y cuantos tipos de museos existen?* [online]. 2015. Disponible en: <http://www.cromacultura.com/tipos-de-museos/>
- [14] Eusantana. *Los museos y las nuevas tecnologías*. 2014. Disponible en: <http://www.todalacultura.com/los-museos-y-las-nuevas-tecnologias/>
- [15] Amanda. 15/05/2015. *¿Por qué deberían los museos trabajar con generadores de códigos QR?* [online]. Disponible en: <https://uqr.me/es/blog/museos-generadores-codigos-qr/>
- [16] Felipe Ramírez García. *Cuatro grandes museos donde la tecnología es el alma de la exhibición* [online]. Enter.co. 2013. Disponible en: <http://www.enter.co/cultura-digital/tecnoviajero/cuatro-grandes-museos-donde-la-tecnologia-es-el-alma-de-la-exhibicion/>
- [17] Manuel Torres Gil. *Fundamentos del diseño de software* [online]. España: Universidad de Almería. Disponible en: <http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>
- [18] *Cuerpo de conocimientos de la ingeniería de software* [online]. México: Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias sociales y Administrativas. Disponible en: <http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/Complemento%20MateriaI%20Didactico/Maest-Ing-Soft-Sergio/Cuerpoconocimiento/Construcci%C3%B3n%20del%20software.htm>
- [19] SG Buzz. Ingeniería de Software. *Desarrollar es mucho más que programar* [online]. México. Disponible en: <https://sg.com.mx/content/view/444>
- [20] Arely Escobar. Paradigmas De la Ingeniería De Software [online]. Bligoo. Disponible en: <http://arelyescobar.bligoo.com.mx/paradigmas-de-la-ingenieria-de-software#.WAKinvI96Uk>
- [21] Ok Hosting. *Metodologías del desarrollo de software* [online]. OkHosting. Disponible en: <http://okhosting.com/blog/metodologias-del-desarrollo-de-software/>
- [22] *Rational Unified Process (RUP)* [online]. España: Universidad de Gerona. 2013. Disponible en: <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf>
- [23] Proyectosagiles.org. *Que es SCRUM* [online]. La web de Scrum. Disponible en: <https://proyectosagiles.org/que-es-scrum/>
- [24] Mex.tl. *Programación Extrema* [online]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
- [25] Alexey Mathotkin. *Control de Versiones. Producir Software de Código Abierto* [online]. 2013. Disponible en: <http://producingoss.com/es/vc.html>
- [26] Guillem Borrell. *El control de versiones* [online]. 2006. Disponible en: <http://torroja.dmt.upm.es/media/files/cversiones.pdf>
- [27] Luis Artola. *Tipos de pruebas automatizadas de software* [online]. Programania.net. 2009. Disponible en: <http://www.programania.net/disenio-de-software/tipos-de-pruebas-automatizadas-de-software/>
- [28] CCM. *Lenguajes de programación* [online]. 2016. Disponible en: <http://es.ccm.net/contents/304-lenguajes-de-programacion>
- [29] Tecnología W Definista. *Definición de Windows Phone* [online]. Conceptodefinicion.de. 2014. Disponible en: <http://conceptodefinicion.de/windows-phone/>

- [30] Jose Manuel Martinez Lainez. *Desarrollo y comercialización de una aplicación para una plataforma móvil* [online]. España: Escuela técnica superior de ingenieros industriales y de telecomunicación. Disponible en: <http://academica-e.unavarra.es/bitstream/handle/2454/5892/577862.pdf?sequence=1&isAllowed=y>
- [31] El vocero. *Códigos QR* [online]. 2016. Disponible en: <http://elvocero.com/tag/codigos-qr/>
- [32] Staff Capital 21. *Publican modificaciones a la ley de publicidad exterior de la cdmx* [online]. México: Capital 21. 22/01/2016. Disponible en: <http://www.capital21.df.gob.mx/publican-modificaciones-a-la-ley-de-publicidad-exterior-de-la-cdmx/>
- [33] Fernando Massa. *Los códigos QR invadieron el paisaje urbano* [online]. Argentina: La nación. Disponible en: <http://www.lanacion.com.ar/1424546-codigos-qr>
- [34] Carole Gray and Julian Malins. *Visualizing Research. A Guide to the Research Process in Art and Design* [online]. España: Universidad politecnica de Valencia. 2004. Disponible en: http://www.upv.es/laboluz/master/seminario/textos/proceso_investigacion.pdf
- [35] Susana Galarza Ganan. *Modelo XP (Xtreme Programming) para desarrollo de proyecto* [online]. Ecuador: Universidad de Milagro. Disponible en: <http://es.slideshare.net/johitaamiga/modelo-xp-para-desarrollo-de-proyecto>
- [36] Iso.or. *Systems and software engineering — Software life cycle processes. ISO/IEC* [online]. 2008. Disponible en: <https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en>
- [37] IEEE Computer Society. *Guía al cuerpo de conocimiento de la ingeniería de software SWEBOK* [online]. Estados Unidos. 2004. Disponible en: <http://www.cc.uah.es/drg/b/HispaSWEBOK.Borrador.pdf>
- [38] Daniel Álvarez. *Leer y generar códigos con Zxing* [online]. 2012. Disponible en: <http://zomwi.blogspot.com.co/2012/09/zxing.html>
- [39] Biljet App. *Librería zXing para lectura de códigos QR en Android* [online]. Disponible en: <https://biljetapp.wordpress.com/2013/03/04/presentacion/>