

Version100

por Jose David Gomez

| | | | |
|--------------------------------|-------------------------------------|-----------------------|--------|
| ARCHIVO | PROYECTO_IV_18-05-2017.DOCX (2.71M) | | |
| HORA DE LA ENTREGA | 20-MAY-2017 07:11A.M. | NÚMERO DE PALABRAS | 27905 |
| IDENTIFICADOR DE LA ENTREGA | 816696334 | SUMA DE CARACTERES | 165071 |

APLICACIÓN PARA DISPOSITIVO MÓVIL EN TECNOLOGÍA MICROSOFT
“WINDOWS PHONE” QUE PERMITA RECIBIR INFORMACIÓN DE LOS
CONTENIDOS DE UN MUSEO A PARTIR DE CÓDIGOS E IMAGENES QR

JOSE DAVID GOMEZ CRUZ

UNIVERSIDAD MANUELA BELTRÁN
PROYECTO IV
INGENIERÍA DE SISTEMAS
BOGOTA
2016

APLICACIÓN PARA DISPOSITIVO MÓVIL EN TECNOLOGIA MICROSOFT
"WINDOWS PHONE" QUE PERMITA RECIBIR INFORMACIÓN DE LOS
CONTENIDOS DE UN MUSEO A PARTIR DE CÓDIGOS E IMÁGENES QR

JOSE DAVID GOMEZ CRUZ

99
TRABAJO DE INVESTIGACIÓN PARA OPTAR AL TITULO DE INGENIERO

SOFTWARE

ASESOR:

JAIRO ALEJANDRO BUITRAGO ROMERO

INGENIERO DE SISTEMAS Y COMPUTACION ESPECIALISTA EN
ESPECIALISTE EN SOFTWARE PARA REDES, MAESTRIA EN INGENIERIA

INFORMATICA

UNIVERSIDAD MANUELA BELTRÁN

PROYECTO IV

INGENIERÍA DE SISTEMAS

BOGOTA

2017

DR. GUIDO ECHEVERRY PIEDRAHITA

Rector

DR. JAVIER LOPEZ MARTINEZ

Vicerrector académico

ING. DIANA ESTEY GUTIERREZ GALVIZ

85

Vicerrector de investigaciones

DRA. ROCIO BERNAL GARAY

Vicerrectora de calidad

INGENIERO HUGO MALAVER

Director Ingeniería de Software

DEDICATORIA

38

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 y mi novia 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.

AGRADECIMIENTOS

37

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 y a mi novia 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.

Tabla de contenido

| | |
|--|-----|
| INTRODUCCIÓN | 1 |
| 1. ANTECEDENTES..... | 3 |
| 1.1. PLANTEAMIENTO DEL PROBLEMA | 3 |
| 1.2. JUSTIFICACION | 4 |
| 1.3. DELIMITACIÓN DEL PROBLEMA..... | 5 |
| 1.4. PREGUNTA DEL PROBLEMA | 5 |
| 1.5. OBJETIVOS | 6 |
| 1.5.1. OBJETIVO GENERAL | 6 |
| 1.5.2. OBJETIVOS ESPECIFICOS..... | 6 |
| 2. MARCO TEORICO | 8 |
| 2.1. ANTECEDENTES DE LA INVESTIGACIÓN..... | 8 |
| 2.2. BASES TEORICAS Y FUNDAMENTOS TEORICOS | 10 |
| 2.2.1. Fundamentos del problema de investigación | 10 |
| 2.2.2. Fundamentos de ingeniería de software..... | 14 |
| 2.3. ANTECEDENTES DE LA EMPRESA Y CONTEXTO | 95 |
| 2.3.1. Códigos QR | 95 |
| 2.3.2. Librería para lector de códigos QR | 103 |
| 2.3.3. Contexto de la aplicación..... | 103 |
| 2.4. MARCO LEGAL | 105 |
| 3. METODOLOGIA | 107 |
| 3.1. TIPO DE INVESTIGACIÓN..... | 107 |
| 3.2. DISEÑO DE INVESTIGACIÓN | 108 |
| 3.2.1. Proceso de investigación | 108 |
| 3.2.2. Áreas, Paradigmas, Técnicas y Herramientas..... | 110 |
| 3.2.3. Cronograma..... | 118 |
| 3.2.4. Instrumentos de Investigación | 120 |
| 3.2.5. Presupuesto <i>Tabla 4</i> | 123 |
| 3.2.6. Aplicación de la metodología de Investigación | 124 |
| 3.2.6.1 Fase de Investigación: Inicio..... | 124 |

| | |
|--|-------------------------------------|
| 3.2.6.2 Fase de Investigación: Planificación..... | 124 |
| 3.2.6.3 Fase de Investigación: Ejecución de la metodología XP..... | 125 |
| 3.2.6.3.1 Fase XP: Exploración..... | 125 |
| 3.2.6.3.2 Fase XP: Planificación..... | 126 |
| 3.2.6.3.3 Fase XP: Iteraciones..... | 127 |
| 3.2.6.3.4 Fase XP: Producción..... | 142 |
| 3.2.6.3.5 Fase XP: Mantenimiento..... | 142 |
| 3.2.6.3.6 Fase XP: Muerte del proyecto..... | 143 |
| 3.2.6.1 Fase de Investigación: Cierre..... | 143 |
| 4. RESULTADOS DE LA INVESTIGACION | Error! Bookmark not defined. |
| 4.1. ANALISIS DE RESULTADOS..... | Error! Bookmark not defined. |
| 5. CONCLUSIONES | Error! Bookmark not defined. |
| REFERENCIAS BIBLIOGRAFICAS | Error! Bookmark not defined. |
| 4. RESULTADOS DE LA INVESTIGACION | 144 |
| 4.1. ANALISIS DE RESULTADOS..... | 144 |
| 5. CONCLUSIONES | 148 |
| REFERENCIAS BIBLIOGRAFICAS | 149 |

20
INDICE DE ILUSTRACIONES

| | |
|----------------------|------------------------------|
| Ilustración 1 | 66 |
| Ilustración 2 | 83 |
| Ilustración 3 | 84 |
| Ilustración 4 | 85 |
| Ilustración 5 | 86 |
| Ilustración 6 | 90 |
| Ilustración 7 | Error! Bookmark not defined. |
| Ilustración 8 | 109 |
| Ilustración 9 | 115 |
| Ilustración 10 | 118 |
| Ilustración 11 | 119 |
| Ilustración 12 | 129 |
| Ilustración 13 | 133 |
| Ilustración 14 | 134 |
| Ilustración 15 | 135 |
| Ilustración 16 | 144 |
| Ilustración 17 | 144 |
| Ilustración 18 | 145 |
| Ilustración 19 | 145 |
| Ilustración 20 | 146 |
| Ilustración 21 | 147 |

20
INDICE DE TABLAS

| | |
|----------------------|-----|
| Tabla 1 | 91 |
| Tabla 2 | 93 |
| Tabla 3 | 96 |
| Tabla 4 | 123 |
| Tabla 5 | 126 |
| Tabla 6 | 127 |
| Tabla 7 | 129 |
| Tabla 8 | 130 |
| Tabla 9 | 136 |

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

42

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 realizará 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 presenten.

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 ⁹⁸ 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 DEL PROBLEMA

¿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 intérprete información gráfica de códigos QR y realice una posterior entrega de información, entorno a un museo.

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.

13

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]

13

Sus orígenes proceden, a partir de un código de matriz (código de barras de dos dimensiones) creado por la empresa japonesa Denso-Wave en 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]

13
El uso de los datos se puede traducir por medio de un código QR por cualquier generador de QR, 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, BIDIs, DataMatrix, etc; Se pude decir a partir de esto que los más famosos y usados son los llamados DataMAtrix, creados 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 23 existen dos tipos diferentes: el japonés JIS X 0510 creado por la JIS y distribuido en enero de 1999 y el correspondiente estándar de la ISO, ISO/IEC 18004 aprobado en junio de 2000 y revisado en 2006 (ISO/IEC 18004:2006); Desde 1999 la empresa que patento 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

23

los “QR codes” como etiquetas en las que la gente podía 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 “μουσεῖον” (mouseion), el cual hace

68

referencia a un templo dedicado a las nueve musas las cuales son Calíope, Clío, Erato, Euterpe, Melpómene, Polimnia, Talía, Terpsíclore 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: Acá 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 ⁸⁹ desde finales del siglo XIX hasta 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 folclor 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: Tratan sobre ⁸⁸ la navegación y todo lo relacionado con el mar.
- 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 ingeniera de software

2.2.2.1.1. Requerimientos del software

1

El área del conocimiento de los requisitos del software se refiere al análisis, a la especificación, y a la validación de los requisitos del software. Está extensamente reconocido dentro de la industria del software que los proyectos de la ingeniería de software son críticamente vulnerables cuando estas actividades se realizan mal.

[37]

Los requisitos del software expresan las necesidades y los apremios colocados 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

1

2.2.2.1.1.1.1. Definición de un requisito del software

Básicamente, un requisito del software es una característica que se debe exhibir para solucionar un cierto problema en el del mundo real. La guía se refiere a requisitos de “software” porque se refiere a los problemas que se tratarán por el software. Por lo tanto, un requisito del software es una característica que se debe exhibir por el software desarrollado o adaptado para solucionar un problema particular. El problema puede ser automatizar la parte de una tarea de alguien que utilizará el software, para apoyar los procesos del negocio de la organización que ha comisionado el software, a corregir los defectos del software existente, al control de dispositivos, y muchos más. El funcionamiento de los usuarios, los procesos del negocio, y los dispositivos es típicamente complejo. Por lo tanto, los requisitos de software son típicamente una combinación compleja de requisitos de

diversa gente en diversos niveles de una organización y del ambiente en el cual el software funcionará. [37]

2.2.2.1.1.2. Producto y requisitos del proceso

Los parámetros del producto son requisitos en software para ser convertido, un parámetro de proceso es esencialmente un constreñimiento en el desarrollo del software. Éstos se conocen a veces como requisitos de proceso. [37]

2.2.2.1.1.3. Requisitos funcionales y no funcionales

Los requisitos funcionales describen las funciones que el software va a ejecutar; por ejemplo, ajustarse a un formato de texto o modular una señal. Se conocen también como capacidades. [37]

Los requisitos no funcionales son los que actúan para obligar la solución. Los requisitos no funcionales se conocen a veces como apremios o requisitos de calidad. Pueden ser clasificados más a fondo según si son requisitos de funcionamiento, requisitos de capacidad de mantenimiento, requisitos de seguridad, requisitos de confiabilidad, o uno de muchos otros tipos de requisitos del software. [37]

2.2.2.1.1.4. Características inesperadas

Algunos requisitos representan características inesperadas del software esto es, los requisitos que no pueden ser tratados por un solo componente, pero que su satisfacción va a depender de cómo todos los componentes de software interactúan. [37]

1

2.2.2.1.1.5. Requisitos cuantificables

Los requisitos del software se deben indicar tan clara e inequívocamente como sea posible, y cuantitativamente. Es importante evitar requisitos vagos e inverificables que dependen para su interpretación del juicio subjetivo. [37]

1

2.2.2.1.1.6. Requisitos del sistema y requisitos del software

En este asunto, el sistema significa —una combinación recíproca de los elementos para lograr un objetivo definido. Éstos incluyen el hardware, software, soporte lógico inalterable, gente, información, técnicas, instalaciones, servicios, y otros elementos de apoyo, según lo definido por el consejo internacional sobre la ingeniería de sistemas. [37]

2.2.2.1.2. Proceso de los requisitos

2.2.2.1.1.2.1. Modelos de proceso

1

- No es una actividad anticipada discreta del ciclo vital del software, sino un proceso iniciado en principio de un proyecto y a continuación refinado a través del ciclo vital [37]
- Identifica los requisitos del software como elementos de configuración, y los maneja usando las mismas prácticas de gerencia de la configuración del software como otros productos de los procesos del ciclo vital del software [37]
- Necesita ser adaptado a la organización y al contexto del proyecto[37]

2.2.2.1.1.2.2. Agentes de proceso

1

Este proceso es fundamental interdisciplinario, y el especialista de los requisitos necesita mediar entre el dominio del tenedor de apuestas y el de la tecnología de dotación lógica. Hay mucha gente implicada además del especialista de los requisitos, cada uno de ellos tiene una función en el software. [37]

1

2.2.2.1.1.2.3. Ayuda y gerencia de proceso

Introduce los recursos de la gerencia de proyecto requeridos y consumidos por el proceso de los requisitos. Él establece el contexto para la primera subzona (definición de la iniciación y del alcance) de la tecnología de dotación lógica de la gerencia. [37]

1

2.2.2.1.1.2.4. Calidad y mejora de proceso

Se refiere al gravamen de la calidad y de la mejora del proceso de los requisitos. Su propósito es acentuar el papel dominante que los requisitos procesan en términos de coste y puntualidad de un producto de software, y de la satisfacción del cliente con ello. [37]

2.2.2.1.1.3. Captura de los requisitos

1

La captura de los requisitos se refiere de donde vienen los requisitos del software y cómo el ingeniero de software puede recogerlos. Es la primera etapa en la construcción de una comprensión del problema que el software requiere solucionar. Es fundamental una actividad humana, y es donde identifican a los stakeholders y las relaciones se establecen entre el equipo del desarrollo y el cliente. También se conoce como "descubrimiento de los requisitos", y "adquisición de los requisitos". [37]

1

2.2.2.1.4. Análisis de requisitos

Este asunto se refiere al proceso de analizar requisitos para [37]

- Detectar y resolver los conflictos entre los requisitos
- Descubrir los límites del software y cómo debe obrar recíprocamente con su ambiente
- Elaborar los requisitos del sistema para derivar requisitos software

La vista tradicional del análisis de requisitos ha sido que esté reducida a modelado conceptual utilizando uno de varios métodos de análisis tales como Análisis Estructurados y Técnicas de Diseño. Mientras que el modelado conceptual es importante, nosotros incluimos la clasificación de requisitos para ayudar a informar a compensaciones entre los requisitos (clasificación de los requisitos) y el proceso de establecer estas compensaciones (negociación de los requisitos). [37]

2.2.2.1.5. Especificación de requisitos

1

El término, “especificación” de requisitos del software se refiere típicamente a la producción de un documento, o a su equivalente electrónico, que puede estar sistemáticamente repasado, evaluado, y aprobado. Para los sistemas complejos, particularmente éhos que implican componentes no-software, se elaboran tres tipos de documentos: definición de sistema, sistema requisitos, y requisitos del software. Para sistemas simples, solamente el tercero de éstos es requerido. Los tres documentos se describen aquí, entendiendo que combinados pueden ser apropiados. [37]

1

2.2.2.1.1.6. **Validación de los requisitos**

Los documentos de los requisitos pueden estar conformes a la validación y procedimientos de verificación. Los requisitos pueden ser validados para asegurarse de que el ingeniero del software entiende los requisitos, y es también importante para verificar que un documento de requisitos se conforma con la compañía de los estándares, y éste es comprensible, constante, y finito. Las notaciones formales ofrecen la ventaja importante de permitir que las dos características pasadas sean probadas (en un sentido estricto, por lo menos). Los documentos de los requisitos son conformes a las mismas prácticas de gerencia de la configuración del software como los otros puntos relevantes de los procesos del ciclo de vida del software. [37]

97

2.2.2.1.2. **Diseño de software**

Se sabe que 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]

Siendo más específicos, por un lado existe el diseño arquitectónico el cual permite describir la arquitectura y organización de alto nivel de un determinado sistema, donde se identifica de manera sistemática 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 concentra mucho conocimiento con el fin de entender a fondo los principios como la abstracción, el acoplamiento, la cohesión, la descomposición y

encapsulación para así mismo tener diseños robustos. Por tal razón es necesario saber resolver aspectos prácticos como la persistencia de datos, sistemas distribuidos, peticiones concurrentes, manejo de eventos, recuperación a fallas, etc. [17]

2.2.2.1.2.1. Fundamentos del diseño del software

1 El software no es el único campo donde está implicado el diseño. En el sentido amplio, podemos ver diseño como forma de solucionar un problema. El diseño del software generalmente se considera un proceso de dos etapas: El diseño arquitectónico describe cómo el software se descompone y se organiza en los componentes (la arquitectura) del software y el diseño detallado describe el comportamiento específico de estos componentes. La salida de este proceso es un sistema de modelos y los artefactos que registran las decisiones principales que se han tomado. [37]

1 Los principios del diseño del software, también llamados técnicas permisibles, son nociones dominantes que consideran fundamental a los diversos acercamientos y conceptos del diseño del software. Las técnicas que lo permiten son las siguientes: [37]

- Abstracción
- Acoplador y cohesión
- Descomposición y modularización
- Encapsulación del ocultar de la información
- Separación del interfaz y de la puesta en práctica
- Desahogo, lo completo y deshaciéndose de lo primitivo

1

2.2.2.1.2.2. Cuestiones claves en diseño del software

Se tiene que tener en cuenta a la hora de diseñar software una serie de principios claves: [37]

- Conurrencia
- Control y dirección de acontecimientos
- Distribución de componentes
- **Dirección del error y de excepción y tolerancia de fallos**
- Interacción y presentación
- Persistencia de los datos

1

2.2.2.1.2.3. Estructura y arquitectura del software

En su sentido terminante, una arquitectura del software es “una descripción de los subsistemas y de los componentes de un sistema de software y de las relaciones entre ellas”. La arquitectura procura así definir la estructura interna [37]

2.2.2.1.2.3.1. Estructuras y puntos de vista arquitectónicos

1
Estas facetas a menudo se llaman las opiniones: “Una visión representa un aspecto parcial de una arquitectura del software que demuestre características específicas de un sistema de software”. Estas visiones distintas pertenecen a las ediciones distintas asociadas a diseño del software - por ejemplo, la visión lógica (que satisface los requisitos funcionales) contra la visión de proceso (ediciones de la concurrencia) contra la visión física (ediciones de la distribución) contra la opinión del desarrollo (cómo el diseño se analiza en unidades de la puesta en práctica). [37]

- 1
- Estructura general (por ejemplo, capas, pipas, y filtros, pizarra)
 - Sistemas distribuidos (por ejemplo, servidor de cliente, tres gradas, corredor)
 - Sistemas interactivos (por ejemplo, regulador de la Modelo-Vista, Presentación Abstracción-Control)
 - Sistemas adaptables (por ejemplo, micro-núcleo, reflexión)

2.2.2.1.2.3.2. Patrones del diseño

1
Un patrón es “una solución común a un problema común en un contexto dado”. Mientras que los estilos arquitectónicos se pueden ver como patrones que describen la organización de un nivel alto del software (su arquitectura macro); otros patrones del diseño se pueden utilizar para describir los detalles en un nivel más bajo, más local (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

Varios atributos son generalmente importantes para obtener un diseño del software de buena calidad varios “ilities” (capacidad de mantenimiento, portabilidad, testeo, trazabilidad), también “nesses” (corrección, robustez), incluyendo la “aptitud del propósito”. [37]

2.2.2.1.2.4.2. Técnicas de evaluación y calidad del análisis

Varias técnicas pueden ayudar a asegurar la calidad de un diseño del software:
[37]

- Revisiones de diseño del software
- Análisis estático
- Simulación y prototipado

2.2.2.1.2.4.3. 1 Medidas

Las medidas se pueden utilizar para determinar o para estimar cuantitativamente varios aspectos del tamaño, de la estructura, o de la calidad de un diseño del software. La mayoría de las medidas se han propuesto que dependen generalmente del acercamiento usado para producir el diseño. [37]

2.2.2.1.2.5. 1 Notaciones del diseño del software

Muchas notaciones e idiomas existen para representar los artefactos del diseño del software. Algunos se utilizan principalmente para describir la organización estructural de un diseño, otras para representar comportamiento del software. Ciertas notaciones se utilizan sobre todo durante el diseño arquitectónico y otros principalmente durante el diseño detallado, aunque algunas notaciones se pueden utilizar en ambos pasos. Además, algunas notaciones se utilizan sobre todo en el contexto de métodos específicos. [37]

2.2.2.1.2.5.1. Descripción estructural

1 Las siguientes notaciones, sobre todo (pero no siempre) gráficas, describen y representan los aspectos estructurales del diseño de software las cuales, describen los componentes principales y cómo se interconectan (visión estática): [37]

- Lenguajes descriptivos de la arquitectura

- Diagramas de la clase y objeto
- Diagramas de componente
- **Tarjetas del colaborador de la responsabilidad de la clase**
- 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

1
Se utilizan para describir el comportamiento dinámico del software y de los componentes. [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

1
Existen varias estrategias generales para ayudar a dirigir el proceso de diseño. Al contrario que en las estrategias generales, los métodos son más específicos,

sugieren y proporcionan generalmente un sistema de notaciones que se utilizarán con el método, una descripción del proceso que se utilizará después del método y un sistema de pautas al usar el método. Tales métodos son útiles como medios de transferir conocimiento y como marco común para los equipos de los ingenieros de software. [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 centra el conocimiento basado 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, es decir, manipula la creación y aplicación de algoritmos para solucionar problemas y de la misma forma implementar una solución por medio de un lenguaje de programación. Los retos acá se encuentran en minimizar la complejidad y cumplir ciertos criterios o estándares que ayudan a entender 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

Los fundamentos de la construcción del software incluyen:

- Minimizar la complejidad: El principal factor que hace que la gente utilice ordenadores consiste en la limitadísima capacidad que tiene para retener estructuras complejas e información en su memoria operativa, especialmente durante largos períodos de tiempo. Esto lleva a uno de los más fuertes impulsores de la construcción del software: minimizar la complejidad. La necesidad de reducir la complejidad se aplica esencialmente a todo aspecto de la construcción del software, y es de crítica importancia para el proceso de verificación y pruebas de las construcciones del software. [37]
- Anticiparse a los cambios: La mayoría del software cambiará a lo largo del tiempo, y el anticiparse a los cambios dirige muchos aspectos de la construcción del software. El software es inevitablemente parte de los ambientes externos que cambian continuamente, y los cambios en esos ambientes externos afectan al software de diversos modos. [37]
- Construir para verificar: significa construir software de tal manera que los ingenieros del software puedan sacar a relucir los fallos con facilidad al estar escribiendo el código, además de cuando realizan pruebas independientes y actividades operacionales. Las técnicas específicas que sirven de base para construir con vistas a verificar incluyen el seguimiento de estándares de codificación que permitan las revisiones del código, las pruebas unitarias, la organización del código que permita pruebas automáticas, y el uso restringido de estructuras de lenguaje que sean complejas o difíciles de entender, entre otras. [37]

- Estándares en la construcción: Los estándares que afectan directamente a elementos de la construcción incluyen: [37]
 - Métodos de comunicación (por ejemplo, estándares para los formatos de los documentos y de los contenidos)
 - Programación de lenguajes (por ejemplo, estándares de lenguaje para lenguajes como Java y C++)
 - Plataformas (por ejemplo, estándares de interfaces del programador para llamadas al sistema operativo)
 - Herramientas (por ejemplo, estándares diagramáticos para notaciones como UML (Lenguaje Unificado de Modelado))

2.2.2.1.3.2. Gestión de la Construcción

1

2.2.2.1.3.2.1. Modelos de Construcción

Se han creado numerosos modelos para el desarrollo del software, algunos de los cuales ponen más énfasis en la construcción que otros. Algunos modelos son más lineales que otros desde el punto de vista de la construcción tales como los modelos en cascada y los del ciclo de vida de entregas por etapas. Estos modelos tratan la construcción como una actividad que sucede sólo después de que se haya completado un significativo trabajo con los prerequisitos, incluyendo un trabajo detallado sobre los requisitos, un extensivo trabajo sobre el diseño y una planificación detallada. [37]

1

2.2.2.1.3.2.2. Planificación de la Construcción

La elección de un método de construcción es un aspecto clave de la planificación de la actividad de construcción. La elección de un método de construcción afecta hasta dónde se realizan los requisitos de construcción, el orden en el que se realizan, y el grado hasta el que se espera que se completen antes de que comience el trabajo de construcción. El modo como se afronta la construcción afecta a la habilidad del proyecto para reducir la complejidad, anticipar cambios y construir para verificar. 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]

1

2.2.2.1.3.2.3. Medición de la Construcción

Se pueden medir 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 mediciones pueden ser útiles para propósitos de gestión de la construcción, asegurando la calidad durante la construcción, mejorando los procesos de construcción. [37]

1

2.2.2.1.3.3. Consideraciones Prácticas

La construcción es una actividad en la cual el software se las tiene que ver con restricciones arbitrarias y caóticas del mundo real, y hacer exactamente lo que piden. Gracias a su proximidad a las restricciones del mundo real, la construcción está guiada por consideraciones prácticas más que otras. [37]

2.2.2.1.3.3.1. Diseño de la Construcción

1

Algunos proyectos asignan una mayor actividad de diseño a la construcción; otros a una fase que se centra explícitamente en el diseño. Independientemente de su asignación exacta, en el nivel de construcción también se trabaja algo el diseño detallado y ese trabajo de diseño tiende a estar dictaminado por restricciones inamovibles impuestas por un problema del mundo real que está siendo afrontado por el software. [37]

2.2.2.1.3.3.2. Lenguajes de Construcción

7

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

2.2.2.1.3.3.3. Codificación

1

Las consideraciones siguientes se aplican a la actividad de construcción del código del software: [37]

- Técnicas para crear código fuente comprensible, que incluye la asignación de nombres y el esquema del código fuente
- Utilización de clases, tipos enumerados, variables, constantes predefinidas, y otras entidades similares
- Utilización de estructuras de control

- Tratamiento de las condiciones de error tanto los errores planeados como las excepciones (la entrada de datos malos, por ejemplo)
- 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 y disciplina en el acceso serial a recursos reutilizables (incluyendo threads o bloqueos de bases de datos)
- Organización del código fuente (en declaraciones, rutinas, clases, paquetes u otras estructuras)
- Documentación del código
- Puesta a punto del código

2.2.2.1.3.3.4. Pruebas de Construcción

1

Construir implica dos tipos de pruebas, que por lo general las realiza el mismo ingeniero del software que escribió el código: [37]

- Pruebas unitarias
- Pruebas de integración

El propósito de las pruebas de construcción es reducir la brecha entre el tiempo en el que se introducen fallos en el código y el tiempo en el que se detectan esos fallos. En algunos casos, las pruebas de construcción se llevan a cabo después de la escritura del código. En otros casos, se pueden elaborar casos de pruebas antes de que se escriba el código. [37]

2.2.2.1.3.3.5. Reutilización

1

El implementar la utilización del software conlleva algo más que crear y utilizar librerías de recursos. Requiere formalizar la práctica de la reutilización por medio de la integración de procesos y actividades de reutilización en el ciclo de vida del software. Sin embargo, la reutilización tiene suficiente importancia en la construcción del software como para dedicarle aquí un tema. 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

2.2.2.1.3.3.6. Calidad de la Construcción

1

La técnica o técnicas específicas elegidas dependen de la naturaleza del software que se está construyendo, así como del conjunto de habilidades de los ingenieros del software que llevan a cabo la construcción: [37]

- Las pruebas unitarias y las pruebas de integración
1
- Pruebas de Construcción
1
- 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 encontramos 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 diferentes técnicas que ayudan a la simplicidad del mismo proceso, que su objetivo final es medir la respuesta correcta del software, por lo que se puede usar tablas de decisión, análisis de fronteras, máquinas de estados e incluso hasta la misma experiencia. [19]

1

Las pruebas del software consisten en verificar el comportamiento de un programa dinámicamente a través de un grupo finito de casos de prueba, debidamente seleccionados del, típicamente, ámbito de ejecuciones infinito, en relación al comportamiento esperado. [37]

5

La definición anterior corresponde con aspectos esenciales en la identificación del “Área de Conocimiento de las Pruebas del Software”. En particular:

- Dinámicamente: Este término significa que hacer pruebas siempre supone ejecutar el programa con entrada de datos (valorados). Para precisar, es preciso afirmar que la entrada de valores no es siempre suficiente para definir una prueba, dado que un sistema complejo y no determinista podría

tener diferentes comportamientos con la misma entrada de datos,

dependiendo del estado en el que se encuentre. [37]

1

- Finito: Incluso en programas sencillos, teóricamente podría haber tantas pruebas que realizar, que hacer pruebas exhaustivas podría llevar meses o años. Esta es la razón por la que en la práctica el grupo completo de pruebas se podría considerar infinito. Hacer pruebas siempre supone un compromiso entre recursos y calendarios de trabajo limitados, por un lado, y necesidades inherentes de pruebas ilimitadas, por otro. [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. La forma de identificar el criterio de selección de pruebas más apropiado para un conjunto de condiciones particulares es un problema complejo; en la práctica se usa la experiencia en el diseño de pruebas y técnicas de análisis de riesgo. [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. El comportamiento observado se puede comprobar con los resultados esperados por el usuario (normalmente conocido como pruebas de validación), con las especificaciones (pruebas de verificación), o,

finalmente, con el comportamiento anticipado de requerimientos implícitos o expectativas razonables. [37]

2.2.2.1.5. Mantenimiento del software

Cuando se habla de mantenimiento, se refiere a modificaciones dentro del software que implican la prevención de fallas (de manera preventiva), así como para corregirlas (correctivo), también para mejorar el desempeño (perfectivo) o para adaptación a cambios (adaptivo). Hoy en día 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]

1
El mantenimiento de software es una parte del ciclo de vida de software. Sin embargo, históricamente, no ha recibido el mismo grado de atención que otras fases del ciclo de vida. Históricamente, el desarrollo de software ha tenido un perfil mucho más alto que el mantenimiento de software en la mayor parte de organizaciones. Esto ha cambiado ahora, las organizaciones se esfuerzan en exprimir al máximo su inversión de desarrollo de software por lo que desean que el software funcione tanto tiempo como sea posible. Las preocupaciones sobre el efecto 2000, enfocó una atención significativa en la fase de mantenimiento de software, y el paradigma Open Source ha atraído la atención a la cuestión de mantener software desarrollado por otros. [37]

1

2.2.2.1.5.1. El Proceso de Mantenimiento

La subárea Proceso de Mantenimiento proporciona referencias y normas para poner en práctica el proceso de mantenimiento de software. El punto de Actividades de Mantenimiento diferencia el mantenimiento del desarrollo y muestra su relación con otras actividades de la ingeniería del software. La necesidad del proceso de ingeniería de software está bien documentada. Los modelos CMMI (Modelo de capacidad y madurez) se aplican a procesos de mantenimiento de software, y son similares a los procesos de los desarrolladores. [37]

6

- Procesos de Mantenimiento: Los procesos de Mantenimiento proporcionan actividades necesarias y 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 realizan el análisis, el diseño, la codificación, pruebas, y la documentación. Ellos deben rastrear exigencias en sus actividades tal cual hechas en el desarrollo, y la documentación de actualización como el cambio de líneas de fondo. [37]

1

Hay un número de procesos, actividades, y prácticas que son únicas al mantenimiento de software, por ejemplo:

- Transición: una secuencia controlada y coordinada de actividades durante las cuales el software es transferido cada vez más del desarrollador al mantenedor [37]
- La Aceptación/Rechazo de Petición de Modificación: el trabajo de petición de modificación sobre un cierto tamaño/esfuerzo/complejidad puede ser rechazado por mantenedores y desviado a un desarrollador [37]
- La petición de Modificación y el Escritorio de Ayuda de Informe de Problema: una función de apoyo de usuario final que provoca la evaluación, la ordenación, y de presupuesto de solicitud de modificación [37]
- El Análisis de Impacto [37]
- El Apoyo de Software: ayuda y aconseja a usuarios que solicitan información (por ejemplo, reglas de gestión, validación, datos que quieren decir y ad hoc solicita/hace un informe) [37]
- Los Acuerdos de Nivel de Servicio (SLAs) y los contratos de mantenimiento especializados (específicos de dominio) que son responsabilidad de los mantenedores [37]

2.2.2.1.5.2. Las técnicas para el Mantenimiento

1

- Comprensión de Programa: Los Programadores gastan un tiempo considerable en la lectura y el entendimiento de programas para poner en práctica los cambios. Los navegadores de código son instrumentos claves

para la comprensión de programa. La documentación clara y concisa puede ayudar en la comprensión de programa [37]

1

- **Reingeniería:** Reingeniería se define como el examen y la alteración de software para reconstituirlo en una nueva forma, e incluye la puesta en práctica subsecuente de la nueva forma. [37]
- **Ingeniería de revés:** La ingeniería de Revés es el proceso de analizar el software para identificar los componentes del software y sus relaciones mutuas y crear las representaciones del software en otra forma o en los niveles más altos de abstracción. La ingeniería de revés es pasiva; esto no cambia el software, o causa el nuevo software. Los esfuerzos de la ingeniería del revés producen gráficos de llamada y gráficos de flujo de control del código original. Un tipo de ingeniería de revés es la nueva documentación. [37]

48

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, cuyos resultados se consideran de alta importancia 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

1

Los requerimientos del software definen las características de calidad requeridas de ese software e influyen en los métodos de medición y criterios de aceptación para evaluar estas características. [37]

- 1 • Valor y coste de la calidad: El concepto de calidad no es tan simple como parece, para un ingeniero de productos hay muchas calidades deseadas relevantes para una perspectiva determinada de un producto, para que esto pueda ser tratado y determinado en el tiempo las exigencias de producto son puestas por escrito. Las características de calidad pueden requerirse o no, o se pueden requerir en un mayor o menor grado, y pueden hacerse compensaciones entre ellas. [37]
- 7 • Modelos y Características de Calidad: La terminología para las características de calidad del software difiere de una taxonomía (o modelo de calidad de software) a otra, cada modelo quizás tenga un número diferente de niveles jerárquicos y un número total diferente de 67 características. Acá se presentan 2 aspectos como la calidad del proceso en la ingeniería del software y calidad de producto software. [37]
- 5 • Mejora de Calidad: La calidad de los productos software puede ser mejorada mediante un proceso iterativo de mejora continua que requiere control de dirección, coordinación, y retroalimentación de muchos procesos simultáneos: (1) los procesos de ciclo de vida de software, (2) El proceso de detección de error/defecto, retirada de los mismo y prevención, (y 3) el proceso de mejora de calidad. [37]

1 2.2.2.1.6.2. Procesos de Gestión de Calidad del Software

La gestión de calidad de software resulta de aplicación a todas las perspectivas de procesos de software, productos, y recursos. Esto define procesos, propietarios

de proceso, y requerimientos para aquellos procesos, medidas del Proceso y sus correspondientes salidas, y canales de retroalimentación. [37]

Los procesos de gestión de calidad del software consisten en numerosas actividades. Algunos de ellos pueden encontrar defectos directamente, mientras otros indican donde pueden resultar valiosas más revisiones. Estos últimos también son conocidos como actividades de "direct-defect-finding". [37]

40

La planificación para la calidad de software implica: [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

1

Los procesos SQA proporcionan la garantía de que los productos software y los procesos en el ciclo de vida de proyecto son conformes a los requerimientos especificados por medio de la planificación, emitiendo, y realizando un conjunto de actividades para generar la confianza adecuada en que se está construyendo calidad dentro del software. Ello significa asegurar que el problema está clara y suficientemente identificado y que los requerimientos de la solución están correctamente definidos y expresados. El SQA procura mantener la calidad a lo largo de todo el desarrollo y mantenimiento del producto mediante la ejecución de una variedad de actividades en cada etapa que pueden permitir identificación temprana de problemas, un rasgo casi inevitable de cualquier actividad compleja.

[37]

2.2.2.1.6.2.2. Verificaciones y Validación

1

La verificación y validación del software es un acercamiento disciplinado a la evaluación de productos de software a lo largo de todo el ciclo de vida de producto. Un esfuerzo en verificaciones y validaciones es esforzarse en asegurar que la calidad es construida dentro del software y que el software satisface exigencias de usuario. [37]

1

Además trata directamente la calidad de producto software y emplea técnicas de prueba que pueden localizar defectos de tal manera que estos puedan ser tratados. También evalúa los productos intermedios, como, y, en esta capacidad, los pasos intermedios de los procesos de ciclo de vida de software. [37]

2.2.2.1.6.2.3. Revisiones y Auditorias

1

- Revisiones de gestión: El objetivo de una revisión de gestión es supervisar el progreso, determinando el estado de planes y programas, requerimientos confirmados y su sistema de localización, o evaluar la efectividad de los enfoques de gestión empleados para lograr la idoneidad del objetivo. [37]

1

- Revisiones técnicas: El propósito de una revisión técnica es evaluar el producto software para determinar si es idóneo para su correspondiente uso. El objetivo es identificar discrepancias con especificaciones aprobadas y estándares. El resultado debería proporcionar gestión con evidencias (o no) de que el producto converge con sus especificaciones y se adhiere a los estándares, y que los cambios están controlados. [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 el objetivo de formar a una audiencia en cuanto a un producto de software.
- Auditorias: El objetivo de una auditoría de software es proporcionar una evaluación independiente de la conformidad de productos software y procesos a regulaciones aplicables, 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 forman un sistema, convirtiéndose desde hace unos años en una disciplina que identifica la configuración en distintos puntos del tiempo con la finalidad de obtener control sobre el mismo, sin perder la integridad y rastreabilidad durante el ciclo de vida del software. [19]

También se puede considerar como una colección de versiones específicas de elementos de hardware, firmware o software que se combinan de acuerdo con un proceso de construcción específico para un satisfacer un propósito particular. Por tanto la gestión de configuración es la disciplina de identificar la configuración de un sistema en momentos diferentes con el propósito de controlar de una manera sistemática los cambios en la configuración y mantener la integridad y el seguimiento de los cambios en la configuración durante el ciclo de vida del sistema. [37]

1 La gestión de la configuración del software es un proceso que soporta el ciclo de vida del software (IEEE12207.0-96) que beneficia a la gestión de proyectos, las actividades de desarrollo y mantenimiento, las actividades de garantía y a los clientes y usuarios del producto final. [37]

1 2.2.2.1.7.1. Gestión del proceso de la gestión de la configuración del software

1 La gestión de la configuración del software controla la evolución e integridad de un producto identificando sus elementos, gestionando y controlando los cambios y verificando, guardando y produciendo informes de la información de configuración. Desde la perspectiva del ingeniero de software, la gestión de la configuración del software facilita las actividades del desarrollo e implementación de cambios. El éxito de una implementación de la gestión de la configuración del software requiere una planificación y gestión cuidadosas. [37]

12 2.2.2.1.7.2. Identificación de la Configuración del Software

La actividad de la identificación de la configuración del software identifica elementos que se han de controlar, establece métodos de identificación para los elementos y sus versiones y establece las herramientas y técnicas que se usarán para adquirir y gestionar los elementos controlados. Estas actividades proporcionan la base para las otras actividades de la gestión de la configuración del software. [37]

12 2.2.2.1.7.3. Control de la Configuración del Software

Al control de la configuración del software le concierne la gestión de cambios durante el ciclo de vida del software. Cubre los procesos que determinan los

cambios que se realizarán, la autoridad requerida para aprobar ciertos cambios, el soporte para la implementación de dichos cambios y el concepto de desviación formal de los requerimientos del proyecto, además de las cancelaciones de requerimientos. La información derivada de estas actividades es útil para medir el tráfico de cambios y ruptura y aspectos por rehacer. [37]

7

2.2.2.1.7.4. Registro del Estado de la Configuración del Software

La contabilidad del estado de la configuración del software es la actividad de registrar y proporcionar la información necesaria para una gestión efectiva de la configuración del software. [37]

7

- Información del Estado de la Configuración del Software: La actividad de la contabilidad del estado de la configuración del software diseña y opera un sistema para la captura y generación de los informes necesarios durante el ciclo de vida. Como en cualquier sistema de información, se debe identificar, recoger y mantener la información del estado de la configuración que se ha de gestionar según las configuraciones evolucionan. [37]

1

1

- Informes del Estado de la Configuración del Software: Los informes generados pueden ser usados por varios elementos de la organización o del proyecto, incluyendo el equipo de desarrollo, el equipo de mantenimiento, la gestión del proyecto y las actividades de calidad de software. Los informes pueden tener la forma de respuestas inmediatas a preguntas específicas o ser informes prediseñados producidos periódicamente. [37]

6

2.2.2.1.7.5. Auditoría de la Configuración del Software

La auditoría de software es una actividad que se realiza para evaluar independientemente la conformidad de productos de software y procesos con las regulaciones, estándares, guías, planes y procedimientos. Las auditorías se llevan a cabo de acuerdo con un proceso bien definido que consiste en varias responsabilidades y papeles de auditoría. En consecuencia, cada auditoría se debe planear con cuidado. Una auditoría requiere un número de personas que realizarán una variedad de tareas en un periodo de tiempo bastante reducido. Herramientas que den soporte a la planificación y ejecución de la auditoría pueden facilitar el proceso enormemente. [37]

1

La actividad de auditoría de la configuración del software determina el grado en que un elemento satisface las características funcionales y físicas. Se pueden realizar auditorías informales de este tipo en momentos clave del ciclo de vida. Hay dos tipos de auditorías que podrían ser requeridas por el contrato (por ejemplo, en contratos para software crítico): la Auditoría de la Configuración Funcional (FCA) y la Auditoría de la Configuración Física (PCA). [37]

6

2.2.2.1.7.6. Gestión del Lanzamiento y Distribución del Software

El término “lanzamiento” se usa en este contexto para referirse a la distribución un elemento de la configuración del software fuera de la actividad de desarrollo. Esto incluye tanto lanzamientos internos como la distribución a clientes. Cuando una versión diferente de un elemento de software está disponible para ser entregada,

como las versiones para diferentes plataformas o versiones con diferentes capacidades, es normalmente necesario preparar una versión específica y empaquetar los materiales adecuados para distribuirla. La biblioteca de software es un elemento clave para realizar las tareas de lanzamiento y distribución. [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]

1

2.2.2.1.8.1. Iniciación y Alcance

El enfoque de este conjunto de actividades se centra en la determinación eficaz de los requisitos del software por medio de varios métodos de inducción y la valoración de la viabilidad del proyecto desde distintos puntos de vista. Una vez que se ha establecido la viabilidad, la tarea pendiente dentro de este proceso es la especificación de la validación de requisitos y del cambio de procedimientos. [37]

1

2.2.2.1.8.2. Planificación de un Proyecto de Software

El proceso de planificación iterativa está regulado por el alcance y requisitos, y por el establecimiento de la viabilidad. A estas alturas, se evalúan los procesos del ciclo de vida del software y se selecciona el más apropiado (considerando la naturaleza del proyecto, su grado de novedad, su complejidad funcional y técnica, sus requisitos de calidad, etc.). Si la situación lo aconseja, se planea entonces el propio proyecto en la forma de una descomposición jerárquica de tareas, se especifican y caracterizan los entregables asociados de cada tarea en términos de

calidad y de otros atributos en la línea de los requisitos declarados, y se emprende la descripción detallada del esfuerzo de realización, el calendario y la estimación de costes. Más adelante se asignan los recursos a las tareas para optimizar la productividad del personal (a nivel de individuo, de equipo y organizacional), el uso de equipos y materiales, y la adhesión a los horarios. Se emprende una gestión de riesgos detallada y se discute el “perfil de riesgo” entre todos los contratistas de relieve, llegando a un acuerdo. Se determinan los procesos comprehensivos de gestión de la calidad del software como parte del proceso en términos de procedimientos y responsabilidades para asegurar la calidad del software, la verificación y la validación, las revisiones y las auditorías. [37]

1

2.2.2.1.8.3. Promulgación del Proyecto de Software

Aquí se ejecutan los planes y se promulgan los procesos incluidos en los planes. A lo largo de este proceso todo se centra en la adhesión a los planes, con una expectativa arrolladora de que tal adhesión llevará a la satisfacción plena de los requisitos del contratista y al logro de los objetivos del proyecto. Las actividades actuales de gestión para medir, supervisar, controlar e informar son fundamentales para la promulgación. [37]

1

2.2.2.1.8.4. Revisión y Evaluación

En puntos críticos del proyecto, se evalúan el progreso global hacia el logro de los objetivos prefijados y la satisfacción de los requisitos del contratista. De igual modo, en hitos particulares se llevan a cabo valoraciones sobre la efectividad del proceso global hasta la fecha, del personal involucrado, y de las herramientas y métodos utilizados. [37]

2.2.2.1.8.5. Cierre

El proyecto llega a su fin cuando todos los planes y procesos implicados se han promulgado y completado. En esta fase, se repasan los criterios para el éxito del proyecto. Una vez que se ha establecido el cierre, se llevan a cabo actividades de archivado, post mortem y de mejoras de los procesos. [37]

2.2.2.1.8.6. Medidas de la Ingeniería del Software

- 1 • Establecer y Sostener el Compromiso de Medir: Aceptar los requisitos de medición. Cada tentativa de medición debe estar guiada por objetivos organizacionales, e impulsada por un conjunto de requisitos de medición establecidos por la organización y por el proyecto. [37]
- 1 • Planificar el Proceso de Medición: identificar las necesidades de información. Las necesidades de información se basan en las metas, restricciones, riesgos y problemas de la unidad organizacional. Seleccionar las mediciones. Se deben elegir las mediciones candidatas al puesto, claramente vinculadas a las necesidades de información. Las mediciones deben seleccionarse en base a las prioridades de las necesidades de información y otros criterios como el coste de recolección de datos, el grado de trastorno del proceso durante la recolección, la facilidad de análisis, la facilidad de obtener datos precisos y consistentes, etc. [37]
- 1 • Realizar el Proceso de Medición: Integrar los procedimientos de medición con los procesos pertinentes. Los procedimientos de medición, tales como la recolección de datos, deben integrarse en los procesos que están midiendo. Esto puede que implique cambiar los procesos actuales para

adaptar la recolección de datos o las actividades de generación. Puede también implicar el análisis de los actuales procesos para minimizar esfuerzos adicionales y evaluaciones del efecto en los empleados, con el fin de asegurarse de que serán aceptados los procedimientos de medición. Se necesita considerar los temas morales y otros factores humanos. Además, los procedimientos de medición deben comunicarse a los proveedores de datos, puede que se tenga que proporcionar entrenamiento, y se debe proporcionar el típico apoyo. [37]

1

- **Evaluar las Mediciones:** Evaluar los productos de información. Evaluar los productos de información contrastándolos con los criterios de evaluación específicos y determinar las fuerzas y debilidades de los productos de información. Esto puede realizarlo un proceso interno o una auditoría externa y debe incluir una retroalimentación de los usuarios de las mediciones. [37]

2.2.2.1.9. Proceso de ingeniera 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]

1

2.2.2.1.9.1. Proceso de Implementación y Cambios

Esta subárea se centra en los cambios organizacionales. Describe la infraestructura, actividades, modelos y consideraciones prácticas de un proceso de implementación y cambios. [37]

1

Aquí se describe la situación en la que los procesos se despliegan por primera vez (por ejemplo, introduciendo un proceso de inspección en un proyecto o un método que cubra todo el ciclo de vida), y donde se cambian los procesos actuales (por ejemplo, introduciendo una herramienta u optimizando un procedimiento). [37]

1

2.2.2.1.9.2. Definición de Procesos

Una definición de un proceso puede ser un procedimiento, una política, o un estándar. Los procesos de ciclo de vida del software se definen por muchas razones, que incluiría el incrementar la calidad del producto, el facilitar el entendimiento y la comunicación humana, apoyar las mejoras de los procesos, apoyar la gestión de los procesos, suministrar una guía automatizada para los procesos, y suministrar un apoyo para ejecuciones automatizadas. Los tipos de definiciones de procesos requeridos dependerán, al menos parcialmente, de las razones para la definición. [37]

1

2.2.2.1.9.3. Valoración del Proceso

La valoración del proceso se lleva a cabo utilizando tanto un modelo de valoración como un método de valoración. En algunas instancias, el término "apreciación" se utiliza en vez de valoración, y el término "evaluación de la capacidad" se utiliza cuando la apreciación tiene como propósito la adjudicación de un contrato. [37]

2.2.2.1.9.4. Medición de los Procesos y Productos

5

La medición puede realizarse para apoyar la iniciación de unos procesos de implementación y cambio o para evaluar las consecuencias de un proceso de implementación y cambio, o puede realizarse en el producto mismo. [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

Los cinco primeros asuntos del subárea de los Instrumentos de Ingeniería de Software corresponden a las cinco primeras áreas del conocimiento de la Guía (Exigencias de Software, el Diseño de Software, la Construcción de Software, Pruebas de Software, y el Mantenimiento de Software). Los cuatro siguientes asuntos corresponden a las áreas de conocimiento restantes (la Dirección de Configuración de Software, la Dirección de la Ingeniería de Software, el Proceso de Ingeniería de Software, y la Calidad de Software). Proporcionan un asunto adicional, dirigiendo áreas como las técnicas de integración de instrumento que son potencialmente aplicables a todas las clases de instrumentos. [37]

- **Las herramientas de Exigencias de Software:** Los instrumentos para tratar con exigencias de software han sido clasificados en dos categorías: modelado e instrumentos de capacidad de rastreo. [37]
- **Las herramientas Diseño de Software:** Este asunto cubre instrumentos para crear y comprobar diseños de software. Hay una variedad de tales instrumentos, con la mayor parte de esta variedad siendo una consecuencia de la diversidad de notaciones de diseño de software y métodos. A pesar de esta variedad, ninguna división convincente para este asunto ha sido encontrada. [37]
- **Las Herramientas de Construcción de Software:** Este asunto cubre instrumentos de construcción de software. Estos instrumentos son usados para producir y traducir la representación de programa (por ejemplo, el

código original) que suficientemente es detallado y explícito para permitir la ejecución de máquina. [37]

5

- Herramientas de Pruebas de Software: Generadores de pruebas, marcos de ejecución de prueba, herramientas de evaluación de prueba, herramientas de dirección de prueba, herramientas de análisis de Funcionamiento. [37]
- Herramientas de Mantenimiento de Software: herramientas de comprensión y herramientas de reingeniería. [37]

1

2.2.2.1.10.2. Los Métodos de la Ingeniería de Software:

Los Métodos de la Ingeniería de Software están dividido en tres temas: métodos heurísticos que tratan con accesos informales, métodos formales que tratan con accesos matemáticamente basados, y métodos de prototipado que tratan con software que trama accesos basados en varias formas de prototipado.

1

- Métodos heurísticos: contiene Métodos Estructurados, Métodos Orientados a datos y Métodos Orientados a objetos.
- Métodos Formales: Esta subdivisión trata con el software matemáticamente basado métodos de la ingeniería, y es subdividida según varios aspectos de métodos formales como la Especificación del lenguaje y notaciones, el Refinamiento y las Propiedades de Verificación/confirmación.
- Métodos de prototipado: Esta subdivisión cubre métodos que implican el prototipado de software y es subdividida en estilos de prototipado, objetivos, y técnicas de evaluación.

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]

84

Hoy en día es muy 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]

66

2.2.2.2.1.2. Diccionario de datos

El diccionario de datos se conforma por un listado organizado de todos aquellos datos 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

41

El modelo de datos es un lenguaje de programación orientado a describir una base de datos donde encontramos: [20]

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

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]

56

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)

4

Este paradigma se define como un marco de trabajo de arquitecturas para desarrollo de software con tres objetivos principales: portabilidad, interoperabilidad

4

y reusabilidad. A partir de estos, se dice que posee una gran habilidad para contemplar el ciclo completo de 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 rigurosos para garantizar que los estándares de implementaciones tecnológicas tengan las mismas reglas del negocio.
- Independencia de plataforma: reduciendo el tiempo, costo y 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 enfoca en el contexto y requisitos de una aplicación, sin preocupación de su estructura o procesamiento.
 - Independiente de la plataforma (PIM): Hace énfasis en aquellas capacidades operacionales 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).** MOF es un lenguaje para la especificación de meta-modelos, donde se consideran 4 capas: Un modelo abstracto de los objetos y sus relaciones; Un conjunto de reglas para mapear un meta modelo a interfaces independientes del lenguaje; Reglas definiendo el ciclo de vida, composición y semántica de los modelos basados en MOF; y una jerarquía de interfaces reflexivas definiendo operaciones genéricas para descubrir y manipular las propiedades de los meta-modelos basados en MOF [8].
- **Common Warehouse Metamodel (CWM).** Especifica interfaces que pueden ser usadas para habilitar el intercambio de metadatos de almacenes de datos e inteligencia de negocio, entre distintas herramientas, plataformas y metadatos [8].

- 4
- Unified Modeling Language (UML). El Lenguaje de Unificado (UML) sirve como notación base para la definición de CMW [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 cierto nivel de abstracción, con mapeos de un conjunto de modelos hacia otro que incluyen: [8]

- Un CIM donde se contiene todas aquellas reglas de negocio definidas.
- Un PIM que defina el modelo conceptual completo con todas sus relaciones.
- Varios PSM para crear componentes de ejecución y pruebas de los sistemas.

La idea básica 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 con evolutivo, iniciando con la definición de sus objetivos globales 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, significa que la prueba falló y se debe hacer correcciones del error que se tenga hasta que el usuario quede satisfecho. ^[3] [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]

- 3
• 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.
- 3
• Es muy útil cuando 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]

3

- **Modelo de Prototipos rápido:** Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y prescinde del prototipo cuando el próximo diseño es desarrollado mediante un nuevo prototipo.
- **Modelo de Prototipos reutilizable:** También conocido como "Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real. Mayormente es utilizado en el desarrollo de software, si bien determinados productos de hardware pueden hacer uso del prototipo como la base del diseño de moldes en la fabricación con plásticos o en el diseño de carrocerías de automóviles.
- **Modelo de Prototipos Modular:** También conocido como Prototipado Incremental (Incremental prototyping); se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresá.
- **Modelo de Prototipos Horizontal:** El prototipo cubre un amplio número de aspectos y funciones pero la mayoría no son operativas. Resulta muy útil para evaluar el alcance del producto, pero no su uso real.
- **Modelo de Prototipos Vertical:** El prototipo cubre sólo un pequeño número de funciones operativas. Resulta muy útil para evaluar el uso real sobre una pequeña parte del producto.
- **Modelo de Prototipos de Baja-fidelidad:** El prototipo se implementa con papel y lápiz, emulando la función del producto real sin mostrar el aspecto real del mismo. Resulta muy útil para realizar tests baratos.

- **Modelo de Prototipos de Alta-fidelidad:** El prototipo se implementa de la forma más cercana posible al diseño real en términos 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 ¹¹ enfoque clásico para el ciclo de vida del desarrollo de sistemas, donde se cree que es un método puro que implica un desarrollo rígido, siguiendo secuencia de ¹¹ actividades consistentes para el análisis 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 las necesidades del producto y por lo general su parte final es texto.
- El diseño describe la estructura del producto de manera interna y suele ser representado con diagramas de flujo o texto.
- La implementación por su parte hace referencia a programación. Como ¹¹ producto final es el código en cualquier nivel incluido el que se genera por sistemas de compilación automática.
- La integración es el proceso en el cual se ensambla las partes para completar un producto como fase final.

También se caracteriza fuertemente por ordenar de una manera inflexible las ¹¹ 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]

11

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

9

Se precisa un Caso de Uso como un pedazo de funcionalidad de un determinado sistema que suministra al usuario un valor agregado, es decir, representan los requisitos funcionales de dicho sistema. En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba mediante el proceso centrado en la arquitectura [11].

9

La arquitectura de un sistema es la organización o estructura de sus piezas más importantes, lo que permite tener una visión común entre todos los involucrados

(desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [11].

9

RUP se centra en utilizar aquellos casos de uso con el fin de guiar el proceso que se presta en especial atención del establecimiento de una buena arquitectura que no se vea fuertemente impactada ante posteriores cambios dentro de la misma construcción y el futuro mantenimiento del desarrollo. [11]

9

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]

100

59

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]

18

Una iteración se puede ejecutar por medio de una cascada, donde básicamente pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y pruebas). Adicionalmente existe una planificación de la iteración, un posterior análisis y otras actividades adicionales de la iteración en específico. Para finalizar se hace una integración 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]

- 9 • Las fases de inicio se enfocan a la comprensión del problema y la tecnología, así como la delimitación del ámbito 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.
- 9 • En la fase de construcción, se lleva a cabo 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.

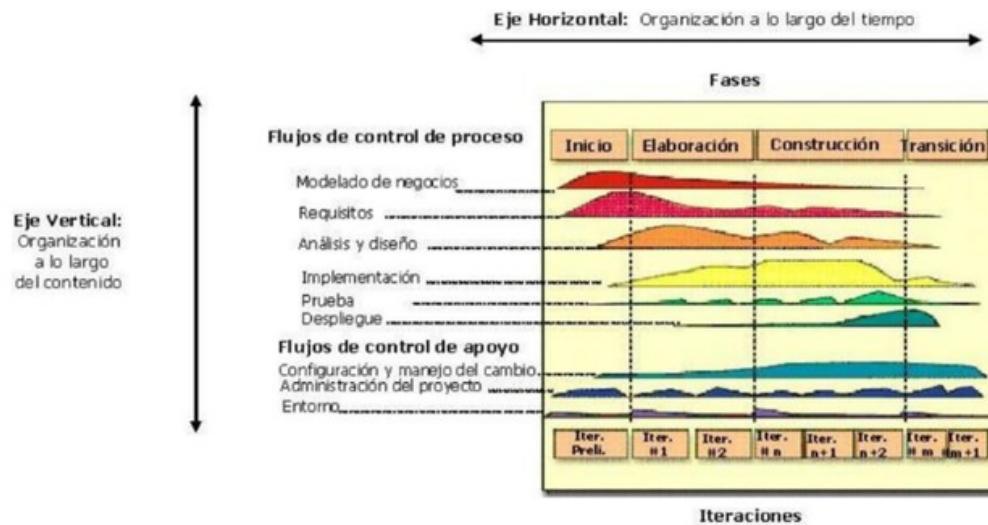
Dentro de la estructura del proceso de RUP, encontramos una serie de pasos que se pueden considerar por medio de 2 ejes: [11]

- 9 • Horizontal: Representa el tiempo y es usado para poner aspectos dinámicos dentro del proceso, indicando las características del ciclo de vida que lo compone, siendo expresado en término de fases. Inicio, Elaboración, Construcción y transición.

18

- Vertical: Representa los aspectos del proceso que son estáticos. Describe el proceso en términos de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

Ilustración 1



[22]

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 permite 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 prácticas” 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 incremento de producto final que sea susceptible de ser entregado 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

80

La metodología XP por sus siglas (eXtreme Programming) donde se sigue un conjunto 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 está 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 programar 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: Guía a todos los miembros 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 dueño de la organización y el vínculo 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 47 con las herramientas, tecnologías y prácticas que se utilizaran en el transcurso del proceso de creación del software. [35]
- **Fase de planeamiento:** Los desarrolladores consideraran 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 anotaran 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:** Acá el cliente ya no posee más historias de usuario para incorporarlas al desarrollo, por ende las necesidades del cliente han sido satisfechas en aspectos como 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

8 El control de versiones es una gestión de los múltiples cambios que suelen realizarse en los elementos de algún producto o la configuración interna de dicho elemento. Siendo más específico, una revisión, una versión, o edición del producto será el estado en el que se encuentre en el momento del desarrollo o modificación. [25]

Aunque un sistema de control de versiones se puede hacer de forma muy manual, 8 se prefiere que se usen herramientas 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]

49

- Un mecanismo de almacenamiento de los elementos que deba gestionar
- La posibilidad de hacer cambios sobre elementos existentes o ya 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.

8

Por consiguiente, una revisión se define como una versión determinada de información que se gestiona o procesa con el fin de exponerla como producto final o iteración y avance a un producto específico. Para que se pueda marcar 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]

8

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]

8

Un término importante es la integración o fusión, donde se une dos conjuntos de cambios sobre un mismo fichero de una revisión, lo que puede suceder cuando un usuario ⁸ actualiza su copia local con los cambios realizados y añadidos al repositorio por otros usuarios. Sigue 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 solucionaran 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 todos los elementos unitarios que componen un proceso. Acá 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 niveles más bajos y luego realiza integración de todos los módulos hacia arriba.

7

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 deseé.

- **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]

46

- **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 máquina 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 gráficas 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

2.2.2.4.4. 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]

Esta información es de gran volumen de tamaño: estamos hablando de veinte o treinta datos multiplicados por miles o millones de personas. La antigua gestión de datos se basaba en archivos informáticos, pero para las necesidades de hoy en día hacen falta sistemas más perfeccionados que son precisamente lo que se denomina sistema de base de datos. Llegamos a la conclusión de que necesitaríamos una base de datos para automatizar el acceso a la información y poder acceder a ella de manera rápida y fácil además de poder realizar cambios de una manera más eficiente [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 7
- 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

96

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
- Autoincrem  ntales: Campos num  ricos que incrementan con cada registro que agregan a la base

Modelo Entidad- Relaci  n

83

Este diagrama se encarga de modelar la base de datos de un sistema de informaci  n, con el fin de tener un mapa y organizar de una manera jer  quica y detallada todos sus componentes. Este modelo permite representar cualquier tipo de abstracci  n, percepci  n y conocimiento en un sistema de informaci  n formado por un conjunto de objetos que se determinan como entidades y relaciones, incorporando adem  s una vista visual del diagrama en construcci  n. [12]

42

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 informaci  n. 95
- Atributos: Son esas caracter  sticas o rasgos que poseen una entidad
- Relaci  n: Vinculo que define una dependencia entre dos o m  s entidades
- Interrelaci  n: Constituyen los v  nculos entre entidades, pero de forma tal que representa las relaciones de forma efectiva, a trav  s 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 sucesor de Windows Mobile, enfocándose en el mercado de consumo personal y no empresarial. Su interfaz de usuario es totalmente renovada e innovadora que integra varios servicios como OneDrive, Skype y Xbox Live con el fin de competir directamente con Android de Google e iOS de Apple. [30]

Ilustración 2



Recuperado de <http://www.taringa.net/comunidades/winphone/7240486/Aporte-Juegos-Windows-Phone-7-Act-30-05-2014.html>

Windows Phone fue lanzado 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

30

La pantalla de inicio se compone de Live Tiles, mosaicos dinámicos que en realidad son enlaces a aplicaciones u objetos individuales como puede ser contactos, páginas web o archivos multimedia. Estos mosaicos se actualizan permanentemente, manteniendo informado al usuario con las llamadas, mensajes, correos electrónicos, citas, juegos o nuevas aplicaciones. [30]

14

Ilustración 3



Sacado de: En línea: [<http://antiviruspara.net/wp-content/uploads/2015/07/Windows-Phone-82.jpg>]

14

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]

Ilustración 4



Sacado de: 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]

14

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

La tienda de Windows phone es una plataforma web dedicada a que los usuarios descarguen aplicaciones, juegos, música y videos. Esto equivale al Play Store de Android o al 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]

Ilustración 5



Sacado de: En línea: [<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 execute 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]

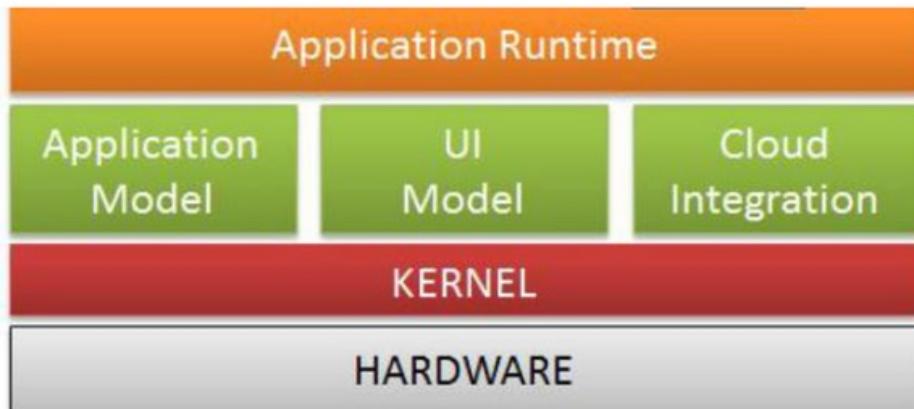
14

- 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 microUSB 2.0
- Cámara trasera obligatoria, con autoenfoque(opcional) y mínimo VGA (flash opcional)
 - 8
- WiFi 802.11b/g y Bluetooth, acelerómetro, sensores de proximidad y luz, motor de vibración
 - 14
- 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: Acá las aplicaciones se ejecutan bajo un fichero con extensión .xap, el cual contiene y establece los recursos válidos para que la aplicación se ejecute.
 - Modelo UI: se refiere a la interfaz de usuario, el cual se compone de elementos y una sesión de interacciones que realiza un usuario sobre la aplicación o incluso implicar otras aplicaciones.
 - Integración con la nube: En este punto se orienta integración 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]

Ilustración 6



Sacado de: En linea:

79

[http://dataoteca.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 ⁷ en el caso de los dispositivos móviles, la cual esta 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]

Tabla 1

| Versión | Descripción |
|------------------------------|--|
| 43 | |
| Windows Phone 7.0 ("Photon") | Versión Inicial ² |
| 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 ² |
| Windows Phone 7.5 ("Mango") | Empezó a funcionar en el 2011 donde aporto más de 500 funcionalidades como multitarea, procedimiento de búsqueda, integración con redes sociales, aplicaciones como office y |

| | |
|---------------------------------|---|
| | Skydrive, Aumento características del 2 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. 87 |
| Windows Phone 7.8 | Se estrenó en enero de 2013 donde se ofrece 2 nuevas ventajas como la pantalla de usuario y fondos personalizados para la imagen de bloqueo. A partir de acá se centró en WP8 para generar nueva versión. |

Comparación de las diferentes versiones de S.O. de Windows Phone dentro del marco 7.0

Windows Phone 8:

Es la segunda generación de los sistemas operativos Windows Phone liberada en
8 el 2012, donde sustituye la arquitectura basada en Windows CE con uno basado
8 de Kernel de Windows NT, permitiéndole a los desarrolladores aportar
aplicaciones fácilmente entre las 2 plataformas. WP8 soporta pantallas con mejor
8 resolución y procesadores multi-núcleo, NFC (que principalmente se puede utilizar
2 para compartir contenidos y realizar pagos), Control infantil: un área disponible y

controlado para los niños, independiente del resto del móvil, Data Sense medidor de consumo de recursos del dispositivo móvil, cartera para guardar tus tarjetas de fidelización y crédito, Nuevo navegador Internet Explorer 10, actualizaciones en el dispositivo móvil (sin usar el ordenador): [6]

Tabla 2

| Versión | Descripción |
|--|--|
| Windows Phone 8 Update 1 (GDR1) ('Pórtico') | Primera versión donde permite conservar la Wi-Fi 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. |
|--|---|

Comparación de las diferentes versiones de S.O. de Windows Phone dentro de la nueva versión 8.0

58

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]

8

- 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]

27

adquisición, de suministro, de desarrollo, de operación [36]

29

gestión de la configuración, de aseguramiento de la calidad, de verificación,

de validación, de revisión conjunta, de auditoria y de solución de problemas.

[36]

- 27
- 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 evolucionó 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]

61

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]

31

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]:

Tabla 3

| NOMBRE | DESCRIPCION | PLATAFORMA | URL DE DESCARGA |
|---------------|---|---|---|
| Beetag Reader | Permite leer códigos bidimensionales QR, datamatrix y un código bidimensional creado por la propia empresa BeetagCode, adicionalmente dispone | 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.beetagg.com/en/download-qr-reader/ |

7

| | | | |
|-----------------------|--|--|---|
| | de soporte para leer códigos de barras unidireccionales del tipo EAN-13/UPC-A. | | |
| UpCode Reader | El lector funciona muy bien con cámaras con autofocus e interacciona perfectamente con el terminal y sobre todo interpretando QR con Vcards. Upcode lee además códigos datamatrix y códigos unidireccionales, características que le confieren un gran versatilidad. | UpCode ofrece un lector de QR perfecto para terminales Nokia con Symbian | <ul style="list-style-type: none"> • http://www.upc.fi/en/upcode/download/ |
| I.nigma Reader | Tiene soporte para una amplia gama de terminales y plataformas. Este lector permite compartir el | Android – iOS – Symbian – Blackberry – Windows phone | <ul style="list-style-type: none"> • http://www.inigma.com/DownloadInigmaReader.html |

| | | | |
|-------------------------|--|--|---|
| | 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. | | |
| 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, | 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, | <ul style="list-style-type: none"> • http://www.quickmark.com.tw/En/basic/downloadMain.asp |

| | | | |
|----------------------|---|---|---|
| | navegar por los mapas, etc. | SoftBank, Sony Ericsson, Sprint, Swisscom, T-Mobile, UBiQUiO, UTStarcom y Vodafone. | |
| Kaywa Reader | Lector de códigos bidimensionales | Symbian, Java como Motorola, Samsung o Sony Ericson. 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.lynee.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. | • 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. | • http://www.neoreader.com/get-neoreader/wap-download |

Comparación de los múltiples lectores de códigos 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 smartphones 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]

10

En la tesis de Juliana Gaviria García, titulada "Nuevas tecnologías, nuevas formas de negocios: Smartphones 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 tipo. "[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

2

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 Vilchez en su artículo “QR Code en museos”: “²⁵ 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 geolocalización 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 ⁴⁵cartelería, 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 ⁴⁵localización 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 una 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.

42
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.

94

3.2. DISEÑO DE INVESTIGACIÓN

En esta investigación se intentara abordar los múltiples conocimientos de manera teórico práctica 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”.

93

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:

Ilustración 7



- **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 del 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 la 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 contará 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 realizará 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 uso la biblioteca externa Zxing la cual permite procesar imágenes en multi formato 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 integrar 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 tomará como base ² la metodología de programación extrema o XP por sus siglas en inglés (eXtreme Programming).

La programación extrema permitirá desarrollar el proyecto poniendo ⁸⁶ 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 ⁷⁸ 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:

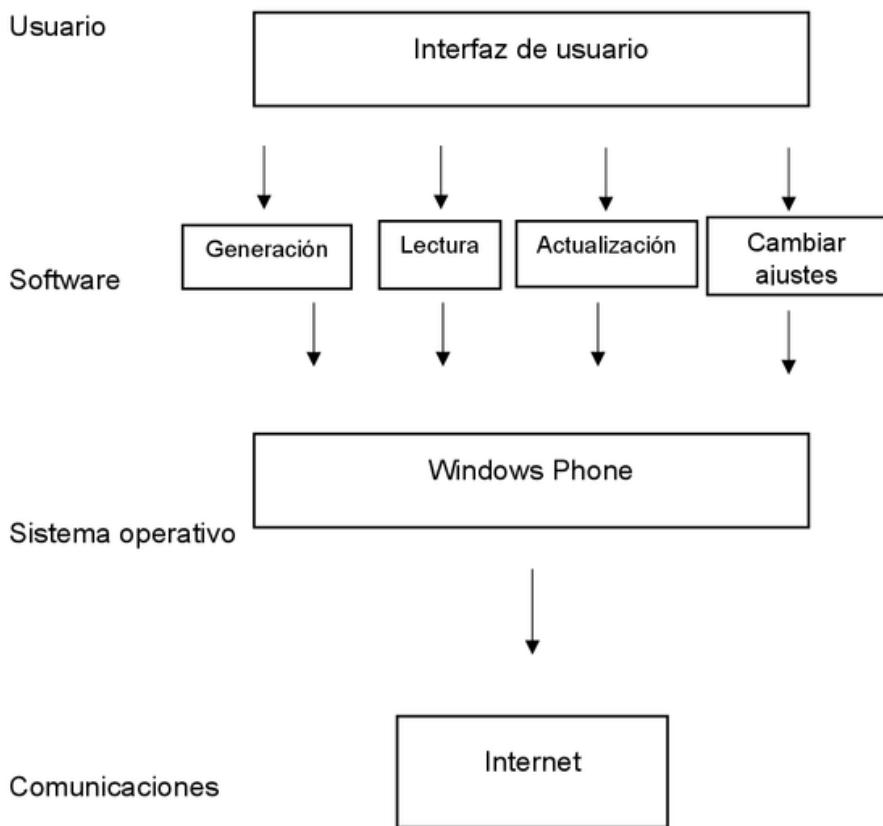
- Programador: Producir el código del sistema.
- Cliente: Expresa la idea para así mismo la implementación de lo que pidió en estancia final.
- Tester: Ejecutara pruebas y difunde los resultados en el equipo, además de ser responsable de herramientas de soporte.
- Consultor: ayuda al equipo a resolver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

47

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:

Ilustración 8



- 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:

92

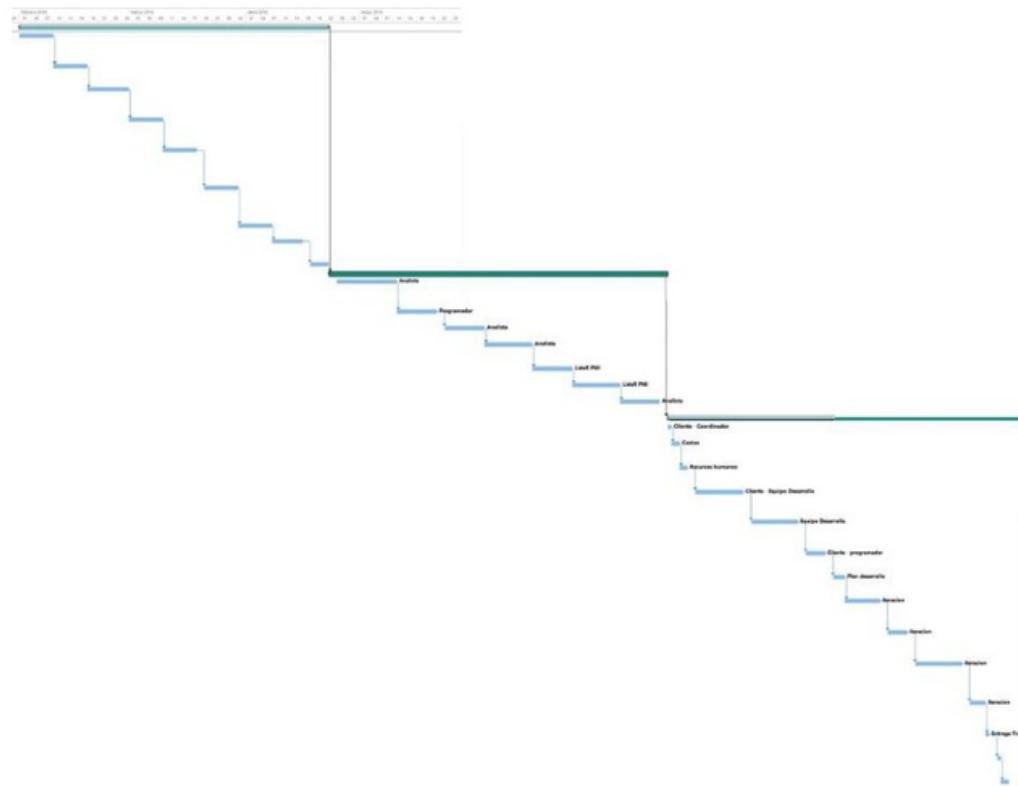
- Version de **Visual Studio**: Express 2012 for Windows phone.
- Complementos: **Windows** Development Tools (WDT).
- **Windows** SDK: 8.0
- Framework: 3.0 o superior
- Librería ZXing para leer códigos QR

3.2.3. Cronograma

Ilustración 9

| Modo de | Nombre de tarea | Duración | Comienzo | Fin | Predesores | 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 | jun 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 | jun 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 | jun 23/03/16 | 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 | jun 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 | jun 25/04/16 | mar 10/05/16 | | |
| 14 | Metodología de desarrollo a usar | 8 días | jun 23/05/16 | vie 01/06/16 | 13 | |
| 15 | Selección de arquitectura de software | 8 días | jue 02/06/16 | mié 11/06/16 | 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 | jun 18/07/16 | mié 12/10/16 | 18 | |
| 20 | Fase Inicio: Reunión de inicio | 1 día | jun 18/07/16 | jue 18/07/16 | | |
| 21 | Fase Ejecución: 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 | jun 25/07/16 | vie 05/08/16 | 22 | |
| 24 | Fase Planificación: Realización de estimaciones | 10 días | jul 06/08/16 | vie 19/08/16 | 23 | |
| 25 | Fase Ejecución: Historias de Usuario | 5 días | jul 22/08/16 | vie 26/08/16 | 24 | |
| 26 | Fase Ejecución: Planificación de recursos | 3 días | jul 25/08/16 | 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 | 3 días | jul 12/09/16 | vie 16/09/16 | 27 | |
| 29 | Fase Ejecución - Iteración 1: Codificar en C# para Windows Phone | 10 días | jul 13/09/16 | vie 30/09/16 | 28 | |
| 30 | Fase Ejecución - Iteración 2: Pruebas funcionales | 4 días | jul 01/10/16 | jue 06/10/16 | 29 | |
| 31 | Fase Ejecución: Reunión muerte del proyecto | 1 día | vie 07/10/16 | vie 07/10/16 | 30 | |
| 32 | Fase Cierre: Aceptación del proyecto | 1 día | jul 10/10/16 | jul 10/10/16 | 31 | |
| 33 | Fase Cierre: Entrega de aplicaciones y documentos finales | 2 días | mar 11/10/16 | mié 12/10/16 | 32 | |
| 34 | + Fase Cierre | 5 días | jul 17/10/16 | vie 21/10/16 | | |
| 35 | Reunión inicial de entrega | 1 día | jul 17/10/16 | jul 17/10/16 | | |
| 36 | Revisões históricas 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 | |

Ilustración 10



3.2.4. Instrumentos de Investigación

Las técnicas de recolección de datos se basan en realizar la estructuración de 91 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_D2Hd83RTgilYeQIQXjfVSnc3XQ/viewform?c=0&w=1 | |
| 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 Nemocon Colombia, más exactamente en la mina de sal de este municipio y toda la información que usted nos brinda será de carácter estrictamente confidencial. | | |
| 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 | ¿Cuántas veces ha visitado usted un lector de códigos QR en los últimos 30 días? | A. 2-3 B. 4-5 C. 6 ó más veces. D. Es la primera vez que visito la Aplicación |

| | | |
|----|--|---|
| B2 | ¿Cree que volverá a usar la aplicación en los próximos 30 días? | A. Sí B. No C. No estoy seguro |
| B3 | ¿Por qué motivo/s ha visitado usted la aplicación? | A. Para obtener mas 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 | ¿Como le gustaría combinar la tecnología con un museo? | |
| B6 | ¿Cree que la tecnología aumenta la visita en los museos? | |

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 Tabla 4

| Componente | Mes 1 | Mes 2 | Mes 3 | Cantidad | Costo | Costo |
|---|--------|---------------------|--------|----------|---------------------|----------------|
| | | | | | Unitario | 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 | 10KWH | 10KWH | 10KWH | 30KWH | 1000 | 30000 |
| Internet | 1 PLAN | 1 PLAN | 1 PLAN | 3 | 50000 | 150000 |
| | | | | | SubTotal | 3110000 |
| | | | | | Otros gastos | 200000 |
| | | | | | Total | 3310000 |

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 uso 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 (eXtreme 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 demoro 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):

Tabla 5

| 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 | | | | | |

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:

Tabla 6

| Descripción | Historias/Tareas | Resultado | Fecha de entrega |
|-------------|------------------|------------|------------------|
| Historias | H1 | Finalizada | 30/09/2016 |

| | | | |
|--------------------------|---|------------|------------|
| realizadas | | | |
| 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 de iteración 1

Requerimientos

Funcionales:

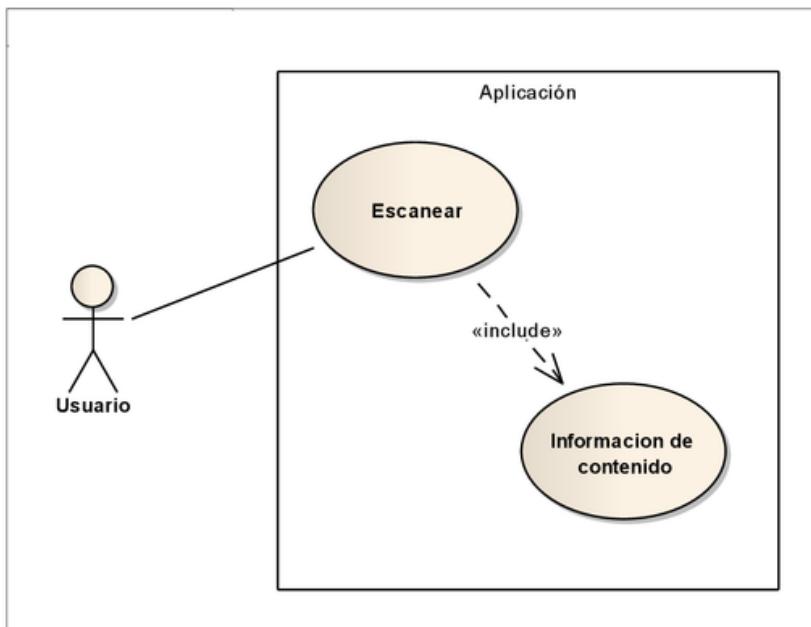
- **RF1.** El usuario Ingresa en la aplicación.
- **RF2.** El usuario Escanea el código.
- **RF3.** El usuario consulta la información obtenida del código.

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 WiFi.

Diagrama de Interacción

Ilustración 11



Documentación:

- a. 001 – Escanear

Tabla 7

| | |
|------------------------|--|
| 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 | <p>Usuario</p> <p>1 Ingresá a la aplicación.</p> <p>3 Escogé la opción de escanear un código nuevo</p> | <p>Aplicación</p> <p>2 Permite ingreso del usuario y muestra las opciones a realizar</p> <p>4 Valida opción.</p> <p>5 Escanea código en el medio o museo.</p> |
| Flujo Alternativo | <p>Usuario</p> <p>1</p> <p>3</p> <p>6 Vuelve a la elección de opciones</p> | <p>Aplicación</p> <p>2</p> <p>4 Valida opción.</p> <p>5 Error en la cámara.</p> |
| Post Condiciones | Mira información del código o consultar nuevo código otra vez. | |

Documentación

b. 002 – Obtener información

Tabla 8

| | | |
|--------------------------|--|---|
| 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 Ingresá a la aplicación. 3 Escogé 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 | Aplicación 2 4 Valida opción. 5 Error en la cámara. |

| | | |
|-----------------------------|--|---|
| | 6 Vuelve a la elección de opciones | 7. Error en la recuperación de información. |
| Post Condiciones | Mira información del código o consultar nuevo código otra vez. | |

Documentación

Diagrama de actividad por carril de Escanear Código QR

Ilustración 12

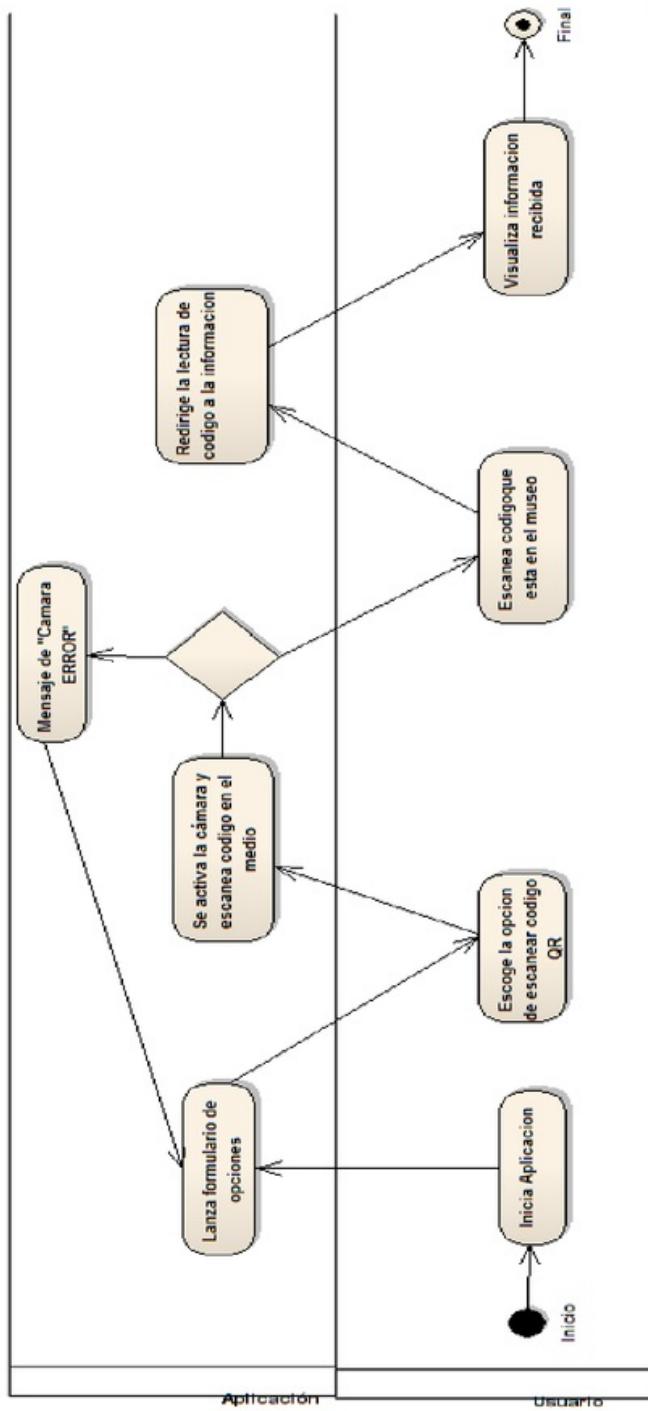


Diagrama de estado de Escanear Código

Ilustración 13

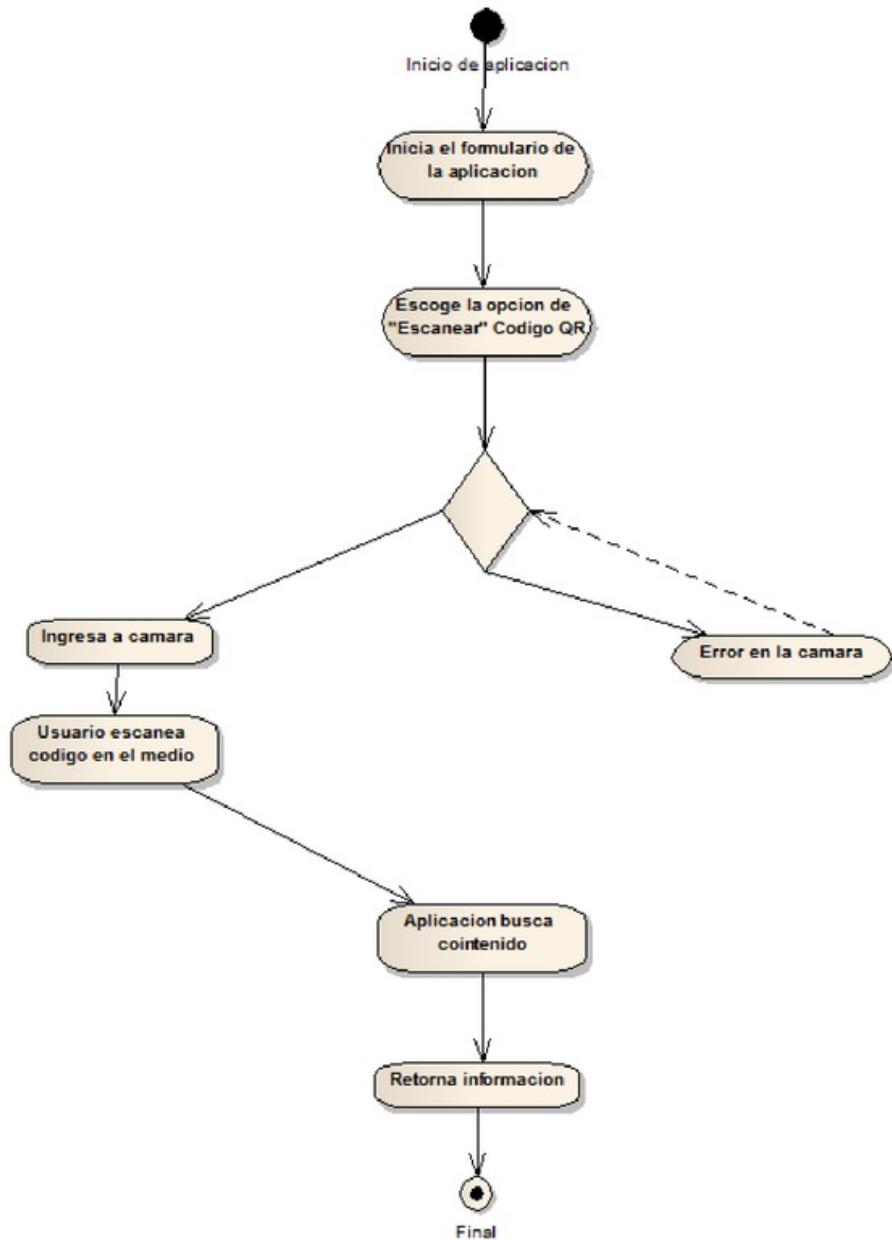
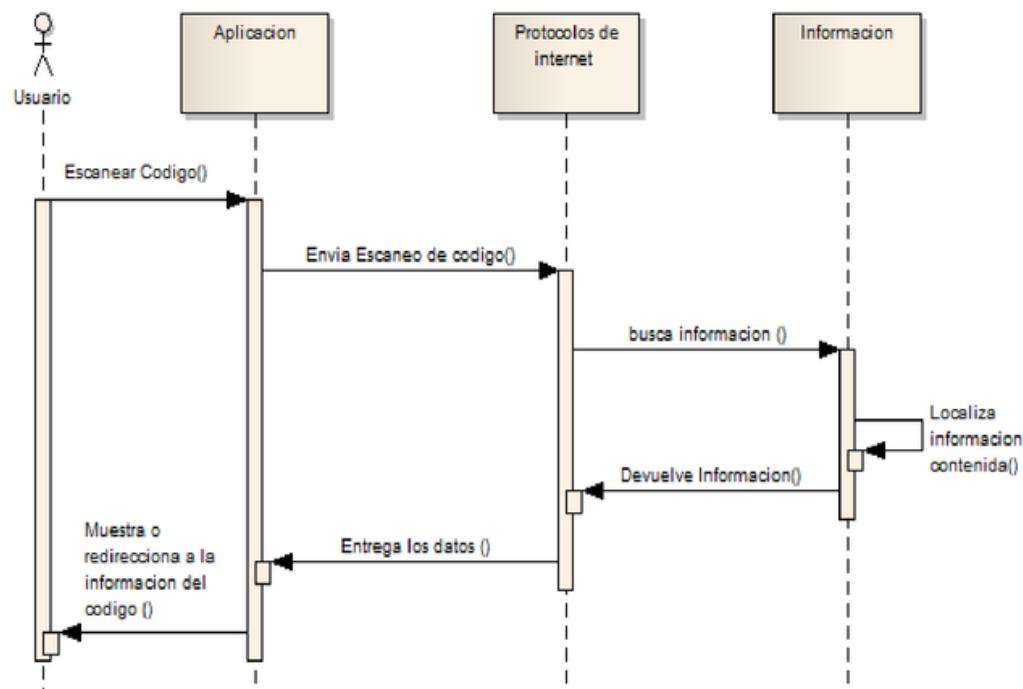


Diagrama de secuencia de Escáner código

Ilustración 14



Iteración 2:

Tabla 9

| 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 de iteración 2

Clase PhotoCameraLuminanceSource

```
52
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;
    }
}

```

19

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">
    16
    <!--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:

90

```
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.

-->

24

```
<!--TitlePanel contiene el nombre de la aplicación y el título de la  
pagina-->  
    <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.  
  
A22s 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">
        16
        <!--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"/>
                55
                <!--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="ExtraB 26 "></ListBox>
                    <Grid x:Name="ContentPanel12" 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();
        }

    76   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);
    17      dispatcherTimer.Interval = TimeSpan.FromMilliseconds(250);
        dispatcherTimer.Tick += (o, arg) => ScanPreviewBuffer();
        dispatcherTimer.Start();
    65      base.OnNavigatedTo(e);
    }

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

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

```

```

    PhotoCameraLuminanceSource _luminance = new
PhotoCameraLuminanceSource(17);
    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 en Windows Phone, se hace uso e 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:

20
Ilustración 15



Ilustración 16

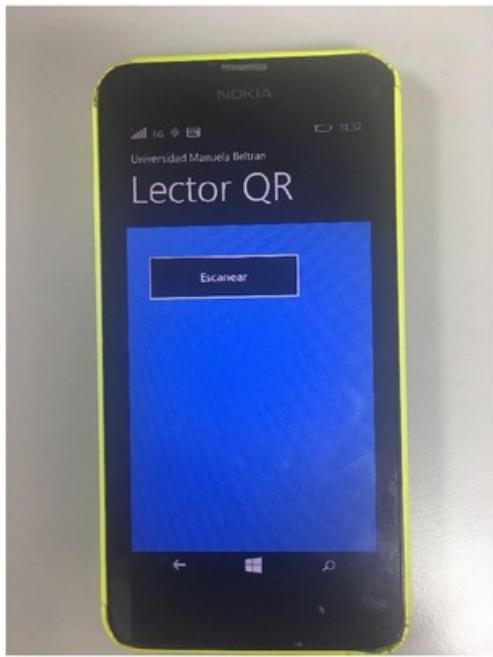


Ilustración 17

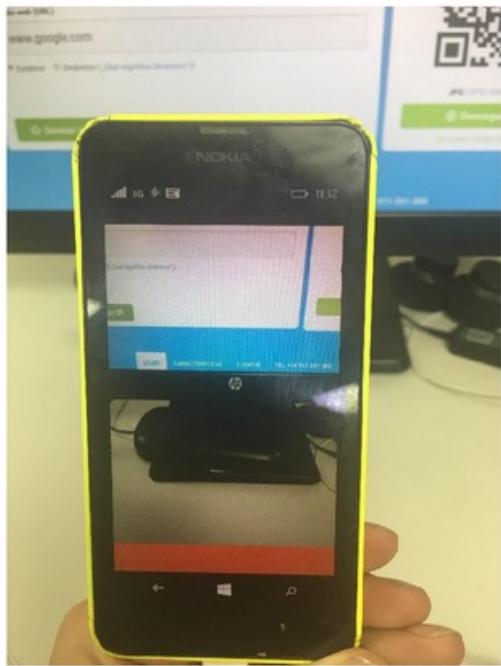


Ilustración 18

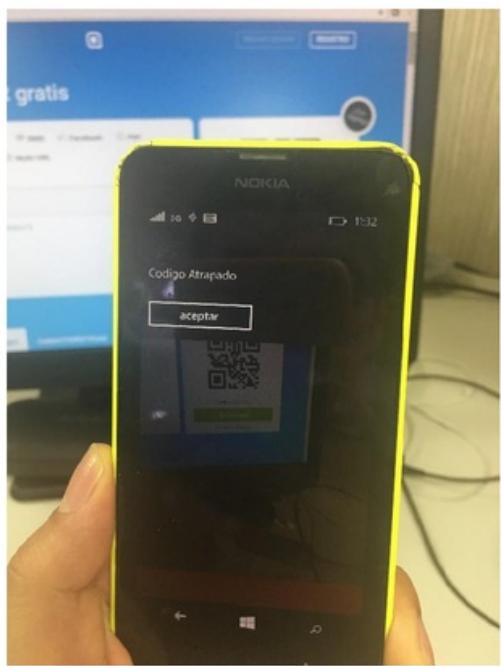


Ilustración 19

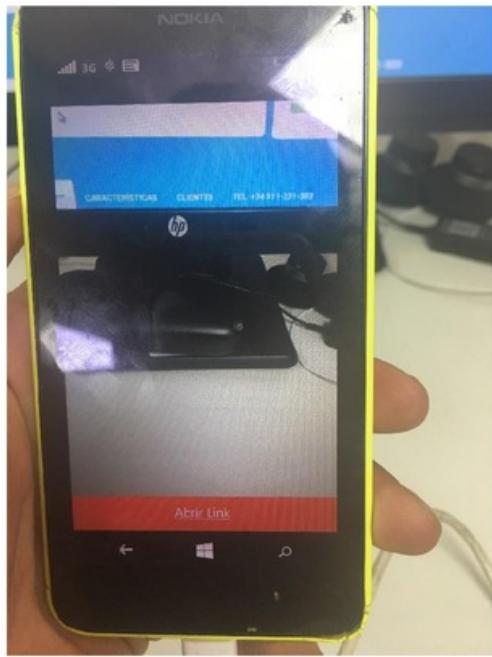
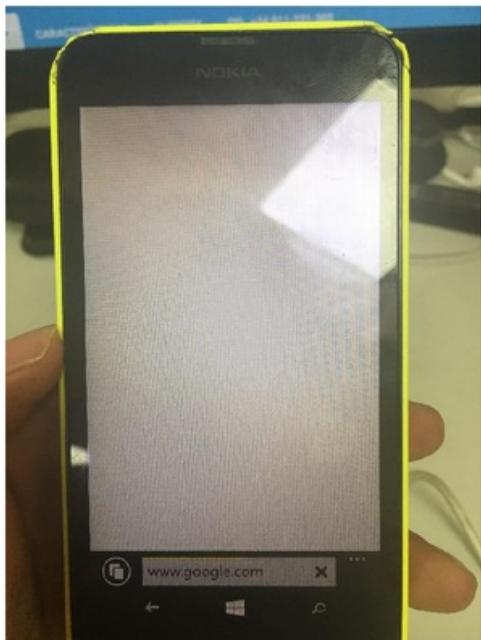


Ilustración 20



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. 9-oct-2010. Aprendizaje Móvil Aplicado en la
Educación Usos prácticos ~ QR Code. México: Universidad autónoma de
México. Recuperado de
repositorio.cuaed.unam.mx:8080/jspui/handle/123456789/1209 [13]
- [2] Denso Wave Incorporated. 2011. Lectores de códigos QR. Recuperado de
www.codigos-qr.com/lectores-codigos-qr [28]
- [3] Juliana Gaviria García. Nuevas tecnologías, nuevas formas de negocios:
Smartphones y códigos QR en Chile. Chile: Universidad del rosario y universidad
del desarrollo. Recuperado de
<http://repository.urosario.edu.co/bitstream/handle/10336/3776/1020744780-2012.pdf?sequence=8&isAllowed=y> [10]
- [4] María Soledad Gómez Vilchez. 2010. QR Code en museos.
<http://mediamusea.files.wordpress.com/2010/10/qr-code-en-museos.pdf> [54]
- [5] Matos García, Bianchi Angleró, Pérez Ortiz, Santiago Guzmán y Silva Delgado.
2014. Proyecto de ley, Estado Asociado de Puerto Rico. Recuperado de
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%2F257B2A9BDA30-C8FC-44D2-90FA-AFB856F0F08C%257D.docx&ei=_Zw9VPqZKNHpggTnvYCIDA&usg=AFQjCNGzYPhsuorbl4EzFufzPxzvebCkLg&sig2=tkyq3tbC_RG1ihen0MKJKw&bvm=bv.77412846,d.eXY [35]

[6] Lucy Johanna Honores Chuchuca, Jenny Gabriela Vizuete Salazar. 2014.
2

Estudio estadístico comparativo entre sensores Android y Windows phone
57
aplicado en la detección de movimientos telúricos. Ecuador: Escuela superior
politécnica de Chimborazo. Recuperado de
<http://dspace.espoch.edu.ec/bitstream/123456789/3540/1/18T00554.pdf>

[7] UNAD. Lección 13 Desarrollo de Aplicaciones. Colombia: Universidad Nacional
Abierta y a Distancia. Recuperado de
http://dataoteca.unad.edu.co/contenidos/233016/EXE_SAM/leccin_13_desarrollo_de_aplicaciones.html

[8] Mike Armas. 2012. Model Driven Architecture (MDA). MSDN Library: artículos
3
técnicos. Recuperado de <https://msdn.microsoft.com/es-es/library/jj130728.aspx>

[9] EcuRed. 09/11/2016. Modelo de prototipos. Recuperado de
https://www.ecured.cu/Modelo_de_Prototipos

[10] Blogspot. 16/04/2013. Metodología en cascada. Recuperado de
<http://metodologiaencascada.blogspot.com.co/>

[11] Rational Unified Process (RUP). Recuperado de
50
<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).
Recuperado de
74
http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=500

[13] Croma Cultura. 2015. ¿Qué es un museo y cuantos tipos de museos existen?. Recuperado de <http://www.cromacultura.com/tipos-de-museos/>

[14] Eusantana. 2014. Los museos y las nuevas tecnologías. Recuperado de <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?. Recuperado de <https://uqr.me/es/blog/museos-generadores-codigos-qr/>

[16] Felipe Ramírez García. 2013. Cuatro grandes museos donde la tecnología es el alma de la exhibición. Enter.co. Recuperado de <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. España: Universidad de Almería. Recuperado de <http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf>

69

[18] Cuerpo de conocimientos de la ingeniería de software. México: Unidad

Profesional Interdisciplinaria de Ingeniería y Ciencias sociales y Administrativas.

Recuperado de <http://www.sites.upiicsa.ipn.mx/polilibros/portal/polilibros/Complemento%20Materia>

I%20Didactico/Maest-Ing-Soft-

Sergio/Cuerpoconocimiento/Construcci%C3%B3n%20del%20software.htm

[19] SG Buzz. Ingeniería de Software. Desarrollar es mucho más que programar.

México. Recuperado de <https://sg.com.mx/content/view/444>

[20] Arely Escobar. Paradigmas De la Ingeniería De Software. Bligoo. Recuperado de <http://arelyescobar.bligoo.com.mx/paradigmas-de-la-ingenieria-de-software#.WAKinvl96Uk>

[21] Ok Hosting. Metodologías del desarrollo de software. OkHosting. Recuperado de <http://okhosting.com/blog/metodologias-del-desarrollo-de-software/>

[22] 2013. Rational Unified Process (RUP). España: Universidad de Gerona. Recuperado de <http://ima.udg.edu/~sellares/EINF-ES2/Present1011/MetodoPesadesRUP.pdf> [50]

[23] Proyectosagiles.org. Que es SCRUM. La web de Scrum. Recuperado de <https://proyectosagiles.org/que-es-scrum/>

[24] Mex.tl. Programación Extrema. Recuperado de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html [56]

[25] Alexey Mathotkin. 2013. Control de Versiones. Producir Software de Código Abierto. Recuperado de <http://producingoss.com/es/vc.html>

[26] Guillem Borrell. 2006. El control de versiones. Recuperado de <http://torroja.dmt.upm.es/media/files/cversiones.pdf>

[27] Luis Artola. 2009. Tipos de pruebas automatizadas de software. Programania.net. recuperado de <http://www.programania.net/diseno-de-software/tipos-de-pruebas-automatizadas-de-software/> [44]

[28] CCM. 2016. Lenguajes de programación. Recuperado de <http://es.ccm.net/contents/304-lenguajes-de-programacion>

[29] Tecnología W Definista. 2014. Definición de Windows Phone. Conceptodefinicionde. Recuperado de <http://conceptodefinicion.de/windows-phone/>

[30] Jose Manuel Martinez Lainez. 2012. Desarrollo y comercialización de una aplicación para una plataforma móvil. España: [Escuela técnica superior de ingenieros industriales y de telecomunicación](#). Recuperado de <http://academica-e.unavarra.es/bitstream/handle/2454/5892/577862.pdf?sequence=1&isAllowed=y>

[31] El vocero. 2016. Códigos QR. Recuperado de <http://elvocero.com/tag/codigos-qr/>

[32] Staff Capital 21. 22/01/2016. Publican modificaciones a la ley de publicidad exterior de la cdmx. México: Capital 21. Recuperado de <http://www.capital21.df.gob.mx/publican-modificaciones-a-la-ley-de-publicidad-exterior-de-la-cdmx/>

[33] Fernando Massa. 23/11/2011. Los códigos QR invadieron el paisaje urbano. Argentina: La nación. Recuperado de <http://www.lanacion.com.ar/1424546-codigos-qr>

[34] Carole Gray and Julian Malins, 2004, Visualizing Research. A Guide to the Research Process in Art and Design, España: Universidad politecnica de Valencia. Recuperado de http://www.upv.es/laboluz/master/seminario/textos/proceso_investigacion.pdf

[35] Susana Galarza Ganan. Modelo XP (Xtreme Programming) para desarrollo de proyecto. Ecuador: Universidad de Milagro. Recuperado de <http://es.slideshare.net/johitaamiga/modelo-xp-para-desarrollo-de-proyecto>

[36] Iso.org. 2008. Systems and software engineering — Software life cycle processes. ISO/IEC. Recuperado de <https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en>

[37] IEEE Computer Society. 2004. Guía al cuerpo de conocimiento de la ingeniería de software SWEBOK. Estados Unidos. Recuperado de <http://www.cc.uah.es/drg/b/HispaSWEBOK.Borrador.pdf>

[38] Daniel Alvarez. 2012. Leer y generar códigos con Zxing. Recuperado de: <http://zomwi.blogspot.com.co/2012/09/zxing.html>

[39] Biljet App. Librería zXing para lectura de códigos QR en Android. Recuperado de <https://biljetapp.wordpress.com/2013/03/04/presentacion/>

Version100

INFORME DE ORIGINALIDAD

38%

INDICE DE SIMILITUD

38%

FUENTES DE
INTERNET

2%

PUBLICACIONES

3%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

www.cc.uah.es

Fuente de Internet

20%

2

docplayer.es

Fuente de Internet

2%

3

ri.ues.edu.sv

Fuente de Internet

1%

4

www.sg.com.mx

Fuente de Internet

1%

5

pt.scribd.com

Fuente de Internet

1%

6

lesteban.info

Fuente de Internet

1%

7

www.slideshare.net

Fuente de Internet

1%

8

es.wikipedia.org

Fuente de Internet

1%

9

ima.udg.edu

Fuente de Internet

1%

| | | |
|----|--|------|
| 10 | repository.urosario.edu.co | 1 % |
| | Fuente de Internet | |
| 11 | metodologiaencascada.blogspot.com.es | 1 % |
| | Fuente de Internet | |
| 12 | prezi.com | <1 % |
| | Fuente de Internet | |
| 13 | www.virtualeduca.info | <1 % |
| | Fuente de Internet | |
| 14 | sites.google.com | <1 % |
| | Fuente de Internet | |
| 15 | Submitted to Southampton Solent University | <1 % |
| | Trabajo del estudiante | |
| 16 | www.apptivismo.org | <1 % |
| | Fuente de Internet | |
| 17 | liny.mrpager.com | <1 % |
| | Fuente de Internet | |
| 18 | saber.ucv.ve | <1 % |
| | Fuente de Internet | |
| 19 | www.c-sharpcorner.com | <1 % |
| | Fuente de Internet | |
| 20 | www.agendasinnovacion.mx | <1 % |
| | Fuente de Internet | |
| 21 | dataoteca.unad.edu.co | |

Fuente de Internet

<1 %

| | | |
|----|---|------|
| 22 | Submitted to West Cheshire College Trabajo del estudiante | <1 % |
| 23 | adigor.es Fuente de Internet | <1 % |
| 24 | geonet.esri.com Fuente de Internet | <1 % |
| 25 | mediamusea.com Fuente de Internet | <1 % |
| 26 | Submitted to Institute of Technology, Sligo Trabajo del estudiante | <1 % |
| 27 | alarcos.inf-cr.uclm.es Fuente de Internet | <1 % |
| 28 | www.codigos-qr.com Fuente de Internet | <1 % |
| 29 | www.pcm.gob.pe Fuente de Internet | <1 % |
| 30 | ultimateintechnology.blogspot.com Fuente de Internet | <1 % |
| 31 | www.codigo-qr.es Fuente de Internet | <1 % |
| 32 | www.monografias.com Fuente de Internet | <1 % |

| | | |
|----|--|------|
| 33 | www.concejoriogrande.gov.ar | <1 % |
| | Fuente de Internet | |
| 34 | Submitted to University of Wales central institutions | <1 % |
| | Trabajo del estudiante | |
| 35 | www.funlam.edu.co | <1 % |
| | Fuente de Internet | |
| 36 | elimpulso.com | <1 % |
| | Fuente de Internet | |
| 37 | www.uci.ac.cr | <1 % |
| | Fuente de Internet | |
| 38 | www.scribd.com | <1 % |
| | Fuente de Internet | |
| 39 | www.spikie.be | <1 % |
| | Fuente de Internet | |
| 40 | hcuadrosb.blogspot.com | <1 % |
| | Fuente de Internet | |
| 41 | es.slideshare.net | <1 % |
| | Fuente de Internet | |
| 42 | tesis.ipn.mx:8080 | <1 % |
| | Fuente de Internet | |
| 43 | servidoresvirtualestopicos.blogspot.com | <1 % |
| | Fuente de Internet | |

| | | |
|----|--|------|
| 44 | www.researchgate.net | <1 % |
| 45 | www.atesar.com | <1 % |
| 46 | karang344.blogspot.com | <1 % |
| 47 | www.willydev.net | <1 % |
| 48 | www.escinf.una.ac.cr | <1 % |
| 49 | eprints.ucm.es | <1 % |
| 50 | pryjectoinf3.googlecode.com | <1 % |
| 51 | www.bdigital.unal.edu.co | <1 % |
| 52 | Submitted to University of London External System Trabajo del estudiante | <1 % |
| 53 | www.derby.ac.uk | <1 % |
| 54 | lab.patrimoniointeligente.com | <1 % |

| | | |
|----|--|------|
| 55 | maromasdigitales.net Fuente de Internet | <1 % |
| 56 | bibdigital.epn.edu.ec Fuente de Internet | <1 % |
| 57 | dspace.ucuenca.edu.ec Fuente de Internet | <1 % |
| 58 | servinf.dif.um.es Fuente de Internet | <1 % |
| 59 | repository.uniminuto.edu:8080 Fuente de Internet | <1 % |
| 60 | Van Arsdale, Suzanne Venzke, Cody. "Predatory innovation in software markets.", Harvard Journal of Law & Technology, Fall 2015 Issue Publicación | <1 % |
| 61 | www.zonamovilidad.es Fuente de Internet | <1 % |
| 62 | www.taringa.net Fuente de Internet | <1 % |
| 63 | repositorio.ug.edu.ec Fuente de Internet | <1 % |
| 64 | ojs.tdea.edu.co Fuente de Internet | <1 % |

Submitted to Nottingham Trent University

65

Trabajo del estudiante

<1 %

66

www.laccei.org

Fuente de Internet

<1 %

67

Quezada-Sarmiento, Pablo Alejandro, Mary Morocho- Quezada, Liliana Pacheco-Jara, and Juan Garbajosa. "Evaluation of occupational and professional profiles in Ecuadorian context based on guide of Knowledge SWEBOK and ontological model", 2016 Third International Conference on eDemocracy & eGovernment (ICEDEG), 2016.

Publicación

<1 %

68

www.tesorillo.com

Fuente de Internet

<1 %

69

www.gestiopolis.com

Fuente de Internet

<1 %

70

Submitted to University of Ulster

Trabajo del estudiante

<1 %

71

infostore.saiglobal.com

Fuente de Internet

<1 %

72

ebookmarket.org

Fuente de Internet

<1 %

73

www.rankdirection.com

Fuente de Internet

<1 %

| | | |
|----|---|------|
| 74 | cipqis.blogspot.com | <1 % |
| 75 | sistemastectuxtla.net | <1 % |
| 76 | www.m0sand.com | <1 % |
| 77 | www.ilides.org | <1 % |
| 78 | www.eqsoft.net | <1 % |
| 79 | aprendiendotecnologiaadmonb.wordpress.com | <1 % |
| 80 | www.navegapolis.net | <1 % |
| 81 | segundomodelo.blogspot.com | <1 % |
| 82 | www.guadalajaraplaza.com | <1 % |
| 83 | elvex.ugr.es | <1 % |
| 84 | mx.livra.com | <1 % |
| 85 | umb.edu.co | <1 % |

| | | |
|----|--|------|
| 86 | www.ctr.unican.es | <1 % |
| 87 | es.windows.wikia.com | <1 % |
| 88 | en.buscotripulantes.com | <1 % |
| 89 | trid.trb.org | <1 % |
| 90 | blog.mycampusnotes.com | <1 % |
| 91 | www.psn.es | <1 % |
| 92 | Microsoft Mapping, 2013. Publicación | <1 % |
| 93 | biblioteca.espoch.edu.ec | <1 % |
| 94 | biblioteca2.ucab.edu.ve | <1 % |
| 95 | wiki.deister.net | <1 % |
| 96 | www.authorstream.com | <1 % |
| 97 | repositorio.utp.edu.co | |

Fuente de Internet

<1 %

98

www.microsoftvirtualacademy.com

Fuente de Internet

<1 %

99

fethusweb2.googlecode.com

Fuente de Internet

<1 %

100

. "Este documento presenta un resumen de Rational Unified Process (RUP). Se describe la historia de la metodologÃ ªa, caracterÃ ¸sticas principales y estructura del proceso", university degree/european languages literature and related subjects/modern foreign languages/spanish/1, 2013.

Publicación

<1 %

EXCLUIR CITAS

APAGADO

EXCLUIR BIBLIOGRAFÍA APAGADO

EXCLUIR
COINCIDENCIAS

APAGADO