

## ÍNDICE DE TABLAS

CONTENIDO	PÁGINAS
TABLA 1: Objetivos específicos y acciones .....	11
TABLA 2: Contenido temático .....	16
TABLA 3: Capacidad de datos código QR .....	68
TABLA 4: Diseño de la investigación junto con las herramientas .....	97
TABLA 5: Identificación de deficiencias .....	107
TABLA 6: Comparativa de modelos de desarrollo de software .....	113
TABLA 7: Tabla de descripción de historias de usuarios .....	116
TABLA 8: Realease plan XP .....	118
TABLA 9: Planificación de iteraciones del proyecto .....	120
TABLA 10: Relación de los entregables y objetivos .....	121
TABLA 11: Tabla comparativa de lenguajes de programación .....	122
TABLA 12: Tabla comparativa de framework .....	124
TABLA 13: Tabla comparativa de Gestores de Base de Datos .....	126
TABLA 14: Historia de usuario 1 .....	128
TABLA 15: Historia de usuario 2 .....	129
TABLA 16: Historia de usuario 3 .....	129
TABLA 17: Historia de usuario 4 .....	130
TABLA 18: Historia de usuario 5 .....	130
TABLA 19: Prueba 1 .....	134
TABLA 20: Prueba 2 .....	135
TABLA 21: Pruebas de aceptación .....	135
TABLA 22: Historia de usuario 6 .....	136
TABLA 23: Historia de usuario 7 .....	136
TABLA 24: Prueba 3 .....	147
TABLA 25: Prueba 4 .....	147
TABLA 26: Pruebas de aceptación .....	147
TABLA 27: Historia de usuario 8 .....	149

TABLA 28: Selección de algoritmo de generación de código único .....	149
TABLA 29: Prueba 5 .....	157
TABLA 30: Pruebas de aceptación .....	157
TABLA 31: Historia de usuario 9 .....	159
TABLA 32: Historia de usuario 10 .....	159
TABLA 33: Prueba 6 .....	167
TABLA 34: Pruebas de aceptación .....	168
TABLA 35: Historia de usuario 11 .....	169
TABLA 36: Historia de usuario 12 .....	169
TABLA 37: Prueba 7 .....	176
TABLA 38: Prueba 8 .....	176
TABLA 39: Pruebas de aceptación de módulo de reportes .....	177
TABLA 40: Pruebas de integración .....	178
TABLA 41: Pruebas de integración nivel 0 .....	179
TABLA 42: Pruebas de integración nivel 1 .....	179
TABLA 43: Pruebas de integración nivel 2 .....	180
TABLA 44: Pruebas de integración nivel 3 .....	180
TABLA 45: Requerimiento mínimo e ideal de hardware .....	181
TABLA 46: Requerimiento mínimo e ideal de software .....	181
TABLA 47: Constante por modo de desarrollo .....	182
TABLA 48: Ecuaciones básicas en COCOMO .....	183
TABLA 49: Ecuaciones básicas en COCOMO intermedio.....	183
TABLA 50: Factores de riesgo en COCOMO .....	183
TABLA 51: Costos en base a los requerimientos mínimos .....	187
TABLA 52: Costos en base a los requerimientos ideales .....	187
TABLA 53: Costos en base a los requerimientos ideales .....	187
TABLA 54: Nivel de complejidad del cargo .....	189

## ÍNDICE DE FIGURAS

CONTENIDO	PÁGINAS
FIGURA 1: Diagramas UML 2.0 .....	35
FIGURA 2: Diagrama de caso de uso .....	35
FIGURA 3: Diagrama de clases.....	36
FIGURA 4: Diagrama de secuencia.....	37
FIGURA 5: Diagrama de colaboración .....	38
FIGURA 6: Características del código de barras .....	47
FIGURA 7: Características del código de barras .....	55
FIGURA 8: Elementos de workflow .....	71
FIGURA 9: Diferentes tecnologías y productos que usa los sistemas workflow ...	72
FIGURA 10: Diagrama de contexto .....	102
FIGURA 11: Modelado de negocio actual documentación cursantes.....	103
FIGURA 12: Modelado de negocio actual documentación docentes .....	104
FIGURA 13: Modelado de negocio actual consulta documentación cursantes ....	105
FIGURA 14: Modelado de negocio actual consulta documentación docentes .....	106
FIGURA 15: Identificación de componentes workflow.....	109
FIGURA 16: Definición de componentes workflow.....	109
FIGURA 17: Proceso alternativo de entrega de documentación de cursantes....	110
FIGURA 18: Proceso alternativo de entrega de documentación de docentes .....	111
FIGURA 19: Proceso alternativo de generación de reportes .....	112
FIGURA 20: Identificación de Usuarios .....	116
FIGURA 21: Actores del módulo de gestión de usuarios.....	131
FIGURA 22: Diagrama de casos de uso general de gestión de usuarios .....	131
FIGURA 23: Diagrama de caso de uso de director .....	132
FIGURA 24: Diagrama de caso de uso del Director de Posgrado.....	132
FIGURA 25: Diagrama de clases gestión de usuarios .....	133
FIGURA 26: Diagrama de base de datos gestión de usuarios .....	133
FIGURA 27: Actores del módulo de admisión de documentación .....	137
FIGURA 28: Actores del módulo de admisión de documentación .....	138

FIGURA 29: Actores del módulo de admisión de documentación .....	139
FIGURA 30: Actor encargada de informática.....	140
FIGURA 31: Actor secretaria .....	141
FIGURA 32: Diagrama de clases.....	142
FIGURA 33: Actor del módulo de recepción de documentación.....	142
FIGURA 34: Diagrama de estados api REST .....	143
FIGURA 35: Diagrama de capas arquitectura MVT .....	144
FIGURA 36: Servicio de registro de api REST .....	146
FIGURA 37: Actores del módulo.....	151
FIGURA 38: Diagrama de caso de uso del módulo .....	152
FIGURA 39: Actor secretaria .....	153
FIGURA 40: Actor encargada de sistemas .....	154
FIGURA 41: Diagrama de clases.....	154
FIGURA 42: Diagrama de base de datos .....	155
FIGURA 43: Diagrama de estados api REST .....	156
FIGURA 44: Actores del módulo.....	160
FIGURA 45: Diagrama de caso de uso .....	161
FIGURA 46: Actor director .....	162
FIGURA 47: Actor coordinador académico.....	163
FIGURA 48: Diagrama de clases.....	164
FIGURA 49: Diagrama de base de datos .....	165
FIGURA 50: Diagrama de estados api REST .....	166
FIGURA 51: Actores del módulo.....	170
FIGURA 52: Diagrama de casos de uso.....	171
FIGURA 53: Actor director .....	172
FIGURA 54: Actor secretaria .....	173
FIGURA 55: Actor coordinador academico.....	174
FIGURA 56: Actor encargada de sistemas .....	175
FIGURA 57: Diagrama de base de datos .....	175
FIGURA 58: Diagrama de estados api REST .....	176
FIGURA 59: Cantidad de líneas de código utilizando CLOC.....	185

## ÍNDICE DE CONTENIDO

CONTENIDO	PÁGINAS
1. GENERALIDADES .....	1
1.1. INTRODUCCIÓN .....	1
1.2. ANTECEDENTES.....	3
1.3. PLANTEAMIENTO DEL PROBLEMA.....	8
1.3.1. Identificación del problema .....	8
1.3.2. Formulación del problema .....	9
1.3.3. Análisis causa – efecto .....	10
1.4. OBJETIVOS Y ACCIONES .....	10
1.4.1. Objetivo general.....	10
1.4.2. Objetivos específicos y acciones .....	10
1.5. JUSTIFICACIÓN.....	13
1.5.1. Justificación técnica .....	13
1.6. DELIMITACION DE LA INVESTIGACIÓN .....	14
1.6.1. Delimitación temática.....	14
1.6.2. Delimitación Espacial.....	14
1.6.3. Delimitación temporal .....	15
1.6.4. Delimitación funcional.....	15
2. MARCO TEÓRICO .....	16
2.1. CONTENIDO TEMÁTICO.....	16
2.2. TÉCNICAS DE RECOPILACIÓN DE INFORMACIÓN .....	19
2.2.1. Entrevistas .....	20
2.2.2. Cuestionarios .....	20
2.2.3. Observación.....	21

2.3.	INGENIERÍA DE SOFTWARE .....	21
2.3.1.	Desarrollo ágil .....	21
2.3.2.	Política de desarrollo ágil .....	22
2.3.3.	Programación Extrema .....	23
2.3.4.	Lenguaje unificado de desarrollo (UML).....	34
2.3.5.	Pruebas de software.....	38
2.4.	ARQUITECTURA ORIENTADA A SERVICIOS.....	44
2.4.1.	Beneficios .....	44
2.4.2.	Características .....	45
2.4.3.	Componentes de la arquitectura orientada a servicios.....	45
2.4.4.	Funcionamiento de SOA.....	46
2.4.5.	Capas de software.....	46
2.5.	SEGURIDAD DE INFORMACION .....	47
2.5.1.	Aspectos generales .....	50
2.5.2.	CRYPTOGRAFÍA.....	51
2.5.3.	Código de barras .....	53
2.5.3.1.	Como funciona el código de barras .....	54
2.5.3.2.	Metodología empleada .....	55
2.5.4.	Generadores de códigos únicos simétricos .....	56
2.5.4.1.	Algoritmo Random Code.....	56
2.5.4.2.	Algoritmo DES .....	58
2.5.4.3.	Universally Unique Identifier UUID .....	60
2.5.5.	Firma digital .....	63
2.5.5.1.	Funcionamiento de firma digital .....	64
2.5.5.2.	Clave asimétrica .....	64

2.5.5.3. Hash.....	64
2.5.5.4. Certificación digital.....	65
2.5.6. Código QR .....	65
2.5.7. Características de código QR.....	67
2.6. WORKFLOW .....	68
2.6.1. Definición de sistemas workflow.....	68
2.6.1.1. Procesamiento de imágenes .....	69
2.6.1.2. Administración de documentos.....	69
2.6.1.3. Correo electrónico y directorios .....	69
2.6.1.4. Aplicaciones basadas en transacciones.....	70
2.6.1.5. Procesamiento de formularios .....	70
2.6.2. Usos del workflow .....	70
2.6.3. Elementos de un sistema workflow.....	71
2.6.4. Tecnologías utilizadas en sistemas workflow .....	72
2.6.5. Tipos de sistemas workflow .....	72
2.6.5.1. Workflow de colaboración.....	72
2.6.5.2. Workflow de administración .....	73
2.7. TECNOLOGÍAS DE DESARROLLO .....	73
2.7.1. Lenguajes de programación .....	73
2.7.2. Frameworks .....	80
2.8. DISEÑO RESPONSIVO WEB .....	85
2.8.1. HTML .....	86
2.8.2. Bootstrap.....	88
2.8.3. CSS.....	88
2.8.4. JavaScript .....	89

2.9.	GESTORES DE BASES DE DATOS .....	90
2.9.2.	Sistemas gestores de base de datos.....	91
3.	INGENIERÍA DEL PROYECTO .....	96
3.1.	MARCO METODOLÓGICO.....	96
3.1.1.	Enfoque de la Investigación.....	96
3.1.2.	Tipo de investigación .....	96
3.1.3.	Método de Investigación.....	96
3.2.	DISEÑO DEL MODELO DE NEGOCIO ACTUAL Y ALTERNATIVO EN BASE A CARACTERISTICAS WORKFLOW .....	101
3.2.1.	Análisis de requerimientos.....	101
3.2.2.	Modelado de negocio actual.....	102
3.2.3.	Identificación de deficiencias .....	107
3.2.4.	Identificación de componentes workflow flujo de trabajo .....	109
3.2.5.	Modelado del diagrama de flujo alternativo. ....	110
3.2.5.1.	Modelado del diagrama de flujo alternativo entrega documentación de cursantes.....	110
3.2.5.2.	Modelado del diagrama de flujo alternativo entrega documentación de docentes.....	111
3.2.5.3.	Modelado del diagrama de flujo alternativo generación de reportes. ....	112
3.2.6.	Selección de metodología de desarrollo .....	112
3.2.7.	Planificación de metodología .....	115
3.2.7.1.	Planificación de los incrementos del sistema en base al método de desarrollo seleccionado .....	115
3.2.7.2.	Identificación de Usuarios .....	115
3.2.7.3.	Historias de Usuario.....	116
3.2.7.2.	Realease del plan XP.....	118

3.2.7.2. Planificación de las iteraciones del proyecto .....	120
3.2.7.2. Relación entre los entregables y objetivos específicos del proyecto.....	121
3.3. DESARROLLO DEL MÓDULO DE REGISTRO DE USUARIOS CON ROLES Y PRIVILEGIOS RESPECTIVOS .....	122
3.3.1. Selección de un lenguaje de programación.....	122
3.3.2. Selección de framework adecuado.....	124
3.3.3. Selección de un gestor de base de datos.....	126
3.3.5. Análisis de requerimientos.....	127
3.3.5.1. Tarjetas de historias de usuarios .....	128
3.3.6. Diagramas UML para el módulo de registro de usuarios.....	130
3.3.6.1. Diagramas de caso de uso para el módulo de registro de usuarios.....	130
3.3.6.2. Diagrama de clases para el módulo de gestión de usuarios. ....	133
3.3.7. Diseño de la base de datos .....	133
3.3.7.1. Diagrama de base de datos.....	133
3.3.7. Implementar módulo de registro de usuarios.....	133
3.3.7.1. Código de modelo de usuarios .....	133
3.3.7.2. Código crear usuarios.....	134
3.3.8. Pruebas de aceptación .....	134
3.4. DESARROLLO DEL MÓDULO DE ADMISIÓN DE DOCUMENTACIÓN DOCENTE, DE CURSANTES Y ALMACENAMIENTO.....	135
3.4.1. Analizar requerimientos .....	135
3.4.1.1. Requerimientos funcionales del primer incremento.....	136
3.4.1.2. Tarjetas de historias de usuario del módulo admisión de documentación....	136
3.4.2. Diagramas UML para el módulo de admisión de documentación .....	137

3.4.2.1. Diagramas de casos de uso para el módulo de admisión de documentación.....	137
3.4.2.2. Diagramas de casos de uso del módulo de recepción de documentos... ..	138
3.4.2.3. Diagrama de casos de uso general .....	138
3.4.2.4. Diagrama de casos de uso por actor .....	139
3.4.2.5. Diagramas de clases módulo recepción de documentación.....	142
3.4.3. Actualizar base de datos.....	142
3.4.3.1. Diagramas de bases de datos módulo recepción de documentación .....	142
3.4.4. Diseño de servicios de la api REST.....	143
3.4.4.1. Diagrama de secuencia para el diseño de la api REST del módulo de admisión de documentación. ....	143
3.4.4.2. Diagrama de por capas con la arquitectura MVT .....	144
3.4.5. Implementar el módulo para la admisión de documentación.....	144
3.4.5.1. Modelo Docentes .....	145
3.4.5.2. Modelo Cursantes.....	145
3.4.5.3. Servicio de registro de la api REST .....	146
3.4.6. Pruebas al módulo.....	147
3.5. DESARROLLO DEL MÓDULO DE GENERACIÓN DE CÓDIGO ÚNICO PARA CADA DOCUMENTO PRESENTADO.....	148
3.5.1.1. Diagrama de base de datos módulo de generación de código único .....	155
3.5.1.2. Diagrama de secuencia para el módulo de generación de código único .....	156
3.5.2. Implementar código para el módulo de generación de código único.....	156
3.5.2.1. Generación de código unico .....	156
3.5.2.2. Implementación en el modelo .....	157
3.6. DESARROLLO DEL MÓDULO DE GENERACION DE CÓDIGO QR A LA DOCUMENTACIÓN PRESENTADA.....	158

3.6.1.	Analizar requerimientos .....	158
3.6.1.1.	Requerimientos funcionales.....	158
3.6.1.2.	Requerimientos no funcionales.....	158
3.6.1.3.	Tarjetas de historias de usuario del módulo de inclusión de código QR a la documentación presentada.....	159
3.6.2.	Diseño de diagramas UML para el módulo.....	160
3.6.2.1.	Identificación de actores .....	160
3.6.2.2.	Diagramas de caso de uso .....	161
3.6.2.3.	Diagrama de clases .....	164
3.6.3.	Actualizar la base de datos .....	165
3.6.3.1.	Diagrama de base de datos.....	165
3.6.4.	Diseño de servicios de la api REST.....	166
3.6.5.	Diagrama de secuencia para el módulo de inclusión de código QR a la documentación presentada.....	166
3.6.6.	Implementar código para el modulo de inclusión de código QR a la documentación presentada.....	166
3.6.6.1.	Modelo de inclusion de QR .....	166
3.6.6.2.	Generar QR .....	167
3.6.7.	Realizar pruebas.....	167
3.7.	GENERACION REPORTES EN BASE A MÓDULOS IMPLEMENTADOS ...	
	168	
3.7.1.	Analizar requerimientos .....	168
3.7.1.1.	Requerimientos funcionales.....	168
3.7.1.2.	Requerimientos no funcionales.....	168
3.7.1.3.	Tarjetas de historias de usuario del módulo de generación de reportes en base a modelos implementados .....	169

3.7.2.	Casos de uso del módulo de generación de reportes .....	170
3.7.2.1.	Identificación de actores .....	170
3.7.2.2.	Casos de uso .....	171
3.7.3.	Diagrama de base de datos del módulo de generación de reportes .....	175
3.7.4.	Diseño de servicios de la api REST.....	176
3.7.4.1.	Diagrama de secuencia para servicios de la api REST del módulo de generación de reportes .....	176
3.7.4.2.	Realizar pruebas al módulo .....	176
3.8.	REALIZAR PRUEBAS AL SISTEMA CONCLUIDO.....	177
3.8.1.	Selección de pruebas adecuadas al proyecto .....	177
3.8.2.	Documentación de pruebas realizadas.....	178
3.8.3.	Pruebas de integración .....	178
3.8.3.1.	Nivel 0.....	178
3.8.3.2.	Nivel 1 .....	179
3.8.3.3.	Nivel 2 .....	180
3.8.3.4.	Nivel 3 .....	180
4.	ANALISIS DE VIABILIDAD .....	181
4.1.	VIABILIDAD TÉCNICA.....	181
4.2.	VIABILIDAD OPERATIVA.....	188
5.	CONCLUSIONES Y RECOMENDACIONES .....	190
5.1.	CONCLUSIONES .....	190
5.2.	RECOMENDACIONES.....	192
5.2.1.	Recomendaciones de funcionamiento.....	192
5.2.2.	Recomendaciones a futuro .....	193

## BIBLIOGRAFÍA

## ANEXOS

ANEXO 'A' ENTREVISTA

ANEXO 'B' LISTA DE ESTUDIANTES INSCRITOS A DIPLOMADO DPDC

ANEXO 'C' LISTA DE DOCENTES

ANEXO 'D' LISTA DE REQUISITOS DE DOCUMENTOS PARA INSCRIPCIÓN DE CURSANTES

## **1. GENERALIDADES**

### **1.1. INTRODUCCIÓN**

A partir de la década de 1980, varios fabricantes comenzaron a desarrollar sistemas de software para gestionar documentos de papel. Estos sistemas trataban con documentos que incluían no sólo documentos impresos y publicados, sino también fotografías, grabados, etc.

La información se ha convertido en un recurso esencial para la creación de valor en las organizaciones. Con el transcurrir del tiempo, la documentación gestionada conforma la memoria de la organización, de ahí la importancia de una adecuada gestión, conservación y difusión del material documental.

La criptografía es la técnica que protege documentos y datos, funciona a través de cifras o códigos para escribir algo secreto en documentos y datos confidenciales que circulan en redes locales o internet. Su utilización es tan antigua como la escritura, los romanos usaban códigos para ocultar sus proyectos de guerra de aquellos que no debían conocerlos, con el fin de que solo las personas que conocían dichos códigos, descifren dichos significados.

La gestión de documentos de forma sistematizada es capaz de personalizar soluciones adaptadas a la medida de las necesidades que una organización demande. Por lo tanto no se trata de una solución específica sino adaptable, permitiendo parametrizar cualquier necesidad a la hora de estructurar documentos, gestionar contenidos o automatizar procesos.

Un software de gestión documental es una aplicación que permite el tratamiento, conservación, publicación y trabajo sobre documentos electrónicos (ya sean documentos escaneados o que se haya creado originalmente en digital). En países de habla hispana, se utiliza el término gestión documental, aunque con las características actuales de las empresas, es más correcto usar el término anglosajón Enterprise Content Management ECM, debido a que las empresas

manejan un amplio tipo de activos digitales que no corresponden sólo con lo que se conoce como documento (Imágenes, vídeos, planos, etc.).

Hoy en día las organizaciones necesitan acceder y consultar de forma constante la información archivada. La importancia de los documentos y la cantidad de información que existe es la principal motivación a buscar soluciones eficaces, que ofrezcan ventajas y puedan generar una maximizaron en sus recursos humanos.

Los archivos físicos representan una acumulación incontrolable de papeles y documentos que se guardan en un deposito y hasta en algunas gestiones sin ningún criterio de organización y bajo ninguna pauta de seguridad.

Todas las empresas grandes o pequeñas, públicas o privadas, trabajan con documentos y contenidos digitales, en ellos está contenida toda la información y el conocimiento del negocio, es por eso que la gestión de estos contenidos se convierte en una clave fundamental para la supervivencia y crecimiento de las organizaciones, en otras palabras, no importa el tamaño o sector de las empresas, todas necesitan la gestión documental.

La gestión de archivos facilita localizar, recopilar, almacenar, proteger y canalizar la información para su revisión y aprobación. Se estima que en promedio un trabajador tarda 12 minutos en procesar un solo documento y nueve minutos en buscar, sacar y volver a archivar el documento. Con la administración de documentos, se puede tener fácil acceso a documentos y buscar en una variedad de tipos de datos, como documentos digitales y fotografías. La administración de documentos ayuda al personal a compartir información con los empleados de manera segura y oportuna.

El propósito de utilizar herramientas criptográficas se ve reflejada en la necesidad de la institución, en este caso la Dirección de Posgrado de la Escuela Militar de Ingeniería U.A. CBBA para mejorar el tratamiento de gestión de documentos, además de garantizar la confidencialidad e integridad de los mismos, para que la

plataforma sea de utilidad para la institución permitiendo manejar toda su información de manera eficaz y oportuna.

El propósito del presente proyecto es desarrollar una plataforma ECM (Enterprise Content Management por sus siglas en inglés) responsive para la gestión de documentación de la Escuela Militar de ingeniería enfocado como caso de estudio a la dirección de posgrado aprovechando las características que estas plataformas tienen como por ejemplo, brindar un completo repositorio de contenido de documentos, la adaptabilidad, rapidez, seguridad, gracias a su arquitectura orientada a los servicios, la plataforma ECM puede diseñarse de manera flexible conforme a necesidades y tareas e integrarse en todas las aplicaciones de la organización, en este caso la dirección de posgrado, archivar y garantizar la integridad de documentos, ademas de desplegar la plataforma ante el uso de cualquier dispositivo electrónico, para mejorar la agilidad, eficacia y calidad de cada uno de sus procesos de gestión de documentación académica, maximizar productividad de los empleados, obtener mayor utilidad y aprovechamiento de la información que maneja, así mismo posteriormente poder ampliarse a otras áreas de la Escuela Militar de Ingeniería.

## **1.2. ANTECEDENTES**

Un 26 de octubre de 1950 mediante un Decreto Supremo (D. S. 02226) se crea la Escuela Militar de Ingeniería (EMI) con sus diferentes carreras de ingeniería. A partir del 10 de noviembre de 1980 se acepta el ingreso de estudiantes civiles. En 1985 el comité ejecutivo de las universidades bolivianas reconoce a la EMI como una universidad del sistema y a partir de 1986 emite títulos en Provisión Nacional.

Mediante resolución del consejo académico N° 01/97 de fecha 03 de febrero de 1977 se crea el departamento de Posgrado de la EMI La Paz y posteriormente se crea el Departamento de Posgrado de la EMI Cochabamba iniciando sus actividades el año 2002 con la primera versión de la maestría en educación superior universitaria (MESU - I).

A partir de la gestión 2013 el departamento de Posgrado se convierte en Dirección de Posgrado, adquiriendo una dependencia de la admisión académica e incrementando la ejecución de varios programas de posgrado tales como maestrías, diplomados y cursos de educación continua.

En la actualidad la dirección de Posgrado Cochabamba esta dirigida por el Cnl. DAEN Rene Bacarreza Molina y cuenta con tres áreas Sistemas, programas, marketing, estas áreas trabajan conjuntamente realizando y coordinando los distintos procesos de inscripción y la gestión de documentación académica, apertura de diplomados, maestrías, doctorados, publicidad, seguimiento de pagos de cursantes de los diferentes programas, análisis de punto de equilibrio para los diferentes programas, etc. Estas áreas realizan una serie de procedimientos para la inscripción de cursantes, registro y contratación de docentes, admisión de documentación tanto para cursantes como docentes.

Los procesos de inscripción la dirección de Posgrado requiere que el cursante haya depositado el monto necesario al número de cuenta institucional en el banco Unión para el programa al que esta postulando, y debe entregar documentos necesarios para la inscripción al programa que este interesado, por lo general estos documentos son documentación académica tales como, título en provisión nacional, certificados de nacimiento, cédula de identidad, DNI's extranjeros si el postulante tiene nacionalidad extranjera, en algunos casos certificados de bachillerato, etc. (Ver anexo "D").

Para los programas de maestría los cursantes deben tener ciertos requisitos para poder cursar ese programa (Ver anexo "D"). Sucede lo mismo para los programas de diplomado, los cursantes deben cumplir con ciertos requisitos para poder cursar un diplomado, recientemente la dirección de posgrado posibilitó de que se puedan realizar doctorados en la EMI U.A. Cochabamba.

Los cursantes aspirantes a los distintos programas ya sean de diplomados, maestrías o doctorados deben entregar un file personal con toda la documentación pertinente al programa que estén interesados después de haber realizado su

inscripción con el comprobante de pago al banco Union, el plazo para que los cursantes entreguen el file personal es de 15 días máximo, anteriormente existía un plazo de 45 días pero este se descartó debido a que dichos files no eran entregados por parte de algunos cursantes. La secretaria de posgrado es quien recepciona la documentación entregada por los cursantes verificando que los documentos sean los correctos y estén a cabalidad, la secretaria utiliza un formulario de registro denominado “Registro: Files de ingreso cursantes” donde están listados la documentación requerida de los cursantes, conforme se revisa cada documentación requerida se tiquea la documentación presentada (Ver anexo “D”), posteriormente la secretaria entrega una copia del formulario de “Registro: Files de ingreso cursantes” con un sello de “Recibido”, algunas oportunidades se recepcionan files con la ausencia de un par de documentos, los cuales el cursante se compromete a entregar en los próximos días, en la copia del formulario de “Registro: Files de ingreso cursantes” que entrega la secretaria al cursante, no se tiquean esos documentos ausentes para que el cursante sepa que documentos se entregó y que documentos le faltan entregar, pero en ocasiones excepcionales estos formularios se extravían y se debe inspeccionar nuevamente que documentos faltan, buscando la carpeta del cursante y hacer un nuevo formulario de “Registro: Files de ingreso cursantes”. Existen oportunidades en que la secretaria esta ausente por unos momentos y algunos cursantes pretenden entregar su documentación pero al estar la secretaria ausente esto se les imposibilita y muchos de los cursantes son personas cortas de tiempo, ante esto algún empleado de posgrado de buena voluntad recepciona la documentación, en base al formulario “Registro: Files de ingreso cursantes” (Ver Anexo “D”) pero a veces sucede que este empleado pasa por alto alguna documentación ya sea debido a distracción u otros factores externos, entonces la documentación es entregada de forma incompleta, cuando se percata de que la documentación de algún cursante esta incompleta viene a ser un problema para la secretaria, puesto que este tipo de situaciones se dan cuando se están titulando los cursantes o están aconteciendo procesos de acreditación o auditoria. Una vez concluida la recolección y recepción de la documentación se procede a almacenarlas en

carpetas las cuales tienen el nombre del cursante y son almacenadas en estantes destinados a los diferentes programas que ofrece posgrado.

Los procesos de registro y contratación de un docente también son realizados en base a una serie de procedimientos los cuales se realizan de manera manual, la Dirección de Posgrado tiene una cierta cantidad de requisitos en cuanto a documentación que cada docente debe poseer y entregar para poder ser registrado y habilitado para dictar clases en cada uno de los programas aperturados y efectuar su contratación.

El coordinador académico realiza la entrevista al postulante a docente y analiza todos los documentos con los que cuenta el posible docente, los cuales deben ser los requeridos para que un docente tenga la posibilidad de dictar clases en los diferentes programas que ofrece Posgrado, posteriormente el coordinador académico debe llevar los documentos a secretaría de Posgrado, para luego ser analizados por el director de Posgrado, una vez analizados los documentos y teniendo la aprobación del director, la secretaría realizará todo el proceso para la aceptación del docente, dicho proceso establece que se debe llenar planillas excel manualmente, el currículum presentado por el docente debe contener los documentos necesarios para que pueda ser contratado, terminado este procedimiento, se debe realizar el proceso de contratación y asignación del docente a un módulo o programa.

La documentación requerida para la contratación de docentes son: títulos en provisión nacional, certificados de maestría, diplomados, doctorados, certificados o avales de educación continua, hoja de vida (Curriculum Vitae), carnet de identidad vigente, certificados de nacimiento, eventualmente títulos homologados por la CUB si existen docentes con formación en el extranjero DNI's internacionales ante la posible nacionalidad extranjera o doble nacionalidad.

Una vez concluida la contratación del docente, el docente tiene 5 días para completar documentación pertinente si es que existiera documentación que faltara ,posteriormente toda la información y documentación es almacenada en

carpetas diferentes donde solo especifica el nombre del docente, estas carpetas son acomodadas en diferentes estantes sin ninguna especificación adicional al nombre correspondiente del docente.

Actualmente la documentación almacenada de docentes y estudiantes esta rebasando la capacidad de los estantes donde las carpetas contenedoras de documentación son almacenadas y con cada gestión que transcurre la documentación tanto de docentes como de estudiantes esta creciendo, por lo tanto la capacidad de almacenaje de esta información histórica de la dirección de posgrado se esta desbordando, la recepción de documentos no tiene un seguimiento de que personal recepcionó dicha documentación mucho menos un comprobante que establezca por quien fue revisada la documentación recepcionada, existen circunstancias donde los documentos entregados por los docentes son extraviados por lo cual se debe solicitar al docente que vuelva a entregar documentación ante el extravío de esta, por lo general la dirección de posgrado se da cuenta de la ausencia de estos documentos ante procesos de auditoria, evaluación de acreditación, lo cual repercute en problemas, llamadas de atención al director de posgrado y por consiguiente al personal civil encargado de estos documentos o los documentos ausentes son encontrados por lo tanto sucede la duplicidad de documentos, también sucede que al requerir consultar la información de un docente o estudiante en particular estos documentos se entrepapelan, debido a las diferentes labores que realiza el personal de posgrado lo cual produce una confusión al momento de querer reorganizarlas e incluso este entrepapelado no es percatado por el personal de posgrado, sino cuando se realizan auditorias o controles de acreditación se ve que algunos documentos académicos no tienen coherencia, perjudicando al personal de posgrado involucrado.

La confidencialidad de esta documentación no es garantizada, puesto que la información académica docente o de cursantes no tiene ningún tipo de seguridad, pues solo están almacenadas físicamente en carpetas dentro de las oficinas, por lo que cualquier persona mal intencionada puede obtener la información y datos

sensibles de docentes o cursantes e incluso estos documentos pueden llegar a ser alterados.

Existen proyectos similares como laserFiche que es un software de gestión de archivos comerciales, otro es Athento un software desarrollado para uso comercial de gestión documental, sin embargo hay que destacar que los precios son muy elevados y existen escaso software de gestión documental ECM (por sus siglas en inglés) de código abierto.

### **1.3. PLANTEAMIENTO DEL PROBLEMA**

A continuación se mencionan los siguientes puntos para poder determinar el problema de manera clara:

#### **1.3.1. Identificación del problema**

En base a entrevistas y observaciones realizadas, se pudo evidenciar que el proceso de registro manual de documentos recepcionados y derivación manual de documentación académica de cursantes y docentes es poco eficiente, los procesos manuales pueden llegar a ocasionar duplicidad, entrepapelado de documentación, el desborde de cantidad de documentación histórica de gestiones pasadas provoca que los documentos almacenados puedan extraviarse, ademas de que la cantidad de documentación recepcionada va incrementando en cada gestión, el tratamiento de documentos con información sensible no es la adecuada, integridad y confidencialidad poco garantizada de la documentación académica de gran importancia para la dirección de posgrado de la EMI U.A. Cochabamba, la búsqueda de información oportuna tiene una demora de tiempo considerable.

### **1.3.1.1. Identificación de la situación problemática**

- Los procedimientos de almacenamiento de documentación en base al uso de carpetas y planillas provoca que los documentos se lleguen a entrepapelar al momento de organizar la documentación.
- La acumulación de documentación de gestiones pasadas provoca el desbordamiento de papeles en los estantes donde la documentación de docentes y cursantes es almacenada.
- El manejo actual de documentos con información sensible provoca que la confidencialidad de los documentos pueda ser vulnerada, la integridad no sea garantizada.
- El registro manual de documentos recepcionados de los cursantes y docentes, genera confusión y duplicidad en la presentación de documentos.

### **1.3.1.2. Identificación de las causas**

Las causas encontradas son:

- Los procedimientos de almacenamiento de documentación en base al uso de carpetas y planillas.
- La acumulación de documentación de gestiones pasadas.
- El manejo actual de documentos con información sensible.
- El registro manual del registro de recepción de documentos entregados por cursantes y docentes.

### **1.3.2. Formulación del problema**

Los inadecuados procesos actuales basados en el registro manual aplicados a la gestión de recepción, organización de documentos y almacenamiento de toda la información académica de docentes y cursantes provoca entrepapelado de documentación, desbordamiento de papeles en los estantes, vulnerabilidad en la confidencialidad e integridad no garantizada.

### **1.3.3. Análisis causa – efecto**

#### **Causas:**

- Los inadecuados procesos actuales basados en el registro manual y la necesidad de comprobantes de seguridad aplicados a la gestión de recepción de documentos, organización de documentos y almacenamiento de toda la información académica de docentes y cursantes.

#### **Efectos:**

- Entrepapelado de documentación académica.
- Desbordamiento de papeles en los estantes.
- Confidencialidad pueda ser vulnerada y la integridad no sea garantizada
- Duplicidad de documentación.

## **1.4. OBJETIVOS Y ACCIONES**

### **1.4.1. Objetivo general**

Desarrollar una plataforma de gestión documental empleando técnicas criptográficas simétricas combinadas para reducir el riesgo en la vulnerabilidad de documentos, brindar una mejor organización de archivo y disminuir el tiempo empleado en la admisión y tratamiento de la documentación recepcionada.

### **1.4.2. Objetivos específicos y acciones**

- Diseñar el modelado de negocio actual y alternativo en base a características workflow.
- Desarrollar el módulo de registro de usuarios con roles y privilegios respectivos.
- Desarrollar el módulo de admisión de documentación docentes, cursantes y almacenamiento.

- Desarrollar el módulo de generación de código único para cada documento presentado.
- Desarrollar el módulo de inclusión de código QR a la documentación presentada.
- Generar reportes en base a módulos implementados.
- Realizar pruebas al sistema concluido.

Para el logro de los objetivos específicos del proyecto se deben seguir las siguientes acciones especificadas en la siguiente tabla.

**TABLA 1: Objetivos específicos y acciones**

OBJETIVOS ESPECÍFICOS	ACCIONES
Diseñar el modelado de negocio actual y alternativo en base a características workflow	<ul style="list-style-type: none"> <li>• Recolectar información referente a los procesos actuales de la gestión de información de la dirección de Posgrado</li> <li>• Modelar el flujo de información de los procesos actuales</li> <li>• Analizar el proceso de modelado actual.</li> <li>• Realizar el modelado de proceso actual</li> <li>• Identificar las deficiencias existentes en los procesos actuales</li> <li>• Identificar componentes de workflow o flujo de trabajo</li> <li>• Diseñar el flujo de trabajo utilizando las características de workflow</li> <li>• Elaborar el modelado de proceso alternativo</li> <li>• Seleccionar de metodología adecuada para el desarrollo</li> <li>• Planificar metodología en base a la metodología seleccionada</li> </ul>
Desarrollar el módulo de registro de usuarios con roles privilegios respectivos.	<ul style="list-style-type: none"> <li>• Seleccionar el entorno de desarrollo</li> <li>• Seleccionar de lenguaje de programación</li> <li>• Seleccionar de framework adecuado</li> <li>• Seleccionar de gestor de base de datos</li> <li>• Seleccionar de patrón de software</li> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar los diagramas UML del módulo de registro de usuarios</li> <li>• Diseñar la base de datos para los usuarios del sistema</li> <li>• Diseñar la arquitectura utilizando características de SOA</li> <li>• Implementar el módulo de registro de usuarios empleando herramientas para el desarrollo responsive</li> <li>• Realizar pruebas al módulo</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES
Desarrollar el módulo de admisión de documentación docente, cursante y almacenamiento	<ul style="list-style-type: none"> <li>• Analizar requerimientos del módulo de admisión de documentación</li> <li>• Analizar el tipo de documentación que será almacenada en formato digital mediante escaneo.</li> <li>• Seleccionar los documentos de acuerdo a requerimientos de la dirección de posgrado que serán almacenados.</li> <li>• Diseñar diagramas UML necesarios para el módulo de admisión de documentación.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el módulo para la admisión de documentación seleccionada de docentes y cursantes aplicando características de workflow.</li> <li>• Realizar pruebas al módulo</li> </ul>
Desarrollar el módulo de generación de código único para cada documento presentado	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Analizar algoritmos de generación de código único</li> <li>• Seleccionar el algoritmo de generación de código único</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el código para la generación de código único aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>
Desarrollar el módulo de inclusión de código QR a la documentación presentada.	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Diseñar servicios de la api Rest</li> <li>• Parametrizar contenido de código QR en base a mecanismos criptográficos</li> <li>• Implementar el código para generación de código QR aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES
Generar reportes en base a los módulos implementados	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para la elaboración de consultas.</li> <li>• Diseñar los diagramas UML necesarios</li> <li>• Diseñar servicios de la api Rest</li> <li>• Implementar el módulo de obtención de reportes en base a los requerimientos establecidos en el flujo de trabajo.</li> <li>• Generar la impresión de reportes en otros formatos</li> <li>• Realizar pruebas funcionales al módulo</li> </ul>
Realizar pruebas al sistema concluido.	<ul style="list-style-type: none"> <li>• Seleccionar los tipos de pruebas adecuados al proyecto.</li> <li>• Aplicar las pruebas seleccionadas</li> <li>• Documentar las pruebas realizadas.</li> </ul>

Fuente: *Elaboración Propia, 2019*

## 1.5. JUSTIFICACIÓN

### 1.5.1. Justificación técnica

Se justifica técnicamente debido a que el uso de herramientas criptográficas y de seguridad, brindaran mayor confidencialidad, autenticidad, archivado con garantía de integridad de toda la documentación académica entregada por docentes y cursantes, utilizar tecnologías de desarrollo responsive permitirán que la plataforma sea desplegada en dispositivos tales como: celulares inteligentes, tabletas electrónicas, ordenadores portátiles, ordenadores de escritorio, etc.

### 1.5.2. Justificación económica

El sistema propuesto no generará gastos económicos adicionales en cuando a hardware y software se refiere, debido a que el software implementado será libre y la dirección de posgrado cuenta con el hardware necesario, por otro lado la Dirección de Posgrado de la Escuela Militar de Ingeniería se beneficiará económicamente debido a que contará con una mejora significativa en los tiempos de proceso y aprovechará de mejor manera el uso de recursos humanos y

materiales, reducción de consumo de papel pues mejorar la situación del medio ambiente es tarea de todos y mucho mas para las casas de estudios superiores, para generar conciencia en toda la sociedad profesional del país.

### **1.5.3. Justificación operativa**

Se justifica operativamente debido a que el departamento de posgrado automatizará procesos manuales, se podrá llevar a cabo el registro de la documentación recepcionada, ademas de brindarle garantías de seguridad, integridad, confidencialidad y disponibilidad a través de mecanismos que corroboren que la documentación decepcionada fue revisada a cabalidad, la arquitectura del proyecto provee que se pueda ampliar a diferentes áreas de organización, el control de usuarios según sus roles permitirá que cada personal tenga accesos a funcionalidad en concreto, según su rol corresponda.

## **1.6. DELIMITACION DE LA INVESTIGACIÓN**

A continuación se definen los siguientes tipos de alcances:

### **1.6.1. Delimitación temática**

Las áreas utilizadas en el presente proyecto son análisis y diseño de sistemas, Ingeniería de software, gestores de bases de datos, sistema web, técnicas de recopilación de información, herramientas de programación y seguridad de sistemas, técnicas criptográficas, arquitectura orientada a servicios.

### **1.6.2. Delimitación Espacial**

El proyecto será diseñado e implementado en procesos de recepción de documentación académica de la dirección de posgrado de la EMI U.A. Cochabamba.

#### **1.6.3. Delimitación temporal**

El desarrollo del sistema propuesto está previsto durante la gestión 2019 - 2020 contando con 6 meses aproximadamente. El sistema tendrá un tiempo de vida útil de aproximadamente 5 años debido a los posibles nuevos requerimientos de la dirección de posgrado, puesto que esta es un área estratégica de la Escuela Militar de Ingeniería UA. Cochabamba la cual prevé que crezca exponencialmente a corto plazo.

#### **1.6.4. Delimitación funcional**

El sistema propuesto permitirá realizar el registro de usuarios, así como el almacenamiento de toda la documentación requerida para la apertura de programas en un mismo sistema, también permitirá realizar la asignación de códigos únicos de seguridad, ademas de códigos Qr basados en estos códigos únicos de seguridad, búsqueda oportuna y ágil de documentación y reportes, las características de la arquitectura orientada a servicios permite que el sistema pueda expandirse a otras áreas de la organización en este caso la EMI U.A. Cochabamba, otorgando la posibilidad de que dicho proyecto no solo se quede en el departamento de posgrado.

## CAPITULO 2: MARCO TEÓRICO

### 2. MARCO TEÓRICO

#### 2.1. CONTENIDO TEMÁTICO

TABLA 2: Contenido temático

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA
Diseñar el modelado de negocio actual y alternativo en base a características workflow	<ul style="list-style-type: none"><li>• Recolectar información referente a los procesos actuales de la gestión de información de la dirección de Posgrado</li><li>• Modelar el flujo de información de los procesos actuales</li><li>• Analizar el proceso de modelado actual.</li><li>• Realizar el modelado de proceso actual</li><li>• Identificar las deficiencias existentes en los procesos actuales</li><li>• Identificar componentes de workflow o flujo de trabajo</li><li>• Diseñar el flujo de trabajo utilizando las características de workflow</li><li>• Elaborar el modelado de proceso alternativo</li><li>• Seleccionar de metodología adecuada para el desarrollo</li><li>• Planificar metodología en base a la metodología seleccionada</li></ul>	<ul style="list-style-type: none"><li>• Técnicas de recolección de información</li><li>• Ingeniería de software</li><li>• Workflow</li></ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA
Desarrollar el módulo de registro de usuarios con roles privilegios respectivos.	<ul style="list-style-type: none"> <li>• Seleccionar el entorno de desarrollo</li> <li>• Seleccionar de lenguaje de programación</li> <li>• Seleccionar de framework adecuado</li> <li>• Seleccionar de gestor de base de datos</li> <li>• Seleccionar de patrón de software</li> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar los diagramas UML del módulo de registro de usuarios</li> <li>• Diseñar la base de datos para los usuarios del sistema</li> <li>• Diseñar la arquitectura utilizando características de SOA</li> <li>• Implementar el módulo de registro de usuarios empleando herramientas para el desarrollo responsive</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Técnicas de recolección de información</li> <li>• Tecnologías de desarrollo</li> <li>• Gestor de base de datos</li> <li>• Diseño responsivo web</li> </ul>
Desarrollar el módulo de admisión de documentación docente, cursante y almacenamiento	<ul style="list-style-type: none"> <li>• Analizar requerimientos del módulo de admisión de documentación</li> <li>• Analizar el tipo de documentación que será almacenada en formato digital mediante escaneo.</li> <li>• Seleccionar los documentos de acuerdo a requerimientos de la dirección de posgrado que serán almacenados.</li> <li>• Diseñar diagramas UML necesarios para el módulo de admisión de documentación.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar de servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el módulo para la admisión de documentación seleccionada de docentes y cursantes aplicando características de workflow.</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Técnicas de recolección de información</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Gestor de base de datos</li> <li>• Diseño responsivo web</li> <li>• Workflow</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA
Desarrollar el módulo de generación de código único para cada documento presentado	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Analizar algoritmos de generación de código único</li> <li>• Seleccionar el algoritmo de generación de código único</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el código para la generación de código único aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Seguridad de información</li> <li>• Gestor de base de datos</li> <li>• Diseño responsivo web</li> </ul>
Desarrollar el módulo de inclusión de código QR a la documentación presentada.	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Diseñar servicios de la api Rest</li> <li>• Parametrizar contenido de código QR en base a mecanismos criptográficos</li> <li>• Implementar el código para generación de código QR aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Seguridad de información</li> <li>• Diseño responsivo web</li> <li>• Workflow</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA
Generar reportes en base a los módulos implementados	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para la elaboración de consultas.</li> <li>• Diseñar los diagramas UML necesarios</li> <li>• Diseñar servicios de la api Rest</li> <li>• Implementar el módulo de obtención de reportes en base a los requerimientos establecidos en el flujo de trabajo.</li> <li>• Generar la impresión de reportes en otros formatos</li> <li>• Realizar pruebas funcionales al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Workflow</li> <li>• Diseño responsivo web</li> </ul>
Realizar pruebas al sistema concluido.	<ul style="list-style-type: none"> <li>• Seleccionar los tipos de pruebas adecuados al proyecto.</li> <li>• Aplicar las pruebas seleccionadas</li> <li>• Documentar las pruebas realizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas e instrumentos de recolección de datos</li> <li>• Ingeniería de software</li> </ul>

Fuente: *Elaboración Propia, 2019*

## 2.2. TÉCNICAS DE RECOPILACIÓN DE INFORMACIÓN

Existen tres métodos interactivos clave que puede usar para obtener los requerimientos humanos de información de los miembros de la organización: entrevistas, encuestas aplicadas a las personas mediante cuestionarios y la observación. Aunque son distintos en su implementación, estos métodos tienen muchas cosas en común. La base de sus propiedades compartidas es hablar con las personas en la organización y escucharlas para comprender sus interacciones con la tecnología, a través de una serie de preguntas cuidadosamente elaboradas. Cada uno de los tres métodos interactivos para recopilar información cuenta con su propio proceso establecido; el cual se debe seguir para interactuar con los usuarios. (Kendall y Kendall, 2011)

Un aspecto muy importante en el proceso de una investigación es el que tiene relación con la obtención de la información, pues de ello dependen la confiabilidad y validez de estudio. Obtener información confiable y valida requiere cuidado y dedicación.

Esta etapa de recolección de información e investigación se conoce también como trabajo de campo (Bernal, 2006).

### **2.2.1. Entrevistas**

Una entrevista para recopilar información es una conversación dirigida con un propósito específico, en la cual se usa un formato de preguntas y respuestas. En la entrevista hay que obtener las opiniones del entrevistado y lo que siente sobre el estado actual del sistema, los objetivos de la organización, el personal, y los procedimientos informales para interactuar con las tecnologías de la información. (Kendall, 2011)

Existen cinco pasos para realizar la preparación de una entrevista:

- Leer y comprender el material sobre los antecedentes de la organización.
- Establecer los objetivos de la entrevista en base a los antecedentes.
- Decidir a quién entrevistar, eligiendo personas clave en la organización.
- Preparar al entrevistado.
- Decidir sobre los tipos de preguntas y su estructura, redactar preguntas que puedan cubrir las áreas más importantes de la organización.

### **2.2.2. Cuestionarios**

El cuestionario es un instrumento muy popular como recurso de investigación, es un sistema de preguntas racionales, ordenadas de forma coherente, expresadas en un lenguaje sencillo y comprensible, que generalmente responde por escrito la persona interrogada sin que sea necesaria la intervención del encuestador. El cuestionario permite la recolección de datos provenientes de fuentes primarias, es decir, de personas que poseen la información que resulta de interés. (Cordova, 2004)

### **2.2.3. Observación**

La observación es el método más directo, útil y antiguo de obtención de información.

Como técnica de investigación, tiene amplia aceptación científica, ésta técnica se utiliza extensamente con el fin de estudiar a las personas en sus actividades de grupo y como miembros de la organización. El propósito de la observación es múltiple: permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuanto tiempo toma, dónde se hace y por qué se hace.

Observar no es simplemente “mirar”. El investigador al observar busca respuestas a interrogantes, responderse preguntas. Ayuda a confirmar o refutar la información que se ha obtenido por medio de las entrevistas, los cuestionarios, análisis de documentos u otros medios, etc. (Encinoza, 2008)

## **2.3. INGENIERÍA DE SOFTWARE**

### **2.3.1. Desarrollo ágil**

Cualquier proceso del software ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave acerca de la mayoría de proyectos de software:

Es difícil predecir qué requerimientos de software persistirán y cuáles cambiarán. También es difícil pronosticar cómo cambiarán las prioridades del cliente a medida que avanza el proyecto.

Para muchos tipos de software, el diseño y la construcción están imbricados. Es decir, ambas actividades deben ejecutarse en forma simultánea, de modo que los modelos de diseño se prueben a medida que se crean. Es difícil predecir cuánto diseño se necesita antes de que se use la construcción para probar el diseño.

El análisis, el diseño, la construcción y las pruebas no son tan predecibles como

gustaría (desde un punto de vista de planeación).

Dadas estas tres suposiciones, surge una pregunta importante: ¿cómo crear un proceso que pueda manejar lo impredecible? La respuesta, como ya se dijo, está en la adaptabilidad del proceso (al cambio rápido del proyecto y a las condiciones técnicas). Por tanto, un proceso ágil debe ser adaptable. (Pressman, 2010)

Pero la adaptación continua logra muy poco si no hay avance. Entonces, un proceso de software ágil debe adaptarse incrementalmente. Para lograr la adaptación incremental, un equipo ágil requiere retroalimentación con el cliente (de modo que sea posible hacer las adaptaciones apropiadas). Un catalizador eficaz para la retroalimentación con el cliente es un prototipo operativo o una porción de un sistema operativo. Así, debe instituirse una estrategia de desarrollo incremental. Deben entregarse incrementos de software (prototipos ejecutables o porciones de un sistema operativo) en periodos cortos de tiempo, de modo que la adaptación vaya a ritmo con el cambio (impredecible). Este enfoque iterativo permite que el cliente evalúe en forma regular el incremento de software, dé la retroalimentación necesaria al equipo de software e influya en las adaptaciones del proceso que se realicen para aprovechar la retroalimentación. (Pressman, 2010)

### **2.3.2. Política de desarrollo ágil**

Hay mucho debate (a veces estridente) sobre los beneficios y aplicabilidad del desarrollo de software ágil como oposición a los procesos más convencionales. Los extremos cuando caracteriza la posición del campo a favor de la agilidad (“agilistas”). “Los metodólogos tradicionales están atrapados en un pantano y producirán una documentación sin defectos en vez de un sistema funcional que satisfaga las necesidades del negocio.” Como contrapunto, plantea (de nuevo como burla) la posición del campo de la ingeniería de software tradicional: “Los metodólogos ligeros, perdón, ‘ágiles’, son un grupo de remendones famosos que se van a llevar una sorpresa cuando intenten convertir sus juguetes en software a la medida de la empresa.”

Como todos los argumentos sobre la tecnología de software, este debate sobre la metodología corre el riesgo de degenerar en una guerra religiosa. Si estalla, desaparece el pensamiento racional y lo que guía la toma de decisiones son las creencias y no los hechos.

Nadie está contra la agilidad. La pregunta real es: ¿cuál es la mejor forma de lograrla? De igual importancia: ¿cómo construir software que satisfaga en el momento las necesidades de los clientes y que tenga características de calidad que permitan ampliarlo y escalarlo para que también las satisfaga en el largo plazo?

No hay respuestas absolutas a ninguna de estas preguntas. Aun dentro de la escuela ágil hay muchos modelos de proceso propuestos, cada uno con un enfoque algo diferente para el problema de la agilidad. Dentro de cada modelo hay un conjunto de “ideas” (los agilistas las llaman “tareas del trabajo”) que representan un alejamiento significativo de la ingeniería de software tradicional. No obstante, muchos conceptos agiles sólo son adaptaciones de algunos que provienen de la buena ingeniería de software. En resumen: hay mucho por ganar si se considera lo mejor de ambas escuelas, y virtualmente no se gana nada si se denigra cualquiera de los enfoques. (Pressman, 2010)

### **2.3.3. Programación Extrema**

A fin de ilustrar un proceso ágil con más detalle, se dará un panorama de la programación extrema (XP), el enfoque más utilizado del desarrollo de software ágil. Aunque las primeras actividades con las ideas y los métodos asociados a XP ocurrieron al final de la década de 1980. Una variante de XP llamada XP industrial [IXP] se propuso en una época más reciente IXP mejora la XP y tiene como objetivo el proceso ágil para ser usado específicamente en organizaciones grandes. (Pressman, 2010)

### **2.3.3.1. Valores XP**

Se define un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP. (Pressman, 2010)

Para alcanzar la simplicidad, XP restringe a los desarrolladores para que diseñen sólo para las necesidades inmediatas, en lugar de considerar las del futuro. El objetivo es crear un diseño sencillo que se implemente con facilidad.

La retroalimentación se obtiene de tres fuentes: el software implementado, el cliente y otros miembros del equipo de software. Al diseñar e implementar una estrategia de pruebas eficaz el software (por medio de los resultados de las pruebas) da retroalimentación al equipo ágil. XP usa la prueba unitaria como su táctica principal de pruebas. A medida que se desarrolla cada clase, el equipo implementa una prueba unitaria para ejecutar cada operación de acuerdo con su funcionalidad especificada. Cuando se entrega un incremento a un cliente, las historias del usuario o casos de uso (véase el capítulo 5) que se implementan con el incremento se utilizan como base para las pruebas de aceptación. El grado en el que el software implementa la salida, función y comportamiento del caso de uso es una forma de retroalimentación. Por último, conforme se obtienen nuevos requerimientos como parte de la planeación iterativa, el equipo da al cliente una retroalimentación rápida con miras al costo y al efecto en la programación de actividades. (Pressman, 2010)

Beck afirma que la adhesión estricta a ciertas prácticas de XP requiere valentía. Un término más apropiado sería disciplina. Por ejemplo, es frecuente que haya mucha presión para diseñar requerimientos futuros. La mayor parte de equipos de software sucumben a ella y se justifican porque “diseñar para el mañana” ahorrará tiempo y esfuerzo en el largo plazo. Un equipo XP ágil debe tener la disciplina (valentía) para diseñar para hoy y reconocer que los requerimientos futuros tal vez cambien mucho, por lo que demandarán repeticiones sustanciales del diseño y del

código implementado.

Al apegarse a cada uno de estos valores, el equipo ágil inculca respeto entre sus miembros, entre otros participantes y los integrantes del equipo, e indirectamente para el software en sí mismo. Conforme logra la entrega exitosa de incrementos de software, el equipo desarrolla más respeto para el proceso XP. (Pressman, 2010)

La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. (Pressman, 2010)

#### **2.3.3.2. Planeación**

La actividad de planeación (también llamada juego de planeación) comienza escuchando actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieren. Escuchar lleva a la creación de algunas “historias” (también llamadas historias del usuario) que describen la salida necesaria, características y funcionalidad del software que se va a elaborar. Cada historia es escrita por el cliente y colocada en una tarjeta indizada. El cliente asigna un valor (es decir, una prioridad) a la historia con base en el valor general de la característica o función para el negocio. Después, los miembros del equipo XP evalúan cada historia y le asignan un costo, medido en semanas de desarrollo. Si se estima que la historia requiere más de tres semanas de desarrollo, se pide al cliente que la descomponga en historias más chicas y de nuevo se asigna un valor y costo. Es importante observar que en cualquier momento es posible escribir nuevas historias. (Pressman, 2010)

Los clientes y desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega (el siguiente incremento de software) que

desarrollará el equipo XP. Una vez que se llega a un compromiso sobre la entrega (acuerdo sobre las historias por incluir, la fecha de entrega y otros aspectos del proyecto), el equipo XP ordena las historias que serán desarrolladas en una de tres formas: 1) todas las historias se implementarán de inmediato (en pocas semanas), 2) las historias con más valor entrarán a la programación de actividades y se implementarán en primer lugar o 3) las historias más riesgosas formarán parte de la programación de actividades y se implementarán primero.

Después de la primera entrega del proyecto (también llamada incremento de software), el equipo XP calcula la velocidad de éste. En pocas palabras, la velocidad del proyecto es el número de historias de los clientes implementadas durante la primera entrega. La velocidad del proyecto se usa para: 1) ayudar a estimar las fechas de entrega y programar las actividades para las entregas posteriores, y 2) determinar si se ha hecho un gran compromiso para todas las historias durante todo el desarrollo del proyecto. Si esto ocurre, se modifica el contenido de las entregas o se cambian las fechas de entrega final. (Pressman, 2010)

A medida que avanza el trabajo, el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas. Entonces, el equipo XP reconsidera todas las entregas faltantes y modifica sus planes en consecuencia. (Pressman, 2010)

**a) Ventajas**

- Da lugar a una programación sumamente organizada.
- Ocasiona eficiencias en el proceso de planificación y pruebas.
- Cuenta con una tasa de errores muy pequeña.
- Propicia la satisfacción del programador.
- Fomenta la comunicación entre los clientes y los desarrolladores.

- Facilita los cambios.
- Permite ahorrar mucho tiempo y dinero.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.
- La XP es mejor utilizada en la implementación de nuevas tecnologías (Pressman, 2010)

**b) Desventajas**

- Es recomendable emplearla solo en proyectos a corto plazo.
- En caso de fallar, las comisiones son muy altas.
- Requiere de un rígido ajuste a los principios de XP.
- Puede no siempre ser más fácil que el desarrollo tradicional. (Pressman, 2010)

**2.3.3. Scrum**

Scrum (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby) es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990.

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso (que se estudia en el párrafo siguiente) llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto)

se adapta al problema en cuestión y se define y con frecuencia se modifica en tiempo real por parte del equipo Scrum. (Pressman, 2010)

### **2.3.3.1. Principios Scrum**

Scrum acentúa el uso de un conjunto de patrones de proceso del software [que han demostrado ser eficaces para proyectos con plazos de entrega muy apretados, requerimientos cambiantes y negocios críticos. Cada uno de estos patrones de proceso define un grupo de acciones de desarrollo:

Retraso: lista de prioridades de los requerimientos o características del proyecto que dan al cliente un valor del negocio. Es posible agregar en cualquier momento otros aspectos al retraso (ésta es la forma en la que se introducen los cambios). El gerente del proyecto evalúa el retraso y actualiza las prioridades según se requiera.

Sprints: consiste en unidades de trabajo que se necesitan para alcanzar un requerimiento definido en el retraso que debe ajustarse en una caja de tiempo predefinida (lo común son 30 días). Durante el sprint no se introducen cambios (por ejemplo, aspectos del trabajo retrasado). Así, el sprint permite a los miembros del equipo trabajar en un ambiente de corto plazo pero estable. (Pressman, 2010)

Reuniones Scrum: son reuniones breves (de 15 minutos, por lo general) que el equipo Scrum efectúa a diario. Hay tres preguntas clave que se pide que respondan todos los miembros del equipo :

¿Qué hiciste desde la última reunión del equipo?

¿Qué obstáculos estás encontrando?

¿Qué planeas hacer mientras llega la siguiente reunión del equipo?

Un líder del equipo, llamado maestro Scrum, dirige la junta y evalúa las respuestas de cada persona. La junta Scrum ayuda al equipo a descubrir los problemas

potenciales tan pronto como sea posible. Asimismo, estas juntas diarias llevan a la “socialización del conocimiento”, con lo que se promueve una estructura de equipo con organización propia.

Demostraciones preliminares: entregar el incremento de software al cliente de modo que la funcionalidad que se haya implementado pueda demostrarse al cliente y éste pueda evaluarla.

Es importante notar que las demostraciones preliminares no contienen toda la funcionalidad planeada, sino que éstas se entregarán dentro de la caja de tiempo establecida. (Pressman, 2010)

Beedle y sus colegas presentan un análisis exhaustivo de estos patrones en el que dicen: “Scrum supone de entrada la existencia de caos...” Los patrones de proceso Scrum permiten que un equipo de software trabaje con éxito en un mundo en el que es imposible eliminar la incertidumbre. (Pressman, 2010)

### **2.3.3.2. Eventos Scrum**

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar las vitales transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse. (Pressman, 2010)

**a) Ventajas**

- Los clientes pueden participar en cada una de las iteraciones y proponer soluciones. De hecho, el proceso en su conjunto está pensado para un tipo de evaluación conjunta.
- Cada iteración arroja una serie de resultados. No es necesario, por tanto, que el cliente espere hasta el final para ver el producto.
- Se adapta a cualquier contexto, área o sector de la gestión. No es una técnica exclusiva de ninguna disciplina.
- Del mismo modo, los riesgos que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición. La intervención de los equipos de trabajo es inmediata.

**b) Desventajas**

- No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia. (Pressman, 2010)

**2.3.3.3. Kanban**

Kanban, llega como metodología de gestión de proyectos, de la mano de la automotriz Toyota, representando estadísticamente, la metodología ágil que menor resistencia presenta en las compañías acostumbradas a las metodologías tradicionales.

La palabra Kanban, de origen japonés, se compone de dos términos: Kan que puede traducirse como "visual" y ban, como "insignia", siendo una traducción aproximada, "insignia visual". (Fuentes C, 2011)

Kanban se basa en un sistema de producción que dispara trabajo solo cuando existe capacidad para procesarlo. El disparador de trabajo es representado por tarjetas kanban de las cuales se dispone de una cantidad limitada.

Cada tarjeta Kanban acompaña a un ítem de trabajo durante todo el proceso de producción, hasta que éste, es empujado fuera del sistema, liberando una tarjeta. Un nuevo ítem de trabajo, solo podrá ser ingresado/aceptado si se dispone de una tarjeta kanban libre.

Este proceso de producción, donde un trabajo se introduce al sistema solo cuando existe disponibilidad para procesarlo, se denomina pull (tirar) en contrapartida al mecanismo push (empujar), donde el trabajo se introduce en función de la demanda.

En el desarrollo de Software, Kanban fue introducido por David Anderson de la Unidad de Negocios XIT de Microsoft, en 2004, reemplazando los sistemas de tarjetas por un tablero visual similar al de Scrum, pero con características extendidas que se mencionan a continuación.

Con tan solo tres simples reglas, Kanban demuestra ser una de las metodologías adaptativas que menos resistencia al cambio presenta. Dichas reglas son:

1. Mostrar el proceso
2. Limitar el trabajo en curso
3. Optimizar el flujo de trabajo (Fuentes C, 2011)

a) **Regla 1 Mostrar el proceso**

Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero físico, generalmente, públicamente asequible. El objetivo de mostrar el proceso, consiste en:

Entender mejor el proceso de trabajo actual;

Conocer los problemas que puedan surgir y tomar decisiones;

Mejorar la comunicación entre todos los interesados/participantes del proyecto;  
Hacer los futuros procesos más predecibles. (Fuentes C, 2011)

Un tablero Kanban, se divide en columnas las cuales representan un proceso de trabajo. Un ejemplo clásico de columnas para dividir un tablero Kanban, sería el siguiente:

Cola de entrada | Análisis | Desarrollo | Test | Deploy | Producción

La cantidad y nombre de las columnas, varía de acuerdo a las necesidades de cada equipo y en la mayoría de los casos, éstas, son subdivididas en dos columnas: cola de espera y en curso (Fuentes C, 2011)

**b) Regla 2 Limitar el trabajo**

Los límites del WIP (work in progress trabajo en curso) consisten en acordar anticipadamente, la cantidad de ítems que pueden abordarse por cada proceso (es decir, por columnas del tablero).

El principal objetivo de establecer estos límites, es el de detectar cuellos de botella.

Los cuellos de botella representan el estancamiento de un proceso determinado.  
Viendo el siguiente tablero ficticio, se puede comprender mejor:

**c) Regla 3 Optimizar el flujo**

El objetivo una la producción estable, continua y previsible. Midiendo el tiempo que el ciclo completo de ejecución del proyecto demanda, se obtiene el CycleTime.

Al dividir, el CycleTime por el WIP, se obtiene el "rendimiento de trabajo", denominado Throughput, es decir, la cantidad de ítems que un equipo puede terminar en un determinado período de tiempo.

Throughput = CycleTime/WIP

Con estos valores, la optimización del flujo de trabajo consistirá en la búsqueda de:

Minimizar el CycleTime

Maximizar el Throughput

Lograr una variabilidad mínima entre CycleTime y Throughput (Fuentes C, 2011)

**d) Ventajas**

- Facilidad de entendimiento y exposición de información a todos los miembros involucrados.
- Facilidad de integración con metodologías ágiles (Scrumban).
- El más adecuado para proyectos que se encuentran en mejora continua.
- No todas las desventajas de Kanban surgidas en manufactura aplican en la industria del software puesto que la implementación interna es diferente.
- Acepta el ingreso de cambios a último momento con facilidad.

**e) Desventajas**

- Menor efectividad en situaciones de Recursos Compartidos: Las órdenes no frecuentes vuelven inefficientes a kanban ya que se tiene que asegurar una producción suficiente por parte de un proceso mientras que a su vez el proceso que es no-frecuente es ejecutado.
- Kanban asume sistemas de producción repetitivos dada la naturaleza de su creación en el área de manufactura.
- Posiblemente Kanban pueda arrojar productos de baja calidad que requieren de ser retrabajados. Kanban funciona a manera de semáforo para administrar el tráfico y así cumplir con las necesidades del cliente indicando cuando empezar, cuando bajar el ritmo y cuando parar. Cualquier variabilidad o evento no esperado puede afectar el funcionamiento del sistema provocando que se generen señales confusas.
- Calidad: El sistema Kanban lleva los niveles de inventario cerca de 0 lo que en caso de alta incertidumbre e interrupciones en la red de transporte representa un peligro ya que eso significa que los clientes se quedan sin suministro de partes. (Fuentes C, 2011)

#### **2.3.4. Lenguaje unificado de desarrollo (UML)**

Un modelo es una representación de la realidad, y en este caso el modelado es una parte central de todas las actividades que conducen a la conducción de un buen software. El modelo se construye para comunicar la estructura deseada y el comportamiento del sistema, para visualizar y controlar la arquitectura del sistema, así tener mayor comprensión del sistema que se está construyendo. (Booch, 2006)

Un modelo proporciona los planos de un sistema. Estos modelos pueden involucrar planos detallados, así como planos más generales que ofrecen una visión global del sistema en consideración. (Booch, 2006)

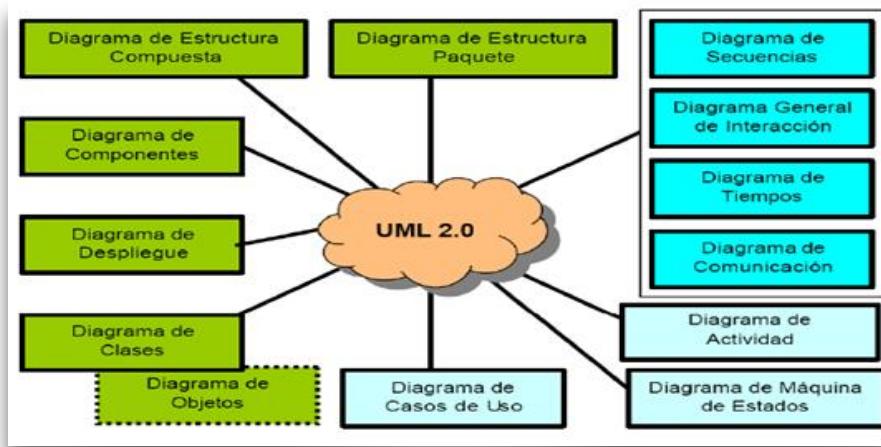
##### **2.3.4.1. Bloques básicos de UML**

El vocabulario de uml incluye tres clases de bloques básicos:

- **Elementos:** existen cuatro tipos, estructurales, de comportamiento, de agrupación y anotación.
- **Relaciones:** existen cuatro tipos de relaciones, dependencia, asociación, generalización y realización.
- **Diagramas:** un diagrama es la representación gráfica de un conjunto de elementos, se dibujan para visualizar un sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema. (Booch, 2006)

En la figura 1 se muestran los diagramas para desarrollar UML

**FIGURA 1: Diagramas UML 2.0**

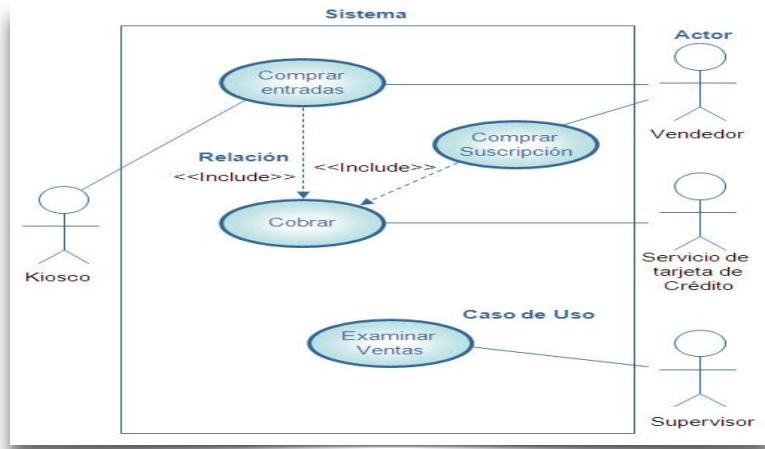


Fuente: (Booch, 2006)

#### 2.3.4.2. Diagrama de caso de uso

Estos diagramas muestran un conjunto de casos de uso, actores y sus relaciones. Cubren la vista de casos de uso estática de un sistema. Estos diagramas son especialmente importantes en el modelado del comportamiento de un sistema. (Booch, 2006)

**FIGURA 2: Diagrama de caso de uso**



Fuente: (Booch, 2006)

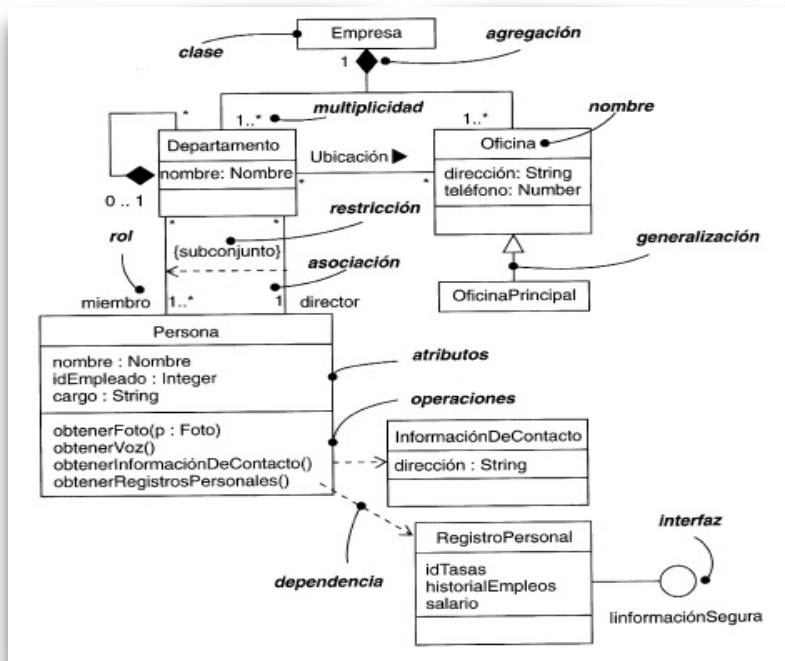
#### 2.3.4.3. Diagrama de clases

Es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos),

y las relaciones entre los objetos. Estos diagramas son los más comunes en el modelado de sistemas orientado a objetos. Los diagramas de clases abarcan la vista de diseño estática de un sistema. (Booch, 2006)

Los diagramas de clases que incluyen clases activas cubren la vista de procesos estática de un sistema. Los diagramas de componentes son una variante de los diagramas de clases. (Booch, 2006)

**FIGURA 3: Diagrama de clases**



Fuente: (Booch, 2006)

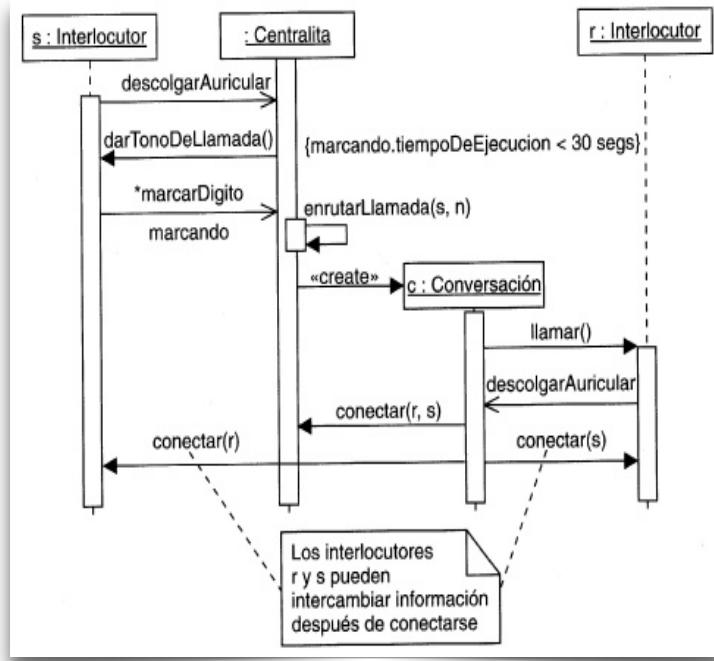
#### 2.3.4.4. Diagrama de secuencias

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. (Booch, 2006)

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios

para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales. (Booch, 2006)

**FIGURA 4: Diagrama de secuencia**

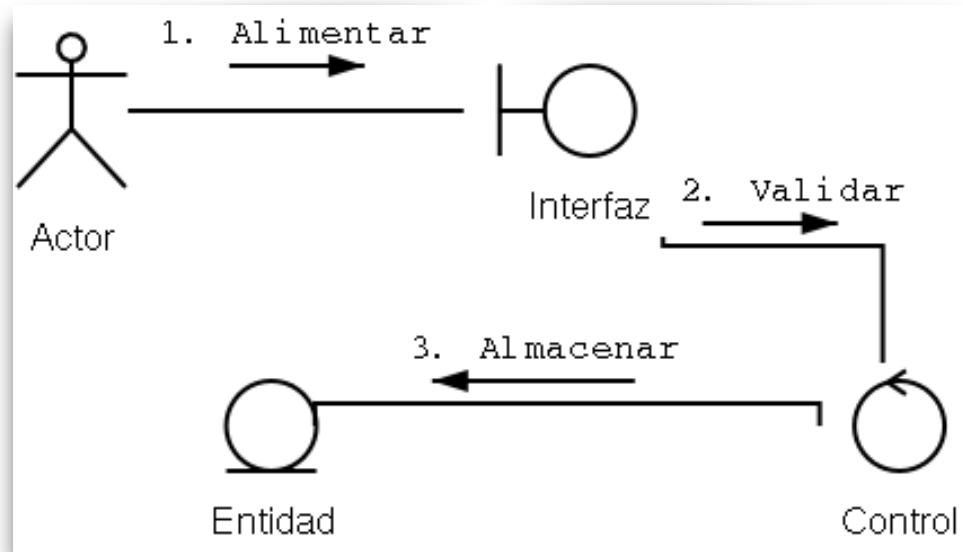


**Fuente:** (Booch, 2006)

#### 2.3.4.5. Diagrama de colaboración

El diagrama de colaboraciones describe las interacciones entre los objetos en términos de mensajes secuenciados. Los diagramas de colaboración representan una combinación de información tomada de los diagramas de clases, de secuencias y de casos de uso, describiendo el comportamiento, tanto de la estructura estática, como de la estructura dinámica de un sistema. (Boch, 2006)

**FIGURA 5: Diagrama de colaboración**



Fuente: (Booch, 2006)

### 2.3.5. Pruebas de software

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo. (Pressman, 2010)

#### 2.3.5.1. Pruebas de funcionalidad

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. (Pressman, 2010)

Las pruebas de funcionalidad se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Dicho de otro modo son pruebas específicas, concretas y

exhaustivas para probar y validar que el software hace lo que debe y sobre todo, lo que se ha especificado. (Pressman, 2010)

**a) Objetivos de las pruebas de funcionalidad**

El objetivo de estas pruebas es asegurar el trabajo apropiado de los requisitos funcionales incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Para poder cumplir con las metas de:

- Verificar el procesamiento, recuperación e implementación adecuada de las reglas de negocio.
- Verificar la apropiada aceptación de datos.

**2.3.5.2. Pruebas de caja negra**

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos o en el acceso a bases de datos externas. Errores de comportamiento o rendimiento.
- Errores de inicialización y terminación.

Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba. Las

herramientas básicas son observar la funcionalidad y contrastar con la especificación. (Pressman, 2010)

#### **2.3.5.3. Pruebas de integración**

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. (Pressman, 2010)

La integración incremental es la antítesis del enfoque big bang. El programa se construye y prueba en pequeños incrementos, donde los errores son más fáciles de aislar y corregir; las interfaces tienen más posibilidades de probarse por completo; y puede aplicarse un enfoque de prueba sistemático.

En los siguientes párrafos se exponen algunas estrategias diferentes de integración incremental: (Pressman, 2010)

- a) **Integración descendente.** La prueba de integración descendente es un enfoque incremental a la construcción de la arquitectura de software. Los módulos se integran al moverse hacia abajo a través de la jerarquía de control, comenzando con el módulo de control principal (programa principal).
- b) **Integración ascendente.** La prueba de integración ascendente, como su nombre implica, comienza la construcción y la prueba con módulos atómicos (es decir, componentes en los niveles inferiores dentro de la estructura del programa). Puesto que los componentes se integran de abajo hacia arriba, la funcionalidad que proporcionan los componentes subordinados en determinado nivel siempre está disponible y se elimina la necesidad de representantes

#### **2.3.5.4. Pruebas de usabilidad**

La prueba de usabilidad evalúa el grado en el cual los usuarios pueden interactuar efectivamente con la webapp y el grado en el que la webapp guía las acciones del usuario, proporciona retroalimentación significativa y refuerza un enfoque de interacción consistente. En lugar de enfocarse atentamente en la semántica de algún objetivo interactivo, las revisiones y pruebas de usabilidad se diseñan para determinar el grado en el cual la interfaz de la webapp facilita la vida del usuario. (Pressman, 2010)

Invariablemente, el ingeniero en software contribuirá con el diseño de las pruebas de usabilidad, pero las pruebas en sí las realizan los usuarios finales. Existe una secuencia de pasos para poder realizar las pruebas de usabilidad las cuales son: (Pressman, 2010)

- Definir un conjunto de categorías de prueba de usabilidad e identificar las metas de cada una.
- Diseñar pruebas que permitirán la evaluación de cada meta.
- Seleccionar a los participantes que realicen las pruebas.
- Instrumentar la interacción de los participantes con la webapp mientras se lleva a cabo la prueba.
- Desarrollar un mecanismo para valorar la usabilidad de la webapp.

#### **2.3.5.5. Pruebas de seguridad**

Las pruebas de seguridad buscan medir la Confidencialidad, Integridad y Disponibilidad de los datos, desde la perspectiva del aplicativo, es decir partiendo a identificar amenazas y riesgos desde el uso o interface de usuario final. Una vez ejecutadas las pruebas de seguridad es posible medir y cuantificar los riesgos a los cuales se ven expuestos los aplicativos tanto en la infraestructura interna como externa. (Pressman, 2010)

Las pruebas de seguridad se diseñan para sondear las vulnerabilidades del entorno lado cliente, las comunicaciones de red que ocurren conforme los datos pasan de cliente a servidor y viceversa, y el entorno del lado servidor. Cada uno de estos dominios puede atacarse, y es tarea del examinador de seguridad descubrir las debilidades que puedan explotar quienes tengan intención de hacerlo. (Pressman, 2010)

Para proteger el sistema de las vulnerabilidades, se implanta uno o más de los siguientes elementos de seguridad:

- **Firewall:** mecanismo de filtrado, que es una combinación de hardware y software que examina cada paquete de información entrante para asegurarse de que proviene de una fuente legítima y que bloquea cualquier dato sospechoso.
- **Autenticación:** mecanismo de verificación que valida la identidad de todos los clientes y servidores, y permite que la comunicación ocurra solamente cuando ambos lados se verifican.
- **Encriptado:** mecanismo de codificación que protege los datos sensibles al modificarlos de forma que hace imposible leerlos por quienes tienen intenciones maliciosas. El encriptado se fortalece usando certificados digitales que permiten al cliente verificar el destino al que se transmiten los datos.
- **Autorización:** mecanismo de filtrado que permite el acceso al entorno cliente o servidor sólo a aquellos individuos con códigos de autorización apropiados (por ejemplo, ID de usuario y contraseña).

Las pruebas de seguridad deben diseñarse para sondear cada una de estas tecnologías de seguridad con la intención de descubrir huecos en la seguridad. (Pressman, 2010)

### **2.3.5.5. Pruebas de aceptación**

Las pruebas de aceptación al igual que las de sistema, se realizan sobre el producto terminado e integrado; pero a diferencia de aquellas, están concebidas para que sea un usuario final quien detecte los posibles errores. (Fontela, 2003)

Se clasifican en dos:

- Pruebas alfa
- Pruebas beta

#### **a) Pruebas alfa**

Las pruebas alfa se realizan por un cliente en un entorno controlado por el equipo de desarrollo. Para que tengan validez, se debe primero crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas y a documentar los resultados.

Cuando el software sea la adaptación de una versión previa, deberán probarse también los procesos de transformación de datos y actualización de archivos de todo tipo. (Fontela, 2003)

#### **b) Pruebas beta**

Las pruebas beta se realizan en las instalaciones propias de los clientes. Para que tengan lugar, en primer término se deben distribuir copias del sistema para que cada cliente lo instale en sus oficinas, dependencias y/o sucursales, según sea el caso. Si se tratase de un número reducido de clientes, el tema de la distribución de las copias no representa grandes dificultades, pero en el caso de productos de venta masiva, la elección de los beta testers debe realizarse con sumo cuidado. En el caso de las pruebas Beta, cada usuario realizará sus propias pruebas y documentará los errores que encuentre, así como las sugerencias que crea conveniente realizar, para que el equipo de desarrollo tenga en cuenta al momento de analizar las posibles modificaciones. (Fontela, 2003)

Cuando el sistema tenga un cliente individual, las pruebas de aceptación se hacen de común acuerdo con éste, y los usuarios se determinan en forma programada, así como también se definen los aspectos a probar y la forma de informar resultados. Cuando, en cambio, se está desarrollando un producto masivo, los usuarios para pruebas se determinan de formas menos estrictas, y hay que ser muy cuidadoso en la evaluación del feedback que proveen. Por lo tanto, en este segundo caso hay que dedicar un esfuerzo considerable a la planificación de las pruebas de aceptación. (Fontela, 2003)

## **2.4. ARQUITECTURA ORIENTADA A SERVICIOS**

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular, pero no es la única forma de realizar la implementación de SOA. (Microsoft, 2006)

### **2.4.1. Beneficios**

Los beneficios de SOA para una organización se plasman a dos niveles distintos: al del usuario corporativo y a nivel de la organización de IT. Desde el punto de vista de la empresa, SOA permite el desarrollo de una nueva generación de aplicaciones dinámicas que resuelven una gran cantidad de problemas de alto nivel, fundamentales para el crecimiento y la competitividad. (Microsoft, 2006)

Algunos de los principales beneficios que se obtiene al implementar una arquitectura SOA son: (Galindo, 2016)

- Agilidad para habilitar rápidamente soluciones innovadoras
- Flexibilidad para reducir los tiempos y costos de implantación

- Rapidez para llegar primero al mercado antes que la competencia
- Obtener mejor visibilidad de la información a través de toda su organización.
- Optimiza sus procesos de negocios.
- Fácil integración con sistemas paralelos.
- Mejor integración de nuevas funcionalidades al sistema.

Otra de sus ventajas es la independencia de las plataformas e infraestructuras tecnológicas, lo que le permite integrarse con sistemas y aplicaciones diferentes de forma sencilla. Gracias a esta independencia SOA es su arquitectura flexible que permite la reutilización de las tecnologías existentes. Así que, una empresa no necesita realizar un cambio integral para adoptar SOA. (Galindo, 2016)

#### **2.4.2. Características**

SOA es un estilo arquitectónico para la construcción de aplicaciones de software en base a servicios disponibles. Entre sus principales características destacan: (Galindo, 2016)

- Su flexibilidad, que permite la reutilización.
- Su versatilidad, que hace posible que los servicios puedan ser consumidos.
- Sus posibilidades, que optimizan el trabajo con datos y su coordinación.

#### **2.4.3. Componentes de la arquitectura orientada a servicios**

Esta arquitectura presenta un modelo de construcción sistemas distribuidos en el que la funcionalidad demandada será entregada a la aplicación a través de servicios. (IDS-IBM, 2015)

En la figura 11, se muestra el esquema de la arquitectura y los componentes que podrían contener la arquitectura. Como puede observarse, el esquema se encuentra dividido en 2 zonas, una que abarca el ámbito funcional de la arquitectura y otra vinculada a la calidad de servicio. (IDS-IBM, 2015)

#### **2.4.4. Funcionamiento de SOA**

SOA propone un nuevo modelo de estructuración, orientado a atacar la definición de arquitecturas en vez de la definición de aplicaciones individuales. Desarrollar procesos de negocio como servicios es la clave de la flexibilidad de la arquitectura, esto permite que otros componentes funcionales hagan uso de diferentes servicios de manera natural con independencia de su ubicación. (IDS-IBM, 2015)

El diseño que sigue los parámetros de esta arquitectura es fácilmente escalable, sobre ella se pueden añadir futuros servicios y permite que cada Servicio evolucione de manera independiente. (IDS-IBM, 2015)

SOA plantea aplicaciones distribuidas, multi-nivel con capas de presentación, lógica de negocios y persistencia. Propone varias capas de servicios que exponen funcionalidades de negocio que a su vez permiten la composición de aplicaciones a partir de los mismos. (IDS-IBM, 2015)

A la hora de desarrollar, en lugar de crear aplicaciones enormes y muy complejas, se desarrollan componentes reutilizables, que son fáciles de mantener y probar. Las aplicaciones se crean diseñando un proceso que interactúa con estos servicios, donde se reutilizan los componentes existentes y sólo se desarrollan aquellos componentes no implementados. (IDS-IBM, 2015)

Para SOA es indiferente la tecnología que se utilice, sin embargo, la tecnología de Web Services o Servicios Web es una de las que más notoriedad ha conseguido. En muchos casos y contextos SOA y Servicios Web es tratado como un mismo concepto, pero Web Services no es sinónimo de SOA. Por el contrario, es posible utilizar Web Services y seguir un esquema de integración punto a punto, así como es posible implantar un esquema SOA sin utilizar Web Services. Sin embargo, tanto Web Services como los estándares asociados (XML, SOAP, WSDL, UDDI, WS-I) conforman una base de tecnologías y estándares que facilitan la implementación de SOA. (IDS-IBM, 2015)

#### **2.4.5. Capas de software**

SOA define las siguientes capas de software: (Microsoft D. N., 2015)

- **Aplicaciones básicas:** sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad;
- **De exposición de funcionalidades:** donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web);
- **De integración de servicios:** facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración;
- **De composición de procesos:** que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio;
- **De entrega:** donde los servicios son desplegados a los usuarios finales.

**FIGURA 6:** Características del código de barras



*Fuente: Elaboración propia 2020*

## 2.5. SEGURIDAD DE INFORMACION

Casi todas las organizaciones públicas o privadas, al igual que las personas, dependen de alguna manera de la tecnología de la información como una

herramienta esencial para lograr sus objetivos de negocio o para poder desarrollar actividades en su vida cotidiana; al mismo tiempo, todos tienen que enfrentarse con una amplia gama de amenazas y vulnerabilidades asociadas a los entornos informáticos de hoy. (Tarazona, 2007)

La seguridad de la información es más que un problema de seguridad de datos en los computadores; debe estar básicamente orientada a proteger la propiedad intelectual y la información importante de las organizaciones y de las personas.

Los riesgos de la información están presentes cuando confluyen dos elementos: amenazas y vulnerabilidades. Las amenazas y vulnerabilidades están íntimamente ligadas, y no puede haber ninguna consecuencia sin la presencia conjunta de éstas. Las amenazas deben tomar ventaja de las vulnerabilidades y pueden venir de cualquier parte, interna o externa, relacionada con el entorno de las organizaciones. (Tarazona, 2007)

Básicamente, se puede agrupar las amenazas a la información en cuatro grandes categorías: Factores Humanos (accidentales, errores); Fallas en los sistemas de procesamiento de información; Desastres naturales y; Actos maliciosos o malintencionados; algunas de estas amenazas son:

- Virus informáticos o código malicioso
- Uso no autorizado de Sistemas Informáticos • Robo de Información
- Fraudes basados en el uso de computadores • Suplantación de identidad
- Denegación de Servicios (DoS)
- Ataques de Fuerza Bruta
- Alteración de la Información
- Divulgación de Información
- Desastres Naturales
- Sabotaje, vandalismo
- Espionaje

“La información es un activo que, al igual que otros activos del negocio, es esencial para la organización, y por lo tanto debe ser protegido de forma adecuada.” (Tarazona, 2007)

La **OCDE** desarrolló por primera vez en 1992 una serie de Directrices para la Seguridad de los Sistemas de Información, las cuales tratan de promover el uso y desarrollo de una cultura de la Seguridad, no solo en el desarrollo de Sistemas y Redes de comunicación, sino mediante la adopción de “nuevas formas de pensamiento y comportamiento en el uso de la interconexión de esos sistemas”.

Con la evolución de los sistemas de información y de la forma de hacer negocios, la información se ha convertido en uno de los activos de mayor valor para las personas y especialmente para las organizaciones. “Los sistemas, redes y servicios de información afines, deben ser fiables y seguros, dado que los participantes son cada más dependientes de estos. Sólo un enfoque que tenga en cuenta los intereses de todos los participantes y la naturaleza de los sistemas, redes y servicios afines, puede proporcionar una seguridad efectiva.” (Tarazona, 2007)

Los objetivos que se buscan con la Gestión de la Seguridad de la Información son la protección de la confidencialidad, integridad y disponibilidad de la información y de los bienes que la contienen o procesan. De esta manera, las organizaciones y personas se pueden proteger de:

Divulgación indebida de información sensible o confidencial, de forma accidental o bien, sin autorización.

- Pérdida de información importante sin posibilidad de recuperarla.
- No tener acceso o disponibilidad de la información

La información debe ser manejada y protegida adecuadamente de los riesgos o amenazas que enfrente. La información valiosa se puede encontrar en diferentes formas: impresa, almacenada electrónicamente, transmitida por diferentes medios

de comunicación o de transporte, divulgada por medios audiovisuales, en el conocimiento de las personas, etc. (Tarazona, 2007)

### **2.5.1. Aspectos generales**

La información ha sido uno de los elementos claves en el desarrollo y éxito de los negocios y en el desarrollo de la gran mayoría de actividades diarias para los seres humanos. Por esta razón, las organizaciones son cada vez más conscientes de la necesidad de proteger la información de las diferentes amenazas a las que están expuestas. (Tarazona, 2007)

Hay muchos tipos de amenazas contra los sistemas de información y contra la información en general. No todas las amenazas están relacionadas con delitos informáticos, pero todas, de alguna forma, significan un riesgo para las organizaciones y sus consecuencias deben ser evaluadas. Fallas de Hardware o software, situaciones ambientales o naturales, accidentes, amenazas deliberadas con carácter delictivo como robo o destrucción de propiedad, y en general cualquier tipo de amenazas con origen interno o externo.

La seguridad de la información está orienta a proteger una serie de atributos que principalmente están relacionados con la confidencialidad, integridad y disponibilidad de ésta y por lo tanto a evitar que las amenazas a estas categorías puedan afectar a las organizaciones. (Tarazona, 2007)

Sin embargo, no hay ninguna medida que pueda garantizar un ambiente libre de amenazas o sin riesgos para la información y para las organizaciones o individuos que la requieren. Por esta razón es que se hace necesario adoptar modelos adecuados de gestión de la seguridad de la información que permitan lograr niveles efectivos de protección, basados en la relación coordinada de los diferentes mecanismos existentes, especialmente, los elementos físicos de protección basados en hardware y software, los elementos administrativos como políticas y procedimientos, y el recurso humano que administra, opera y utiliza los recursos informáticos. (Tarazona, 2007)

## 2.5.2. CRIPTOGRAFÍA

Aportando una visión más específica, la criptografía es la creación de técnicas para el cifrado de datos. Teniendo como objetivo conseguir la confidencialidad de los mensajes. Si la criptografía es la creación de mecanismos para cifrar datos, el criptoanálisis son los métodos para “romper” estos mecanismos y obtener la información. Una vez que nuestros datos han pasado un proceso criptográfico se dice que la información se encuentra cifrada. (H. Corrales, 2014)

El primer sistema criptográfico del que se tiene constancia es la Escítala. Este sistema data del siglo V a.c. y era usado en Esparta.

El sistema consistía en dos varas del mismo grosor, una en poder del emisor y la otra del receptor. Cuando el emisor quería enviar un mensaje, este, enrollaba una cinta en su vara y escribía el mensaje. De este modo al desenrollar la cinta el mensaje era ilegible. Al recibir el mensaje, el receptor enrollaba la cinta en su vara, y de este modo podía leer el mensaje.

Los primeros sistemas de cifrado estuvieron ligados a campañas militares dada la necesidad de evitar que el enemigo obtuviese los movimientos de las tropas al interceptar mensajes.

Otro método de cifrado clásico es el conocido cifrado de Cesar. Su nombre viene de la supuesta utilización por parte de Julio de César de este sistema.

La criptografía siempre había estado vinculada al ámbito militar. ¿Por qué se hizo necesaria para el resto de la gente?

Aunque el uso de comunicaciones seguras ha sido siempre una prioridad militar, la privacidad es requerida en otros sectores. Las empresas necesitan mantener unas comunicaciones seguras para proteger su información. Por esta razón el gobierno de EEUU y la NSA se ven obligados a crear DES. (H. Corrales, 2014)

A parte de a las empresas, se hace necesario otorgar al ciudadano de privacidad y seguridad. Con el nacimiento de internet y la progresiva oferta de servicios telemáticos como acceso al banco, citas médicas y un sinfín de posibilidades se tiene que ofrecer confidencialidad y seguridad a estos servicios.

Por estas razones es necesaria la criptografía. Para otorgar privacidad, confidencialidad y seguridad a nuestras transacciones telemáticas.

La criptografía cuenta con 3 usos: Cifrar, autenticar y firmar. (H. Corrales, 2014)

Cifrar:

Como ya se menciono, siempre hay cierta información que no se pretende que sea conocida más que por las personas que se quieran. En esto ayuda el cifrado. Cifrando un mensaje sucede que este no pueda ser leído por terceras personas consiguiendo así la tan deseada privacidad.

Autenticación:

Otra de las necesidades que surgen con la aparición de internet es la necesidad de demostrar que la persona sea esa persona y que el emisor es quien dice ser. Un método de autenticación puede ser el propio cifrado. Si se cifra un mensaje con una clave solo conocida por nosotros, demostrando que se es quien se dice ser, el receptor podrá constatar nuestra identidad descifrándolo. Esto se puede conseguir mediante clave simétrica (el receptor tiene que estar en posesión de la clave empleada) o usando clave asimétrica en su modo de autenticación. (H. Corrales, 2014)

Firmar:

Dados los trámites que se pueden realizar hoy en día a través de internet se hace necesaria la aparición de la firma digital. Igual que al firmar un documento, la firma digital ofrece la posibilidad de asociar una identidad a un mensaje.

Para la firma digital se utiliza clave asimétrica (dos claves una privada y otra pública). Lo que se cifra con la clave privada (que solo el propietario conoce) sólo se puede descifrar con la pública. De esta forma al cifrar con nuestra clave privada se demuestra que se es quien se dice ser. (H. Corrales, 2014)

La firma digital tiene un problema. ¿Cómo sabe el receptor que la clave corresponde realmente con la persona o entidad que dice poseerla? De este modo surgen las entidades de certificación. Organismos de confianza que actúan como notarios.

Otro sistema existente la red de confianza. En esta red los usuarios certifican si los demás son quien dicen ser. De este modo podría decirse que cada usuario se constituye como entidad certificadora. (H. Corrales, 2014)

Cabe destacar el uso incorrecto del término encriptar, que proviene de una mala traducción del inglés encrypt. La palabra encriptar no está reconocida por la RAE y el término correcto es cifrar. La interpretación del término encriptar sería introducir cuerpos en una cripta. (H. Corrales, 2014)

### **2.5.3. Código de barras**

El Código de barras moderno comenzó en 1948. Bernard Silver, un estudiante graduado en el Instituto de Tecnología de Drexel en Filadelfia, escuchó al presidente de una cadena local de alimentos preguntarle a uno de los decanos para llevar a cabo la investigación para desarrollar un sistema para leer automáticamente la información del producto durante la verificación. Silver le comentó a su amigo Norman Joseph Woodland sobre la solicitud del presidente de la cadena alimentaria. Woodland era un estudiante de veintisiete años de edad y profesor de postgrado en Drexel. El problema lo fascinó y comenzó a trabajar en él.

### **2.5.3.1. Como funciona el código de barras**

Un código de barras funciona en la práctica de manera similar a una linterna común - Leyendo la luz reflejada de una superficie. El proceso comienza con un dispositivo que emite un rayo de luz directa sobre un código de barras. El dispositivo contiene un pequeño sensor que detecta la luz reflejada y la convierte en energía eléctrica. El resultado, es una señal eléctrica que puede ser interpretada y convertida en datos. (Hoyo, 2014)

Los códigos de barras se miden en proporción a la barra más delgada y en mils, o 1/1000 de pulgada. Un código de barras de 15 mils, por ejemplo, tiene una barra delgada de 15/1000 de pulgada de ancho. Añadido a esto, se incluyen las zonas silenciosas o espacios en blanco, a ambos lados del símbolo, para garantizar la lectura de código de barras (Hoyo, 2014)

Un código de barras numérico puede ser de 6, 8, 12 o 13 caracteres

Los códigos de barras son asignados localmente pero son únicos a nivel mundial, se usa en más de 100 países. Para mercancías en venta. Esto se logra a través de la asignación de prefijos que en el caso de México los tres primeros números siempre serán 750 asignado por EAN (European Article Number) y después siguen los números según AMECE asigna estos códigos dependiendo de la compañía. (Hoyo, 2014)

Por ejemplo:

Tiene información sobre la del país, el fabricante, el producto y un dígito verificador. ejemplo: 7501234512343

750 - País (México) asignado por EAN internacional. 12345 - Número de fabricante asignado por AMECE. 1234 - Número de producto asignado por fabricante. 3 - Dígito verificador calculado por un algoritmo. (Hoyo, 2014)

### 2.5.3.2. Metodología empleada

Vamos a ver qué operaciones aritméticas realiza el ordenador sobre los dígitos del código de barras.

1.-Suma los dígitos de las posiciones pares:  $4+4+3+0+0+5=16$

2.-Multiplica el resultado por 3:  $3 \times 16 = 48$

3.-Suma al resultado anterior los dígitos de las posiciones impares ( el dígito de control no):  $48+8+1+2+7+0+1=67$

4.-Resta la suma obtenida del siguiente múltiplo de 10:  $70-67=3$

Como podemos observar obtenemos como resultado el dígito control, esto indica que se ha realizado correctamente el escaneado del precio. (Hoyo, 2014)

**FIGURA 7: Características del código de barras**



*Fuente: Hoyo C., 2014*

Resultados:

Lo que logramos en este proyecto es entender cómo funciona un código de barras y averiguamos el porque este código es tan sencillo pero elemental para cualquier industria. (Hoyo, 2014)

## 2.5.4. Generadores de códigos únicos simétricos

### 2.5.4.1. Algoritmo Random Code

Por sus siglas en inglés Código aleatorio son caracteres elegidos al “Azar” son útiles en diversas aplicaciones, entre las cuales podemos mencionar:

- **Método montecarlo:** Se simula un proceso natural en forma computacional. Estas aplicaciones se realizan en muy variados campos con el fin de emular distintos comportamientos: física (por ejemplo, para simular colisiones entre partículas), ingeniería (diseño de obras hidráulicas, puentes, etc. ), inversiones de capital, redes, servicios a clientes, call centers, etc. La simulación a través de la computadora es una herramienta poderosa para comprender la naturaleza de sistemas complejos. (Carvallo, 2004)
- **Muestreo:** Con el fin de seleccionar una submuestra de una población.
- **Análisis numérico:** Algunas técnicas para resolver problemas de análisis numérico complejos han sido desarrolladas usando números aleatorios.
- **Programación:** La generación de valores aleatorios puede ser útil para poner a prueba la efectividad de un algoritmo. También son útiles en criptología. (Carvallo, 2004)

A pesar de que fue en la década del 40 que las primeras computadoras modernas fueron desarrolladas, la simulación ya existía en forma embrionaria aún antes de que la computadora apareciera en escena. Así, por ejemplo, en la segunda mitad del siglo XIX, se realizaban experiencias arrojando agujas al azar sobre una superficie reglada con el fin de estimar el número  $\pi$ . En 1908 W. S. Gosset, bajo el seudónimo de Student, realizaba un muestreo experimental con el fin de descubrir la distribución de un estimador de la correlación en una distribución normal bivariada. En ese momento los números aleatorios se generaban mediante métodos observacionales (mecanismos físicos) tales como tirar un dado, extraer una carta de un mazo o mediante una ruleta. (Carvallo, 2004)

Dado el esfuerzo que significaba generar números aleatorios cada vez que eran necesarios, parece razonable que se hayan construido tales números y luego tabulado. Tippett (1927) publicó una tabla con 41600 números aleatorios “tomados en forma aleatoria de informes censales”. Cada número era uno de los enteros 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 y el usuario tomaba varios de ellos y agregaba un punto decimal para formar un número aleatorio entre 0 y 1. Desde ese momento una serie generadores de números aleatorios fueron propuestos. La primera máquina fue usada en 1939 por Kendall y Babington-Smith con el fin de producir una tabla de 100000 de dígitos aleatorios y en 1955 la RAND Corporation utilizó extensamente una tabla de 1000000 dígitos aleatorios que fue obtenida a partir de una ruleta electrónica especialmente diseñada. ERNIE fue una famosa máquina de números aleatorios que fue usada por la lotería británica, es decir la British Premium Savings Bonds Lottery.

Poco después de la aparición de las computadoras, se comenzó a buscar maneras eficientes de obtener números aleatorios, pues aún cuando se podían usar las tablas existentes éste era un recurso limitado, ya sea por el espacio de memoria necesario como por resultar, en algunos casos, cortas. Si bien máquinas como ERNIE podrían haber trabajado junto con una computadora, una solución en la que la computadora provee todo parecía más satisfactoria. La búsqueda se orientó, entonces, a la producción de números aleatorios usando operaciones aritméticas de una computadora. John von Neumann sugirió en un principio, alrededor de 1946, usar el método del “cuadrado medio”. Su idea era calcular el cuadrado del número aleatorio anterior y tomar los dígitos del medio del número calculado. Así, por ejemplo, si queremos generar un número aleatorio de 10 dígitos y el número anterior es:

5772156649 —> 33317792380594909201

el nuevo número será 7923805949. (Carvallo, 2004)

La primera pregunta que cabe hacer es porqué motivo un número generado por este procedimiento que es determinístico, va a resultar aleatorio. La respuesta es

que el número no es aleatorio, pero parece serlo, en el sentido en que en una aplicación la relación real entre un número y el siguiente no tiene ningún significado físico. Por lo tanto, el carácter no aleatorio no es una característica indeseable y podría ser que el “cuadrado medio” resultase ser un buen “batido” del número anterior. Es claro, de todas formas, que un mecanismo de esta naturaleza no podría haber reemplazado a ERNIE.

Las secuencias de números generadas en forma determinística reciben el nombre de secuencias pseudo-aleatorias o quasi-aleatorias, si bien nosotros nos referiremos a ellas como secuencias aleatorias, sobreentendiendo que sólo “parecen” aleatorias. Números aleatorios generados en forma determinística en una computadora funcionan muy bien en muchísimas aplicaciones, a condición de que el método de generación sea bueno. (Carvallo, 2004)

**a) Ventajas**

- Ampliamente utilizado en diversas aplicaciones, como métodos de monte carlo, muestreo, análisis numérico, programación.
- Al ser simétrico su implementación no es compleja.
- No requieren grandes recursos de hardware para ser implementada.
- Ampliamente difundido.

**b) Desventajas**

- Vulnerabilidad latente
- Al no ser asimétrico, se pueden descifrar con poco esfuerzo
- Recomendable para entornos locales ejemplo: Intranets. (Carvallo, 2004)

#### **2.5.4.2. Algoritmo DES**

El algoritmo Data Encryption Standard (DES), también conocido como Data Encryption Algorithm (DEA), es el método de cifrado simétrico más ampliamente conocido. Ha sido el primer algoritmo moderno de cifrado para intereses comerciales, difundido de forma abierta y completa a todo el mundo, en todos sus detalles de implementación. Su estándar queda recogido en American Standard

FIPS 46–2 y FIPS 46–3 (donde se añade el algoritmo TripleDES o TDEA: Triple Data Encryption Algorithm). Estos documentos están disponibles y se encuentran fácilmente en la web: vale la pena disponer de ellos.

(FIPS: “Federal Information Processing Standards” son estándares anunciados públicamente desarrollados por el gobierno de los Estados Unidos para su utilización por parte de todas las agencias del gobierno no militares y por los contratistas del gobierno. Muchos estándares FIPS son versiones modificadas de los estándares usados en las comunidades más amplias como por ejemplo ANSI, IEEE, ISO, etc.) (Johannes, 2004)

El diseño de DES se basa en dos conceptos generales de criptografía: en el cifrado de producto y en el cifrado de Feistel. Es conveniente, antes de entrarnos en la descripción del algoritmo DES, presentar estos dos conceptos, que dan luego forma al algoritmo. Veamos pues cada uno de estos dos conceptos: (Johannes, 2004)

### **Cifrado de producto**

La idea básica del cifrado de producto es la de construir una función de cifrado compleja a partir de la composición de varias operaciones simples que se ofrecen de forma complementaria, y que son individualmente insuficientes para lograr un criptosistema robusto. Sus operaciones básicas incluyen las trasposiciones, las translaciones (por ejemplo, las realizadas mediante el operador XOR) y transformaciones lineales, operadores aritméticos, multiplicación modular y simple sustitución.

Un cifrado de producto combina dos o más transformaciones con una intención determinada, de manera que el cifrado resultante es más seguro que el de cada transformación por separado. Muchos de los algoritmos simétricos de cifrado en la práctica son cífrados de producto. (Johannes, 2004)

Llamamos ITERACIÓN DE CIFRADO DE BLOQUE a un cifrado de bloque que implica la repetición secuencial de una función interna. Los parámetros que

definen este cifrado son el número de vueltas o iteraciones, , el tamaño del bloque, y el número de bits. (Johannes, 2004)

**a) Ventajas**

- Es uno de los sistemas mas empleados y extendidos, por tanto es de los mas probados.
- Implementación sencilla.

**b) Desventajas**

- No es seguro
- No se permite una clave de longitud variable, es decir no se puede aumentar el tamaño del código para aumentar la seguridad
- Es vulnerable a criptoanalisis.
- La longitud de clave de 56 bits es demasiado corta, y por tanto vulnerable.  
Actualmente DES ya no es un estándar, debido a que en 1999 fue roto.  
(Johannes, 2004)

#### **2.5.4.3. Universally Unique Identifier UUID**

La firma **UUID** son las siglas de *Universally Unique IDentifier*, que en inglés significa, literalmente, ‘identificador único universal’. Como tal, el UUID es un código identificador estándar empleado en el proceso de construcción de software. Es utilizado para crear identificadores únicos universales que permitan reconocer e distinguir un objeto dentro de un sistema, o el mismo objeto en diferentes contextos. El UUID tiene una longitud de 128 bit, es decir, 16 bytes. El número de posibles UUID es de 3x1038, por ello las probabilidades de coincidir dos códigos UUID son extremadamente bajas. (Gonzales, 2009)

Como tal, un UUID está conformado por 32 dígitos divididos en cinco grupos, que obedecen a la siguiente forma: 8-4-4-4-12. Es decir, 36 caracteres y cuatro guiones que los separan. Por ejemplo: 670b9562-b30d-52d5-b827-655787665500 es una de 128 bits número usado para identificar la información en los sistemas informáticos. El término identificador único global ( GUID ).

Cuando generados de acuerdo con los métodos estándar, UUID son únicas para fines prácticos, sin depender por su singularidad en una autoridad de registro central o la coordinación entre las partes generando ellos, a diferencia de la mayoría de los otros sistemas de numeración. Mientras que la probabilidad de que se duplicará un UUID no es cero, es suficiente a cero cerca de ser insignificante.

Por lo tanto, cualquier persona puede crear un UUID y utilizarlo para identificar algo casi con certeza que el identificador no duplica uno que ya ha sido o va a ser, creado para identificar otra cosa. Información marcada con UUID por partes independientes, por tanto, puede ser más tarde combinan en una sola base de datos, o transmitida en el mismo canal, con una probabilidad despreciable de la duplicación. Los identificadores únicos se utilizan en diferentes áreas. Ejemplos típicos son las redes informáticas, pero también se utilizan en el sector bancario (BIC) y en el comercio (números de serie de los productos). (Gonzales, 2009)

En su representación textual canónica los dieciséis octetos de un UUID se representan como 32 dígitos hexadecimales (base 16) mostrados en cinco grupos separados por guiones de la forma 8-4-4-4-12, dando un total de 36 caracteres (32 caracteres alfanuméricos y cuatro guiones). Por ejemplo:

123e4567-e89b-12d3-a456-426655440000  
xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx

Los tres bits más significativos del dígito *N* indican la variante UUID y los cuatro bits del dígito *M* indican la versión UUID. En el ejemplo, *M* es 1 y *N* es a (10xx), significando que el UUID es una variante 1 de la versión 1; es decir, un DCE / RFC 4122 UUID basado en el tiempo. (Gonzales, 2009)

La cadena 8-4-4-4-12 del formato canónico se basa en el "diseño de registro" para los 16 bytes de la UUID:

- Un número entero de 4 bytes (8 dígitos hexadecimales) "time\_low" con los 32 bits bajos del timestamp.

- Un número entero de 2 bytes (4 dígitos hexadecimales) "time\_mid" con los 16 bits centrales del timestamp.
- Un número entero de 2 bytes (4 dígitos hexadecimales) "time\_hi\_and\_version", con la "versión" de 4 bits en los bits más significativos, seguido por los 12 bits altos del timestamp.
- Dos campos de 1 byte (4 dígitos hexadecimales) "clock\_seq\_hi\_and\_reserved" y "clock\_seq\_lo", con la "variante" en los 1 a 3 bits más significativos de clock\_seq\_hi\_and\_reserved, seguido de la secuencia del reloj
- Seis bytes (12 dígitos hexadecimales) con el "nodo" de 48 bits (Gonzales, 2009)

Estos campos reflejan principalmente los de la versión 1 y 2 UUID, basados en el tiempo, pero la misma representación 8-4-4-4-12 se utiliza para todos los UUID, incluso si la semántica de la UUID es diferente de las versiones 1 y 2 o es opaca.

Los GUID de Microsoft son a veces representados entre corchetes:

{123e4567-e89b-12d3-a456-426655440000}

Este formato no debe confundirse con "formato de registro", que se refiere al formato *dentro de las llaves*.

RFC 4122 define un Uniform Resource Name espacio de nombres (URN) de UUID. Un UUID presentado como un URN aparece como sigue:

urn:uuid:123e4567-e89b-12d3-a456-426655440000 (Gonzales, 2009)

#### a) **Ventajas**

- Es uno de los sistemas de identificadores utilizados en la industria de la banca
- Un UUID puede ser utilizado como un identificador específico y ser utilizado en varias ocasiones para identificar el mismo objeto en diferentes contextos
- UUID está estandarizado por la Open Software Foundation (SOF)
- Se documenta como parte de la norma ISO / IEC 11578
- La probabilidad de que se genere un código similar es muy cercano a cero

**b) Desventajas**

- Mayor uso del espacio en la base de datos
- No puede realizarse el order by ID para conseguir la inserción de la orden
- No recomendable su aplicación en aplicaciones de gran tamaño (Gonzales, 2009)

**2.5.5. Firma digital**

La firma digital es una solución tecnológica que permite añadir a documentos digitales y mensajes de correo electrónico una huella o marca única, a través de ciertas operaciones matemáticas. (Hoyo, 2014)

La firma digital permite al receptor del mensaje o documento:

- Identificar al firmante de forma fehaciente (Autenticación)
- Asegurar que el contenido no pudo ser modificado luego de la firma sin dejar evidencia de la alteración (Integridad)
- Tener garantías de que la firma se realizó bajo el control absoluto del firmante (Exclusividad)
- Demostrar el origen de la firma y la integridad del mensaje ante terceros, de modo que el firmante no pueda negar o repudiar su existencia o autoría (No Repudio)

Facilita el reemplazo de documentación en papel por su equivalente en formato digital. Ahorra costos, simplifica procedimientos y brinda seguridad en el intercambio de información.

Se utiliza principalmente para firmar documentos PDF y correos electrónicos, pero también permite firmar documentos de texto, plantillas, imágenes y virtualmente cualquier tipo de documento. Su tecnología está incorporada en transacciones electrónicas, formularios web y navegación en páginas seguras.

#### **2.5.5.1. Funcionamiento de firma digital**

La tecnología de firma digital se sostiene de dos pilares: un método que hace imposible la alteración de la firma y una infraestructura que permite certificar la identidad del firmante.

#### **2.5.5.2. Clave asimétrica**

La Clave Asimétrica es un método de criptografía o codificación, en el que se generan dos números de gran longitud (usualmente más de 200 cifras) mediante una fórmula matemática compleja. Estos números, llamados “claves”, son distintos, pero están relacionados de modo tal que lo que se cifra o encripta con una clave sólo puede descifrarse con la otra. A este par de claves se los conoce como Clave Pública y Clave Privada. La clave pública se distribuye y la clave privada la conserva el propietario, protegida por una o varias contraseñas que sólo él conoce. El par de claves funciona siempre en conjunto: No es posible cifrar y descifrar un documento con una misma clave.

Cuando se aplica la clave privada sobre un documento digital en su totalidad, este queda cifrado o encriptado. Es decir, se vuelve ilegible para cualquiera que no posea la clave pública con que descifrarlo. En firma digital, ya que no se busca encriptar el mensaje sino darle una marca de autenticación, la clave asimétrica se utiliza de forma indirecta, no sobre el documento, sino sobre un resumen del mismo, denominado hash.

#### **2.5.5.3. Hash**

El hash (también conocido como digesto o huella digital), es un resumen único que identifica a un documento digital. Se puede aplicar a cualquier tipo de documento, incluso a una cadena de texto. Se obtiene al aplicar una fórmula matemática llamada “función unidireccional de resumen”.

#### **2.5.5.4. Certificación digital**

hash Para que el procedimiento de firma y autenticación sea confiable, necesitamos la seguridad de que esa clave pública efectivamente pertenece al firmante. Por eso, el segundo elemento que sostiene el sistema de firma digital es la “Infraestructura de Clave Pública” (PKI, en inglés), que regula cómo se emiten y distribuyen las claves. Para esto, utilizan documentos llamados Certificados de Clave Pública, o según nuestra normativa, Certificados Digitales. Un Certificado Digital es simplemente un documento firmado digitalmente por una autoridad, en el cual se atestigua que una clave pública pertenece a un determinado individuo o entidad. En general, contiene datos de identidad de la persona, su clave pública y el nombre de la autoridad que emitió el certificado. Todos los datos de identidad son previamente validados por esta autoridad, y el certificado se puede autenticar de la misma forma que cualquier otro documento con firma digital.

La Infraestructura de Clave Pública es el conjunto de procedimientos, políticas y roles normados que definen cómo se generan y organizan esos certificados. Si el certificado es auténtico y confiamos en la autoridad emisora, podemos asegurar la identidad del firmante. En nuestro país, esta regulación se conoce como Infraestructura de Firma Digital de la República Argentina (IFDRA).

#### **2.5.6. Código QR**

El Código QR es un código bidimensional, fácilmente identifiable por los tres cuadros ubicados en las esquinas superiores e inferior izquierda. Puede contener información de caracteres alfanuméricos, símbolos, Kanji, Hiragana, Katakana, códigos binarios y códigos de control. Es omnidireccional y su lectura puede realizarse desde un dispositivo móvil. Se realizó una revisión de artículos de las bases Scielo y Pubmed con el objetivo de indagar acerca de las aplicaciones de estos códigos en las ciencias de la salud y proponer algunas de estas para el Sistema Nacional de Salud cubano, cuya introducción ha sido paulatina, principalmente en la práctica y en la educación médica. Sin embargo, la

diseminación y el uso es aún incipiente y existen muchas oportunidades. Un sistema de identificación nacional en salud permitiría una autenticación más fácil, rápida y efectiva, con un ahorro sustancial de recursos. Las empresas farmacéuticas podrían emplear un sistema similar, en este caso con informaciones de medicamentos. A pesar de sus limitaciones, son diversas las aplicaciones que poseen estos códigos en los servicios de salud. Esto, unido a la expansión tecnológica que vive hoy Cuba, permitirá en un futuro mediato la generalización y la difusión de estas tecnologías en beneficio de la sociedad. (Gonzales, 2015)

El Código QR (quick response code, por sus siglas en inglés), diseñado para ser decodificado a alta velocidad, es un código bidimensional, donde la información se encuentra codificada dentro de un cuadrado. Son fácilmente identificables por los tres cuadros ubicados en las esquinas superiores e inferior izquierda. Diversas han sido las causas que han propiciado su expansión a nivel mundial, entre ellas: la estandarización por la norma ISO/IEC18004, en junio de 2000, y tener un código abierto con derechos de patente (propiedad de Denso Wave) no ejercidos. (Gonzales, 2015)

Según las especificaciones técnicas de Denso Wave acerca de los Códigos QR, cuando se reproduce la herramienta en cualquier superficie, se debe dejar alrededor de este un espacio de reserva (quiet zone) equivalente a cuatro módulos (un módulo es el cuadrado mínimo que hay en cada QR) para que el lector lo ubique e interprete correctamente. El código QR posee 9 rasgos de estandarización:

- Capacidad para manejar diferentes tipos de datos: alfanuméricos, símbolos, Kanji, Hiragana, Katakana, códigos binarios y códigos de control.
- Gran capacidad: hasta 7 089 números y 4 296 caracteres alfanuméricos pueden codificarse (cientos de veces más que un código de barras).
- Pequeño tamaño de impresión: la dimensión mínima es de 10 mm cuadrados.

- Alta velocidad de escaneo o lectura: su lectura es omnidireccional, con detectores de patrones de posicionamiento, que burlan los efectos negativos de la interferencia de los fondos.
- Estandarización universal: AIM International Standard, la ISO International Standard (ISO/IEC18004), Japanese Industrial Standard y Japanese Electronic Industry Development Association.
- Correctores de suciedad y daño: los códigos QR permiten un máximo de 30 % de daño sin pérdida de información, con niveles de seguridad [L (7 % de las claves se pueden restaurar), M (15 %), Q (25 %) y H (30 %)].
- Compartimentación: pueden dividirse en múltiples áreas de datos (hasta 16); permiten pequeñas impresiones dentro.
- Representaciones flexibles: las formas y colores pueden ser cambiadas, hasta convertirse en representaciones de arte (códigos QR artísticos).
- Lectura: pueden ser leídos por teléfonos inteligentes, tabletas o computadoras portátiles con cámaras, usando softwares gratuitos. (Gonzales, 2015)

#### **2.5.7. Características de código QR**

Puede contener información variada; es capaz de codificar todos los caracteres ASCII además de información binaria; es omnidireccional y su lectura puede realizarse desde un dispositivo móvil, lo que permite realizar acciones de forma automática en el propio dispositivo. (Gonzales, 2015)

**TABLA 3: Capacidad de datos código QR**

Solo numerico	7089 caracteres
Alfanumerico	4296 caracteres
Binario	2953 caracteres
Kanji/Kana	1817 caracteres

*Fuente: Gonzales J., 2015*

## **2.6. WORKFLOW**

Los sistemas de workflow también conocidos como “BPMS-Business Process Management System”, han adquirido considerable importancia en las empresas de todos los sectores. Una de sus grandes ventajas, es que le permiten a las organizaciones ser más competitivas.

“En los últimos años y a nivel mundial, los sistemas de workflow han estimulado gran interés por dos razones principales: la económica y la tecnológica.

La razón económica, se basa en el reconocimiento por parte de diversos sectores de la economía que debido a la globalización de los mercados; el ambiente competitivo de los negocios actual necesita la automatización de todas las actividades y procesos de trabajo y no solo la automatización de algunas tareas individuales.

La razón tecnológica, resulta de la aparición de nuevos ambientes computacionales, capaces de integrar varias aplicaciones que antes operaban de manera independiente. Debido a estas razones, las expectativas hacia la adopción de sistemas de workflow cada día son más evidentes” (Rodríguez Vargas & Serna Martínez)

### **2.6.1. Definición de sistemas workflow**

El flujo de trabajo workflow, es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su

orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

“Los sistemas de workflow son herramientas que permiten la implementación técnica de los procesos del negocio. Admiten el soporte y agilizan el proceso de negocio ganando tiempo. Autoriza a la gente involucrada para que lleve a cabo procesos complejos independientemente del tiempo y el lugar.

El flujo de trabajo es controlado y coordinado activamente por el sistema de workflow. El control incluye el monitoreo de pasos de trabajo individuales y el inicio de procesos para escalar las tareas que lleguen a su fecha de vencimiento”

EL workflow ha tenido una evolución exorbitante, ello se refleja, en la creación de algunos productos: (Marciales, 2011)

#### **2.6.1.1. Procesamiento de imágenes**

“En este caso, se captura en forma de imagen electrónica (por ejemplo mediante un escáner) la información o documento que se desea, para luego ser pasada entre los diferentes participantes con distintos propósitos, durante la realización de un proceso”.

#### **2.6.1.2. Administración de documentos**

“Esta tecnología está relacionada con la administración del ciclo de vida de los documentos. Esta incluye facilidades para guardar en un deposito común aquellos documentos que se comparten, así como también las facilidades para el acceso o modificación de los mismos mediante un conjunto predefinido de reglas”.

#### **2.6.1.3. Correo electrónico y directorios**

“El correo electrónico provee las facilidades para distribuir información entre individuos de una organización o entre distintas organizaciones. El sistema de directorios no sólo provee una forma de identificar a los participantes dentro de un conjunto de direcciones de correo electrónico, sino que ofrece además, la

potencialidad de registrar la información sobre los participantes; es decir, roles dentro de la empresa u otros atributos”.

#### **2.6.1.4. Aplicaciones basadas en transacciones**

“Las transacciones de workflow guardan la información, reglas, roles y otros elementos sobre un servidor de bases de datos relacionados, ejecutando la aplicación de workflow sobre una interfaz gráfica para los usuarios. Estas aplicaciones típicamente incluyen componentes gráficos para el ingreso de los datos”.

#### **2.6.1.5. Procesamiento de formularios**

“El ambiente de los formularios es amigable y familiar para muchos usuarios. Este es un excelente vehículo para el manejo de la información dentro de una aplicación de workflow, basado en el valor de los campos de un formulario. Algunos productos para implementar aplicaciones de workflow proveen constructores de formularios o se integran a constructores de terceros

### **2.6.2. Usos del workflow**

Existen muchos software para apoyar dominios de los workflows particularmente. Esos sistemas manejan tareas como el encaminamiento automático, proceso e integración parcialmente automatizados entre los diversos usos del software y sistemas funcionales del hardware que contribuyen al proceso de valor-adición subyacente.

### 2.6.3. Elementos de un sistema workflow

FIGURA 8: Elementos de workflow



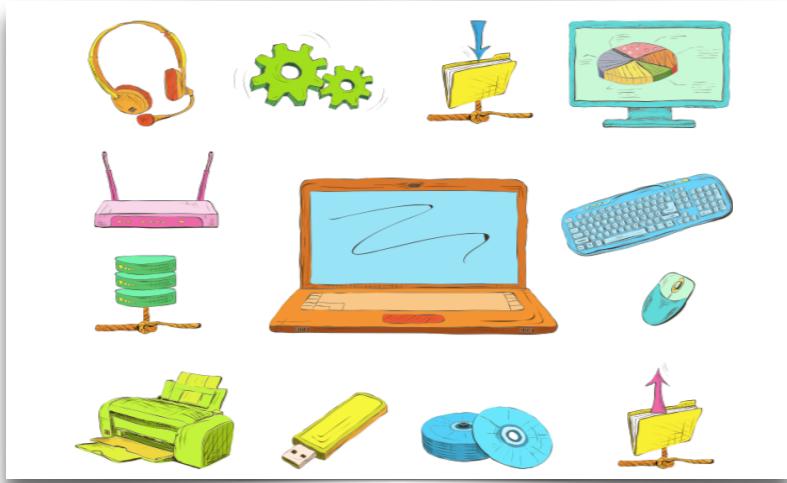
Fuente: Slideshare.net, 1999

Un sistema de workflow, se compone colectivamente de los siguientes elementos:

- Tareas.
- Personas (usuarios).
- Roles.
- Rutas.
- Reglas de transición.
- Datos.
- Eventos.
- Plazos (deadlines).
- Procesos.
- Políticas. (Slideshare.net, 1999)

#### **2.6.4. Tecnologías utilizadas en sistemas workflow**

**FIGURA 9: Diferentes tecnologías y productos que usa los sistemas workflow**



*Fuente: Aguirre, 2014*

Los sistemas de workflow hacen uso de una serie de productos y tecnologías, que permiten gestionar documentos, dar soporte al trabajo en equipo, gestionar proyectos, etc. (Aguirre, 2014)

#### **2.6.5. Tipos de sistemas workflow**

“Frecuentemente, este tipo de workflow es llamado workflow de transacciones. Esto se debe a que la transacción en una base de datos es considerada la clave de todo proceso. Este tipo de Workflow es el segmento más grande en el mercado. En general automatizan procesos de negocios que tienden a ser repetitivos, bien estructurados y con gran manejo de datos.(Aguirre, 2014)

##### **2.6.5.1. Workflow de colaboración**

Los workflow de colaboración estructuran o semi-estructuran procesos de negocios donde participan personas; con el objetivo de lograr una meta común. Típicamente involucran documentos, los cuales son los contenedores de la información. Se sigue la ruta de estos paso a paso, además de las acciones que se toman sobre ellos. Los documentos son la clave y por lo tanto, son esenciales

para la solución de workflow mantener la integridad de dichos documentos".  
(Aguirre, 2014)

### **2.6.5.2. Workflow de administración**

El workflow administrativo, es aquel que involucra procesos de administración en una empresa, tales como órdenes de compra, reportes de ventas, etc. Estos workflow, se emplean cuando existe una gran cantidad de procesos administrativos dentro de la empresa y es necesaria la distribución de soluciones a diferentes usuarios.(Aguirre, 2014)

## **2.7. TECNOLOGÍAS DE DESARROLLO**

### **2.7.1. Lenguajes de programación**

Un lenguaje de programación es un convenio entre personas que puede definirse así:

Conjunto de reglas o normas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador (sin ambigüedades).

- Por tanto, un lenguaje de programación es un convenio o acuerdo acerca de como se debe de interpretar el significado de los programas de dicho lenguaje
- Muchas veces se confunden los lenguajes con los compiladores, interpretes o con los entornos de desarrollo de software (Ureña, 2012)

#### **2.7.1.1. HiperText Pre-Processor (PHP)**

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de linea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. (Group, 2008)

#### a) Características

Se puede mencionar algunas de las características más importantes: (Group, 2008)

- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y exemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de POO.

- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

**b) Ventajas**

La ventajas más importantes de utilizar PHP son: (Group, 2008)

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial usando la enorme cantidad de módulos

**c) Desventajas**

Las desventajas que presenta PHP son: (Group, 2008)

- Como es un lenguaje que se interpreta en ejecución para ciertos usos, puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- La ofuscación es una técnica que puede dificultar la lectura del código pero no la impide y, en ciertos casos, representa un costo en tiempos de ejecución

### 2.7.1.2. JAVA

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados. (Deitel, 2004)

#### a) Características

Las principales características que proporciona java son: (Deitel, 2004)

- **Lenguaje Simple:** se lo conoce como lenguaje simple porque viene de la misma estructura de c y c++; ya que c++ fue un referente para la creación de java por eso utiliza determinadas características de c++ y se han eliminado otras.
- **Orientado a Objeto:** toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.
- **Distribuido:** permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

- **Robusto:** es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.
- **Seguro:** la seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.
- **Indiferente a la arquitectura:** java es compatible con los más variados entornos de red, cualesquiera sean estos desde Windows 95, Unix a Windows Nt y Mac, para poder trabajar con diferentes sistemas operativos.
- Java es muy versátil ya que utiliza byte-codes que es un formato intermedio que sirve para transportar el código eficientemente o de diferentes plataformas (Hardware - Software).

**b) Ventajas**

Las principales ventajas de java son: (Deitel, 2004)

- Manejo automático de la memoria.
- Lenguaje Multi-plataforma: El código escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma.
- Programación Orientada a Objetos: Paradigma muy utilizado hoy en día que facilita y organiza mucho la programación.
- Puede correr en el explorador y en dispositivos móviles.
- Sintaxis similar a C/C++, pero más simple.
- Fácil de aprender.

**c) Desventajas**

Las desventajas que se encuentran en java son: (Deitel, 2004)

- Requiere un intérprete.
- Algunas implementaciones y librerías pueden tener código rebuscado.

- Una mala implementación de un programa en java, puede resultar en algo muy lento.
- Algunas herramientas tienen un costo adicional.

### **2.7.1.3. Python**

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

Se puede escribir un script (o programa) en el interprete de comandos o un archivo por lotes de Windows para algunas de estas tareas, pero los scripts se lucen para mover archivos de un lado a otro y para modificar datos de texto, no para aplicaciones con interfaz de usuario o juegos. Se puede escribir un programa en C/ C++/Java, pero puede tomar mucho tiempo de desarrollo obtener al menos un primer borrador del programa. Python es más fácil de usar, está disponible para sistemas operativos Windows, Mac OS X y Unix. (Van Rossum, 2009)

Python es fácil de usar, pero es un lenguaje de programación de verdad, ofreciendo mucha mayor estructura y soporte para programas grandes que lo que lo que pueden ofrecer los scripts de Unix o archivos por lotes. Por otro lado, Python ofrece mucho más chequeo de error que C, y siendo un lenguaje de muy alto nivel, tiene tipos de datos de alto nivel incorporados como arreglos de tamaño flexible y diccionarios. Debido a sus tipos de datos más generales Python puede aplicarse a un dominio de problemas mayor que Awk o incluso Perl, y aún así muchas cosas siguen siendo al menos igual de fácil en Python que en esos lenguajes.

Python te permite separar tu programa en módulos que pueden reusarse en otros programas en Python. Viene con una gran colección de módulos estándar que

puedes usar como base de tus programas, o como ejemplos para empezar a aprender a programar en Python. Algunos de estos módulos proveen cosas como entrada/salida a archivos, llamadas al sistema, sockets, e incluso interfaces a sistemas de interfaz gráfica.

Python es un lenguaje interpretado, lo cual puede ahorrarte mucho tiempo durante el desarrollo ya que no es necesario compilar ni enlazar. El intérprete puede usarse interactivamente, lo que facilita experimentar con características del lenguaje, escribir programas descartables, o probar funciones cuando se hace desarrollo de programas de abajo hacia arriba. Es también una calculadora de escritorio práctica. (Van Rossum, 2009)

**a) Ventajas**

- Python es un lenguaje de programación poderoso y fácil de aprender.
- Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos
- La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.
- El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C)
- Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables.
- Se puede escribir un script (o programa) en el intérprete de comandos o un archivo por lotes de Windows para algunas de estas tareas
- Python es más fácil de usar, está disponible para sistemas operativos Windows, Mac OS X y Unix.
- Python ofrece mucho más chequeo de error que C, y siendo un lenguaje de muy alto nivel, tiene tipos de datos de alto nivel incorporados como arreglos de tamaño flexible y diccionarios. (Van Rossum, 2009)

**b) Desventajas**

Las ventajas que presenta el lenguaje de programación Python son:

- Los scripts se lucen para mover archivos de un lado a otro y para modificar datos de texto, no para aplicaciones con interfaz de usuario o juegos
- Se puede escribir un programa en C/C++/Java, pero puede tomar mucho tiempo de desarrollo obtener al menos un primer borrador del programa (Van Rossum, 2009)

**2.7.2. Frameworks**

Un framework (amazon), es una abstracción en la que cierto código común provee una funcionalidad genérica que puede ser sobreescrita o especializada de forma selectiva por medio de código con funcionalidad específica provisto por los clientes del framework (desarrolladores de software / programadores).

Un framework es una solución incompleta (no funcional) pero concreta (a diferencia de los estilos arquitectónicos o los patrones de diseño) a un problema recurrente bien conocido.

Un framework facilita el desarrollo de software permitiendo a los diseñadores y programadores dedicar su tiempo a lograr los requerimientos de software en lugar de lidiar con los detalles de bajo nivel necesarios para obtener un sistema funcional. De esta forma se puede reducir el tiempo total de desarrollo de la aplicación.

Por ejemplo, un equipo que está desarrollando un sistema WEB bancario al usar un framework de desarrollo WEB puede enfocarse en el desarrollo de las operaciones de retiro y transferencias de dinero en lugar de tener que enfocarse en la mecánica del manejo de las peticiones HTTP o el manejo de las sesiones de los usuarios y el estado de la aplicación.

Los frameworks están conformados por zonas congeladas (frozen spots) y zonas

calientes (hot spots).

Las partes congeladas definen la arquitectura general de un sistema de software, es decir, sus componentes básicos y las relaciones entre estos. Esas partes permanecen inalteradas (congeladas) en cualquier instancia del framework.

Un framework caja blanca (white box) requiere que los usuarios tengan conocimiento de la estructura y código interno del framework, generalmente vienen con el código fuente y normalmente su comportamiento se extiende por medio del uso de subclases y herencia.

Un framework caja negra (black box) no requiere un entendimiento o conocimiento profundo del funcionamiento interno (estructura / código) del framework. Generalmente el framework se extiende componiendo y delegando comportamiento entre objetos (Muchos de los cuales son las extensiones del usuario (Gutierrez D., 2010)

#### **2.7.2.1. Laravel**

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".

Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP.

##### **a) Funcionamiento MVC**

- **Modelo:** Laravel incluye un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Es opcional el uso

de Eloquent, pues también dispone de otros recursos que facilitan interactuar con los datos, o específicamente la creación de modelos.

- **Vista:** Laravel incluye un paquete de procesamiento de plantillas llamado Blade. Este sistema de plantillas favorece un código mucho más limpio en las Vistas, además de incluir un sistema de Caché que lo hace mucho más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura. Por ejemplo, en vez pintar la vista usando el código PHP:
- **Controlador:** Los controladores contienen la lógica de la aplicación y permiten organizar el código en clases sin tener que escribirlo todo en las rutas.

**b) Características**

Algunas otras características interesantes de Laravel incluyen:

- Posee un poderoso conjunto de librerías.
- ORM increíble.
- Fácil enrutamiento.
- Autenticación simple.
- Sistema de ruteo, también RESTful.
- Blade, Motor de plantillas.
- Peticiones Fluent.
- Eloquent ORM.
- Basado en Composer.
- Soporte para el caché.
- Soporte para MVC.

**c) Ventajas**

- Reducción de costos y tiempos en el desarrollo y mantenimiento.
- Curva de aprendizaje relativamente Baja (en comparación con otros framework Php).

- Flexible y adaptable no solo al MVC Tradicional (Modelo vista controlador) sino que para reducir código propone usar “Routes with closures”
- Es modular y con una amplio sistemas de paquetes y drivers con el que se puede extender la funcionalidad de forma fácil, robusta y segura.
- Hace que el manejo de los datos en Laravel no sea complejo; mediante Eloquent (que es un ORM basado en el patrón active record) la interacción con las bases de datos es totalmente orientada a objetos, siendo compatible con la gran mayoría de las bases de datos del mercado actual y facilitando la migración de datos de una forma fácil y segura. Otro punto es que permite la creación de consultas robustas y complejas.
- Facilita el manejo de ruteo de la aplicación como así también la generación de url amigables y control de enlaces auto–actualizables lo que hace más fácil el mantenimiento de un sitio web.
- El sistema de plantillas Blade de Laravel, trae consigo la generación de mejoras en la parte de presentación de la aplicación como la generación de plantillas más simples y limpias en el código y además incluye un sistema de cache que las hace más rápidas, lo que mejora el rendimiento de la aplicación.
- También cuenta con una herramienta de interfaces de líneas de comando llamada Artisan que me permite programar tareas programadas como por ejemplo ejecutar migraciones, pruebas programadas, etc.

#### **2.7.2.3. Django**

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando construyes un sitio web, siempre necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

Hace tiempo que otros desarrolladores se dieron cuenta de que siempre se enfrentaban a los mismos problemas cuando construían sitios web, y por eso se unieron y crearon frameworks (Django es uno de ellos) con componentes listos para usarse.

Para entender para que sirve realmente Django, se debe observar en cómo funcionan los servidores. Lo primero es que el servidor necesita enterarse de que tu quieres que te sirva una página web.

Imagina un buzón (puerto) en el que alguien está constantemente mirando si hay cartas entrantes (peticiones). Esto es lo que hace un servidor web. El servidor web lee la carta, y envía una respuesta con la página web. Pero para enviar algo, tenemos que tener algún contenido. Y Django nos ayuda a crear ese contenido. (Twomey, 2017)

**a) Ventajas**

Las ventajas que ofrece codelgniter son: (Argudo, 2009)

- Es lo mejor de todo y también es lo que se lleva “casi” todo el crédito a la definición de Django. “Django es un framework web de alto nivel” eso esta muy claro en la definición, “Django es un framework web de alto nivel escrito en Python”. Gracias a esto Django hereda todas las características y facilidades que nos da Python, entre ellas escribir código bastante fácil de entender, y sobre todo te permite desarrollar aplicaciones muy rápidas y potentes.
- Rapidez: Django nació en un ambiente periodístico, donde se subían noticias muy rápido, y como los desarrolladores no pudieron estar a ese ritmo decidieron crear algo que sí lo haga, y así fue como nace Django , es por eso que ha sido estructurado de tal manera que tus aplicaciones web se crean muy rápidas.
- DRY: Django utiliza esta filosofía para no crear bloques de código iguales y fomentar la reutilización del mismo.

- Admin: Django es el único framework que “por defecto” viene con un sistema de administración activo, listo para ser utilizado sin ningún tipo de configuración.
- ORM: Para resumir esto, es una herramienta que te permite realizar consultas SQL a la Base de Datos, sin utilizar SQL.

**b) Desventajas**

- Es un poco mas difícil conseguir hosting.
- La compatibilidad Su puesta en producción es un tanto mas complicado.
- Si se viene programando en php a código limpio, la filosofía MVC complicara las cosas mientras la aprendes bien.
- Toda la documentación oficial de <https://code.djangoproject.com> es extremadamente buena, pero en ocasiones el ingles no genera una buena respuesta y los traductores no son muy buenos. Pero esta la comunidad que apoya mucho en google y duckduckgo se encuentra muy buenas referencias.  
(Twomey, 2017)

## 2.8. DISEÑO RESPONSIVO WEB

Actualmente en nuestro planeta existen más de 5000 millones de dispositivos conectados a Internet y debido a la gran demanda actual de terminales como smartphones, tablets, libros electrónicos, televisiones, videoconsolas, etc... esta evolución crece a pasos agigantados, concretamente según una infografía reciente de Intel , en el año 2020 se estima una cantidad de 31000 millones de dispositivos conectados a internet con 4000 millones de personas detrás de estos.  
(Alonso A. 2012)

“Responsive Web Design” es una filosofía o nuevo enfoque para solucionar los problemas de diseño para la gran diversidad de resoluciones y dispositivos. Este enfoque quiere centrarse en el contenido, y en el cliente, en su experiencia de usuario, si deja de trabajar con su equipo de sobremesa y quiere continuar navegando en la misma página web desde una tablet o smartphone.

“Responsive Web Design” quiere eliminar la necesidad de diseños diferentes y nuevos desarrollos para distintas resoluciones y por el contrario, sugiere que nuestro desarrollo debe dar soporte y responder a la necesidad del contexto sobre el que se esté ejecutando, teniendo en cuenta parámetros como el tamaño de pantalla, el tipo de dispositivo o la orientación. La página web debe de tener la capacidad de adaptarse a cada dispositivo, creando una solución única.

La creación del concepto como tal se le atribuye a Ethan Marcotte. Este describió técnicas y conceptos básicos que deben asumirse a la hora de implementar un diseño Responsive. Sin embargo, aunque se vera en este documento las técnicas y herramientas disponibles, Responsive Web Design es una nueva forma de pensar. (Vega, 20015)

A nivel implementación Responsive Web Design tiene tres conceptos claves. El primero de ellos es el uso de los Media Queries que ofrece CSS3 permitiendo aplicar estilos condicionalmente teniendo en cuenta parámetros de la pantalla. El segundo se trata del diseño web fluido, se trata de layouts definidos en porcentajes que se ajustan a los anchos de la pantalla. Y por último el tercer concepto se trata de los elementos fluidos dentro de estos layouts, como son las fuentes, las imágenes o elementos multimedia.

Al crear un sitio con Responsive Web Design solo se necesita una única versión de HTML y CSS que funcionará adecuadamente en cualquier tipo de dispositivo y resolución. Con Responsive Web Design se debe de dejar de ofuscar en que la web se vea idéntica en cada dispositivo. (Vega, 2015)

### **2.8.1. HTML**

El World Wide Web (WWW) es un sistema que contiene una cantidad de información casi infinita. Pero esta información debe estar ordenada de alguna forma de manera que sea posible encontrar lo que se busca. La unidad básica donde está almacenada esta información son las páginas Web. Estas páginas se caracterizan por contener texto, imágenes, animaciones... e incluso sonido y

video. (Ferrer, 2004)

Una de las características más importantes de las páginas Web es que son hipertexto. Esto quiere decir que las páginas no son elementos aislados, sino que están unidas a otras mediante los links o enlaces hipertexto. Gracias a estos enlaces el navegante de internet puede pulsar sobre un texto de una página para navegar hasta otra página. Será cuestión del programador de la página inicial decidir que palabras o frases serán activas y a donde conducirá pulsar sobre ellas. (Ferrer, 2004)

En algunos ambientes se discute que el World Wide Web ya no es un sistema hipertexto sino hipermedia. Los que defienden el cambio se apoyan en que aunque en sus orígenes el WWW constaba únicamente de texto en la actualidad es un sistema principalmente gráfico y se puede hacer que las zonas activas sean, no sólo texto, sino imágenes, videos, botones,... en definitiva cualquier elemento de una página. Aún así, el término original no ha sido reemplazado todavía y por tanto será

el empleado a lo largo de esta obra. (Ferrer, 2004)

¿Cómo se hace una página Web?

Cuando los diseñadores del WWW se hicieron esta pregunta decidieron que se debían cumplir, entre otras, las siguientes características:

- El Web tenía que ser distribuido: La información repartida en páginas no muy grandes enlazadas entre sí.
- El Web tenía que ser hipertexto y debía ser fácil navegar por él.
- Tenía que ser compatible con todo tipo de ordenadores (PCs, Macintosh, estaciones de trabajo...) y con todo tipo de sistemas operativos (Windows, MS-DOS, UNIX, MAC-OS).
- Debía ser dinámico: el proceso de cambiar y actualizar la información

Este lenguaje será el encargado de convertir un inocente archivo de texto inicial en

una página web con diferentes tipos y tamaños de letra, con imágenes impactantes, animaciones sorprendentes, formularios interactivos, etc. (Ferrer, 2004)

### **2.8.2. Bootstrap**

A parte de ser una gran herramienta para un desarrollador, Bootstrap es un Framework para desarrollo de interfaces de usuario que fue desarrollado por 2 de los desarrolladores de Twitter para asegurarse de que los proyectos tuvieran una apariencia consistente. (Muñoz, 2018)

Bootstrap es diferente. Su propósito no es crear ventanas de dialogo o efectos de desplazamiento (aunque realmente lo hace bastante bien). Bootstrap está diseñado para ayudar a las personas que no son diseñadoras, a equilibrar sus plantillas y diseños.

La parte interesante, es que ha sido desarrollado por desarrolladores y para desarrolladores. En la mayoría de los casos todo lo que se necesita es añadir una clase o algunos atributos de datos HTML. El sistema Grid predeterminado de Bootstrap ayudará a crear un sitio web responsive con mayor facilidad. (Muñoz, 2018)

### **2.8.3. CSS**

Oficialmente CSS nada tiene que ver con HTML5. CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento desarrollado para superar las limitaciones y reducir la complejidad de HTML. Al comienzo, atributos dentro de las etiquetas HTML proveían estilos esenciales para cada elemento, pero a medida que el lenguaje evolucionó, la escritura de códigos se volvió más compleja y HTML por sí mismo no pudo más satisfacer las demandas de diseñadores. En consecuencia, CSS pronto fue adoptado como la forma de separar la estructura de la presentación. Desde entonces, CSS ha crecido y ganado importancia, pero siempre desarrollado en paralelo, enfocado en

las necesidades de los diseñadores y apartado del proceso de evolución de HTML. (Gauchat, 2012)

La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que se menciona HTML5 también se hará referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas.

En este momento las nuevas características incorporadas en CSS3 están siendo implementadas e incluidas junto al resto de la especificación en navegadores compatibles con HTML5. (Gauchat, 2012)

#### **2.8.4. JavaScript**

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que se ve Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje Javascript como la mejor opción para la web. (Gauchat, 2012)

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, Javascript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares,

expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web. (Gauchat, 2012)

## **2.9. GESTORES DE BASES DE DATOS**

Tradicionalmente, para manipular grandes cantidades de datos aparecieron en primer lugar las bases de datos o "bancos de datos", cumpliendo de manera aproximada con la definición presentada en el apartado anterior. Se trataba de almacenar ordenadamente datos en un juego de ficheros, y, mediante unas aplicaciones informáticas y un sistema de índices, gestionarlas adecuadamente. Al aumentar la complejidad de estos bancos de datos (elevado número de ficheros y usuarios, acceso múltiple y simultáneo a los ficheros, aumento del número de registros, etc.) se producen problemas cada vez más graves para asegurar la consistencia, la integridad y la accesibilidad de los datos, produciendo problemas de eficiencia en el tratamiento de los datos.

Para resolver estos problemas aparecen en el mercado los Sistemas de Gestión de Bases de Datos (SGBD), que convierten el acceso a los datos y su gestión en una aplicación cerrada ("caja negra"), interponiéndose entre los usuarios y los ficheros, y haciéndose cargo de todos los problemas de explotación, mantenimiento y comprobación de los datos. De esta manera el usuario pierde de vista todos los detalles relativos al almacenamiento físico de los datos tratando con ellos sólo a través de un lenguaje conceptual sencillo. (Castañeda, 2014)

Manipulación física de los datos. El usuario pierde de vista totalmente cómo y dónde se almacenan los datos, ocupándose el motor del sistema de gestión de todas estas tareas (Castañeda, 2014)

Visión del usuario: manipulación lógica de los datos. El usuario maneja la base de datos a través de un entorno organizado mediante menús o aplicaciones informáticas que le guían sin necesidad de conocimientos de la estructura de la base de datos (Castañeda, 2014)

Descripción de la base de datos mediante un lenguaje conceptual que permiten una visión abstracta del sistema. Se entiende por lenguaje conceptual un lenguaje formado por palabras (comandos) de fácil memorización que permiten efectuar todas las tareas de acceso y manipulación de los datos, así como las tareas de mantenimiento de todo el sistema de gestión.

El lenguaje conceptual del sistema de gestión permite abordar de manera todas las tareas para el mantenimiento y explotación del sistema de gestión. (Castañeda, 2014)

### **2.9.2. Sistemas gestores de base de datos**

#### **2.9.2.1. SQL Server**

Microsoft SQL Server 2005 es una plataforma de base de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico; es también una plataforma de Business Intelligence para soluciones de integración, análisis y creación de informes de datos. (Microsoft Technet, 2013)

SQL Server es un multi-componente del sistema de administración de base de datos centrada en torno a un alto rendimiento, alta disponibilidad de motor de bases de datos. (Microsoft Technet, 2013)

La calidad del motor de base de datos SQL Server es crucial para la fiabilidad del SQL Server en el manejo de grandes cantidades de datos. Sin embargo es mucho más que un motor de base de datos y se compone de un conjunto de herramientas y componentes que se apoya el diseño, gestión, mantenimiento y programación. (Microsoft Technet, 2013)

##### **a) Ventajas**

Las ventajas de utilizar SQL Server son: (Celma & Casamayor, 2003)

- Facilidad de instalación, distribución y utilización.

- Almacenamiento de datos.
- Seguridad de alto nivel
- Integración con internet
- Incluye un potente entorno grafico de administración

**b) Desventajas**

Las desventajas de utilizar SQL Server son: (Celma & Casamayor, 2003)

- Sus versiones difieren en los requerimientos para su instalación según la versión del sistema operativo Windows.
- No tiene soporte en otras plataformas, como ser Linux, Mac OS, Solaris, etc.
- La configuración de protocolos y usuarios es diferente en sus dos últimas ediciones.

**2.9.2.2. MySQL**

MySQL es un sistema de administración de bases de datos (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organizaciones administrativas.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema, proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutados en casi todos los sistemas operativos. (Microsoft Technet, 2013)

**a) Características**

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas (Microsoft Technet, 2013)

**b) Ventajas**

- Es rápida, la cualidad más destacada por quienes desarrollan MySQL es su velocidad y así como el software fue diseñado desde un principio, pensando principalmente en la rapidez.
- MySQL es gratuito.
- Fácil de usar se puede construir e interactuar con una base de datos MySQL, siguiendo simples reglas en el lenguaje SQL que es el lenguaje estándar para la comunicación con los sistemas de gestión de bases de datos (RDBMS).
- Se puede ejecutar en muchos sistemas. MySQL corre en muchos sistemas operativos, Windows, Linux, Mac OS, la mayoría de las variedades de Unix (incluyendo Solaris y AIX), FreeBSD, OS / 2, Irix, y otros.
- Está disponible en casi todos los proveedores de hosting, si se quiere alojar el sitio web en una compañía de web hosting , de seguro se tendrá disponible MySQL sin costo adicional.

- Tiene soporte técnico ampliamente disponible, su gran base de usuarios ofrece soporte gratuito a través de listas de correo. Los desarrolladores de MySQL también participan en estas listas de correo electrónico.
- Es seguro, el sistema flexible de autorización de MySQL, permite a algunos o todos los privilegios de base de datos (como el privilegio de crear o eliminar una base de datos) a usuarios específicos o grupos de ellos. Las contraseñas están cifradas.
- Soporta bases de datos de gran tamaño, MySQL maneja bases de datos de hasta 50 millones de filas o más. El límite de tamaño de archivo predeterminado para una tabla es de 4 GB, pero este se puede incrementar (si el sistema operativo puede manejarlo) a un teórico límite de 8 millones de terabytes (TB).
- Es personalizable, a licencia GPL de código abierto permite a los programadores modificar el software MySQL para adaptarse a sus propios entornos específicos.

**c) Desventajas**

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

**2.9.2.3.PostgreSQL**

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL, similar a la BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. (Microsoft Technet, 2013)

**a) Características**

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos

sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.

- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autentificación
- Acceso encriptado vía SSL

**b) Ventajas**

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Footprint bajo de memoria, bastante poderoso con una configuración adecuada.
- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

**c) Desventajas**

- En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
- Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
- Consumo más recursos que MySQL.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

## **CAPITULO 3: MARCO PRÁCTICO**

### **3. INGENIERÍA DEL PROYECTO**

#### **3.1. MARCO METODOLÓGICO**

##### **3.1.1. Enfoque de la Investigación**

En el presente proyecto se aplicará una metodología cualitativa debido a que se utilizaran técnicas como entrevistas abiertas, técnicas de observación, observación participante, etc. Además de que se estudiarán cualidades y características del objeto de estudio, mediante la observación del participante

##### **3.1.2. Tipo de investigación**

El tipo de investigación será: investigación descriptiva ya que el objeto de estudio será sometido a análisis obteniendo características, situación, eventos. Para poder encontrar las propiedades importantes, dominios y componentes, se definirá a quienes y que se medirá.

##### **3.1.3. Método de Investigación**

El método de investigación que se empleara será el método inductivo debido a que se realizara una primera etapa de observación, posteriormente un análisis, conclusión de los hechos, para poder formular una hipótesis para brindar una solución al problema planteado.

### 3.1.4. Diseño de la investigación junto con las herramientas

**TABLA 4: Diseño de la investigación junto con las herramientas**

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA	INSTRUMENTOS
Diseñar el modelado de negocio actual y alternativo en base a características workflow	<ul style="list-style-type: none"> <li>• Recolectar información referente a los procesos actuales de la gestión de información de la dirección de Posgrado</li> <li>• Modelar el flujo de información de los procesos actuales</li> <li>• Analizar el proceso de modelado actual.</li> <li>• Realizar el modelado de proceso actual</li> <li>• Identificar las deficiencias existentes en los procesos actuales</li> <li>• Identificar componentes de workflow o flujo de trabajo</li> <li>• Diseñar el flujo de trabajo utilizando las características de workflow</li> <li>• Elaborar el modelado de proceso alternativo</li> <li>• Seleccionar de metodología adecuada para el desarrollo</li> <li>• Planificar metodología en base a la metodología seleccionada</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas de recolección de información</li> <li>• Ingeniería de software</li> <li>• Workflow</li> </ul>	<ul style="list-style-type: none"> <li>• Cuestionario</li> <li>• Observación</li> <li>• Entrevista</li> <li>• Diagrama de actividades</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA	INSTRUMENTOS
Desarrollar el módulo de registro de usuarios con roles privilegios respectivos.	<ul style="list-style-type: none"> <li>• Seleccionar el entorno de desarrollo</li> <li>• Seleccionar de lenguaje de programación</li> <li>• Seleccionar de framework adecuado</li> <li>• Seleccionar de gestor de base de datos</li> <li>• Seleccionar de patrón de software</li> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar los diagramas UML del módulo de registro de usuarios</li> <li>• Diseñar la base de datos para los usuarios del sistema</li> <li>• Diseñar la arquitectura utilizando características de SOA</li> <li>• Implementar el módulo de registro de usuarios empleando herramientas para el desarrollo responsive</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Técnicas de recolección de información</li> <li>• Tecnologías de desarrollo</li> <li>• Gestor de base de datos</li> <li>• Diseño responsivo web</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Metodologías de desarrollo</li> <li>• Herramientas CASE</li> <li>• Entorno de desarrollo</li> <li>• Lenguaje de programación</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA	INSTRUMENTOS
Desarrollar el módulo de admisión de documentación docente, cursante y almacenamiento	<ul style="list-style-type: none"> <li>• Analizar requerimientos del módulo de admisión de documentación</li> <li>• Analizar el tipo de documentación que será almacenada en formato digital mediante escaneo.</li> <li>• Seleccionar los documentos de acuerdo a requerimientos de la dirección de posgrado que serán almacenados.</li> <li>• Diseñar diagramas UML necesarios para el módulo de admisión de documentación.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar de servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el módulo para la admisión de documentación seleccionada de docentes y cursantes aplicando características de workflow.</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Técnicas de recolección de información</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Gestor de base de datos</li> <li>• Diseño responsive web</li> <li>• Workflow</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Metodologías de desarrollo</li> <li>• Herramientas CASE</li> <li>• Entorno de desarrollo</li> <li>• Lenguaje de programación</li> <li>• Framework</li> <li>• Gestor de base de datos</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA	INSTRUMENTOS
Desarrollar el módulo de generación de código único para cada documento presentado	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Analizar algoritmos de generación de código único</li> <li>• Seleccionar el algoritmo de generación de código único</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Diseñar servicios de la api Rest</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Implementar el código para la generación de código único aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Seguridad de información</li> <li>• Gestor de base de datos</li> <li>• Diseño responsive web</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Metodologías de desarrollo</li> <li>• Herramientas CASE</li> <li>• Entorno de desarrollo</li> <li>• Lenguaje de programación</li> <li>• Framework</li> <li>• Gestor de base de datos</li> </ul>
Desarrollar el módulo de inclusión de código QR a la documentación presentada.	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para el desarrollo del módulo</li> <li>• Diseñar diagramas UML necesarios para el módulo.</li> <li>• Actualizar las tablas de la base de datos</li> <li>• Actualizar el diseño de la arquitectura</li> <li>• Diseñar servicios de la api Rest</li> <li>• Parametrizar contenido de código QR en base a mecanismos criptográficos</li> <li>• Implementar el código para generación de código QR aplicando características de workflow</li> <li>• Realizar pruebas al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Tecnologías de desarrollo</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Seguridad de información</li> <li>• Diseño responsive web</li> <li>• Workflow</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Metodologías de desarrollo</li> <li>• Herramientas CASE</li> <li>• Entorno de desarrollo</li> <li>• Lenguaje de programación</li> <li>• Framework</li> <li>• Gestor de base de datos</li> <li>• Técnicas criptográficas</li> </ul>

OBJETIVOS ESPECÍFICOS	ACCIONES	FUNDAMENTACIÓN TEÓRICA	INSTRUMENTOS
Generar reportes en base a los módulos implementados	<ul style="list-style-type: none"> <li>• Analizar los requerimientos para la elaboración de consultas.</li> <li>• Diseñar los diagramas UML necesarios</li> <li>• Diseñar servicios de la api Rest</li> <li>• Implementar el módulo de obtención de reportes en base a los requerimientos establecidos en el flujo de trabajo.</li> <li>• Generar la impresión de reportes en otros formatos</li> <li>• Realizar pruebas funcionales al módulo</li> </ul>	<ul style="list-style-type: none"> <li>• Ingeniería de software</li> <li>• Arquitectura Orientada a Servicios</li> <li>• Workflow</li> <li>• Diseño responsive web</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Metodologías de desarrollo</li> <li>• Herramientas CASE</li> <li>• Entorno de desarrollo</li> <li>• Lenguaje de programación</li> <li>• Framework</li> <li>• Gestor de base de datos</li> </ul>
Realizar pruebas al sistema concluido.	<ul style="list-style-type: none"> <li>• Seleccionar los tipos de pruebas adecuados al proyecto.</li> <li>• Aplicar las pruebas seleccionadas</li> <li>• Documentar las pruebas realizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas e instrumentos de recolección de datos</li> <li>• Ingeniería de software</li> </ul>	<ul style="list-style-type: none"> <li>• Diagrama UML</li> <li>• Lenguaje de programación</li> <li>• Gestor de base de datos</li> </ul>

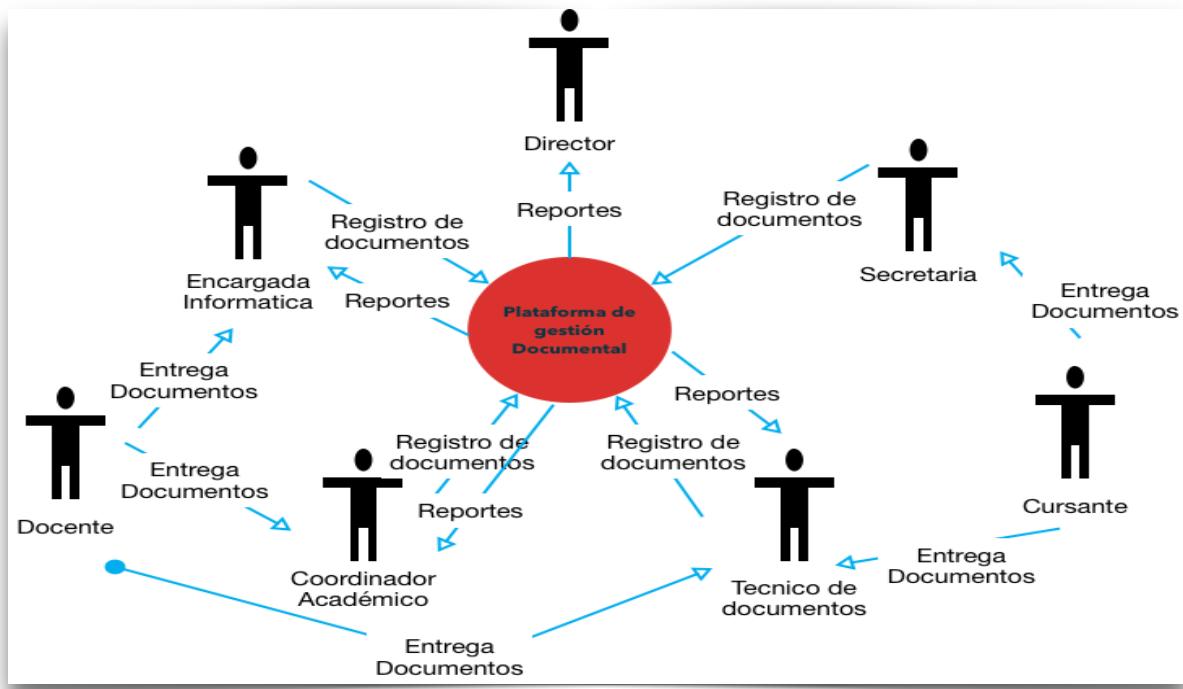
Fuente: Elaboración propia, 2020

### 3.2. DISEÑO DEL MODELO DE NEGOCIO ACTUAL Y ALTERNATIVO EN BASE A CARACTERISTICAS WORKFLOW

#### 3.2.1. Análisis de requerimientos

En base a las entrevistas y cuestionarios obtenidos se realizó el análisis de requerimientos obteniendo el siguiente diagrama de contexto

**FIGURA 10: Diagrama de contexto**

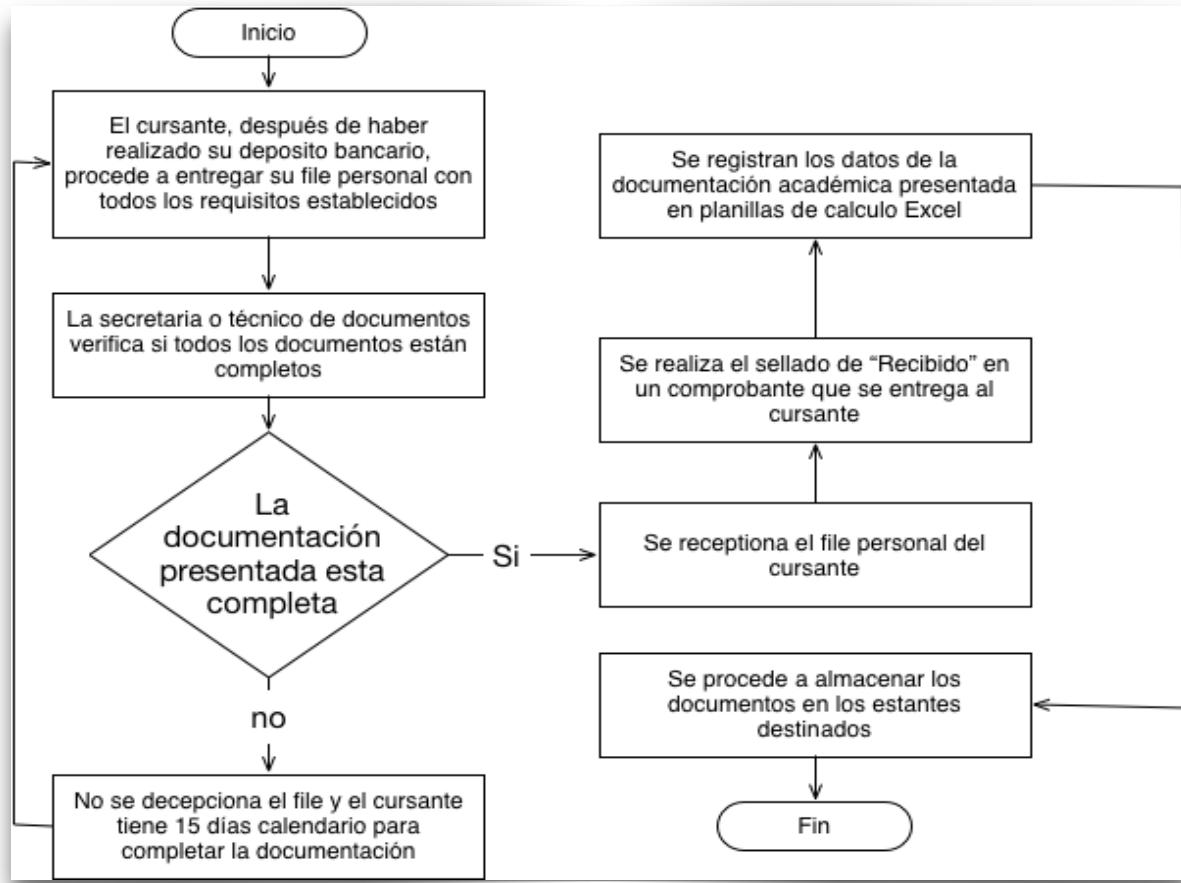


*Fuente: Elaboración propia, 2020*

### 3.2.2. Modelado de negocio actual.

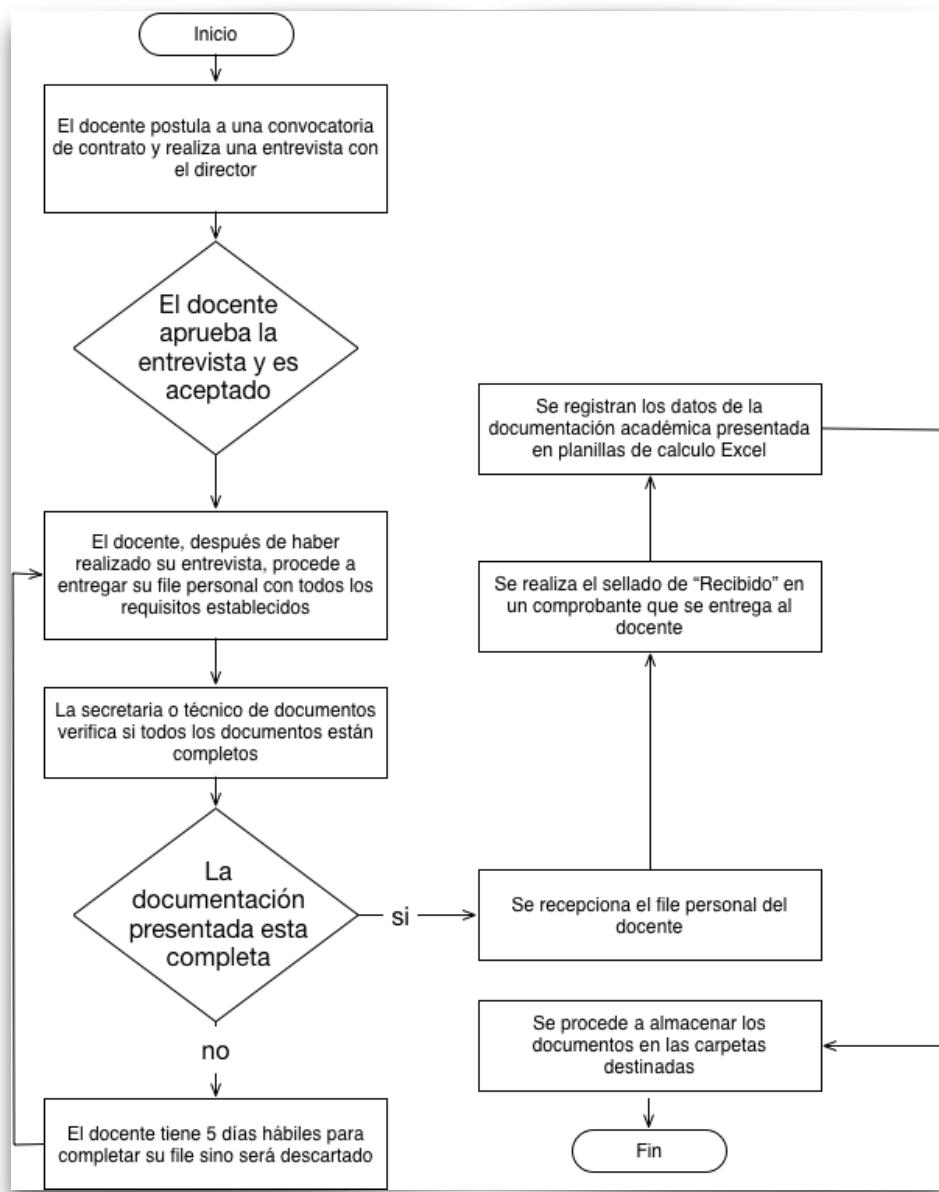
En base a la información obtenida mediante entrevistas y cuestionarios realizados al personal de la Dirección de Posgrado, y en base a la observación de los procedimientos actuales para la gestión de documentación académica se obtuvo el siguiente modelado de negocio actual:

**FIGURA 11: Modelado de negocio actual documentación cursantes**



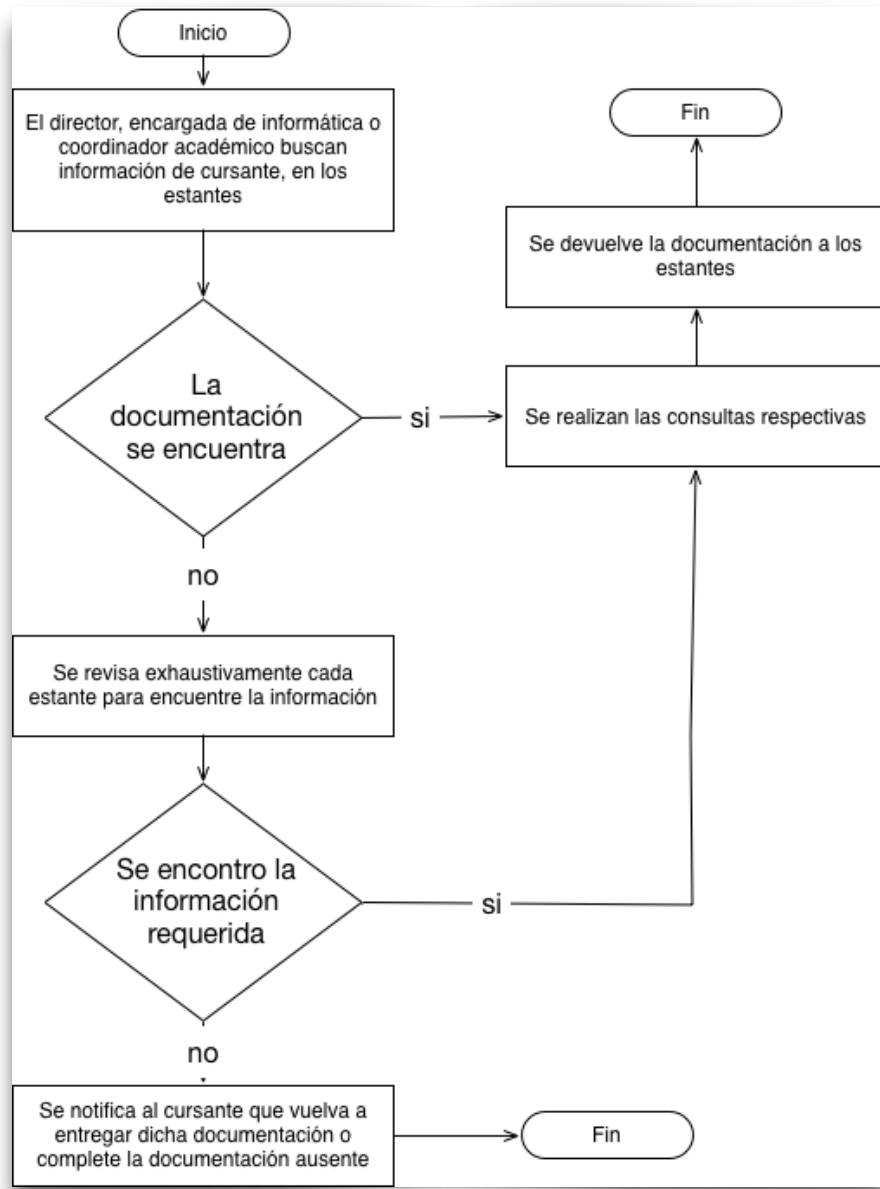
Fuente: Elaboración propia, 2020

**FIGURA 12: Modelado de negocio actual documentación docentes**



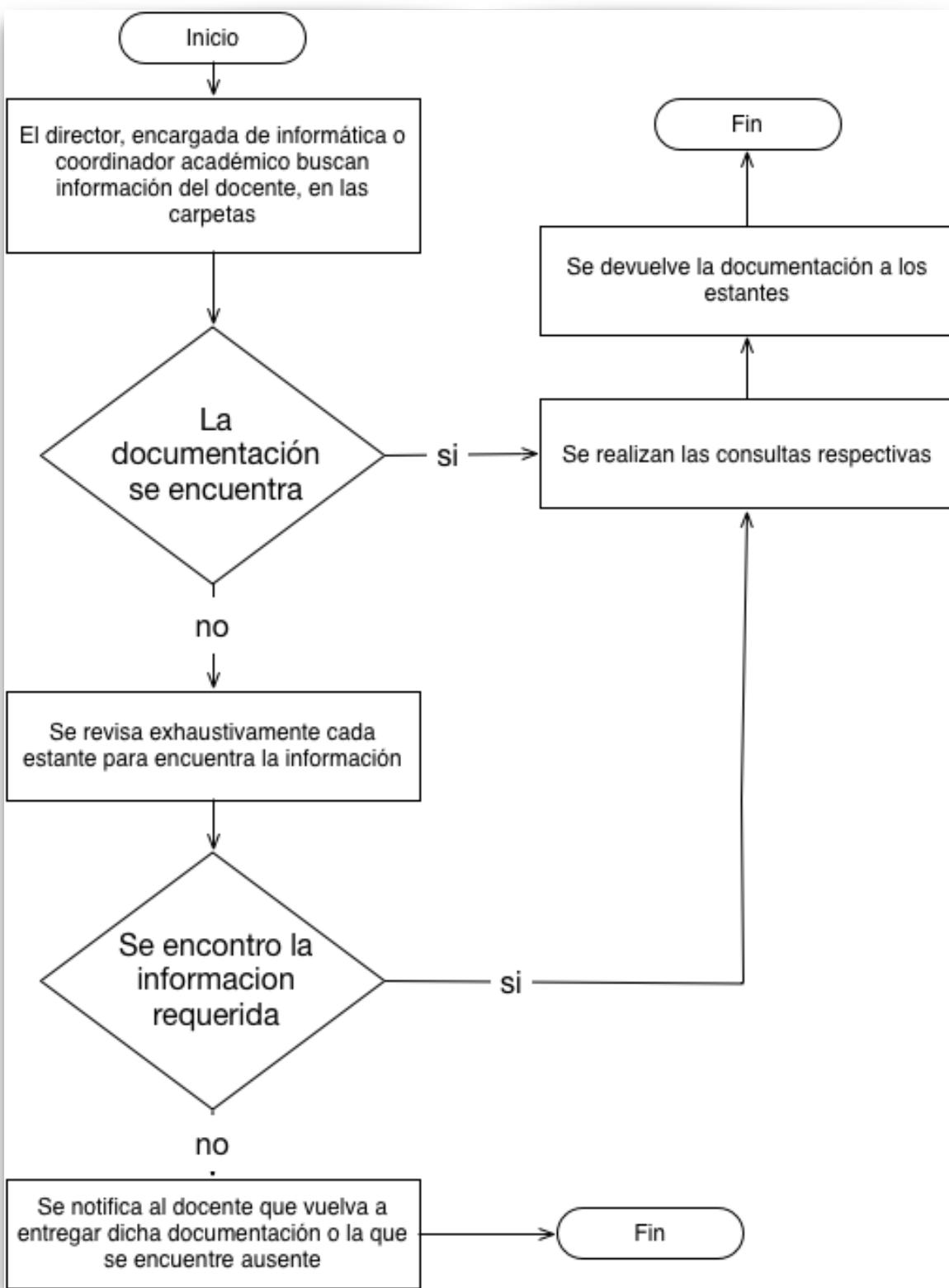
Fuente: Elaboración propia, 2020

**FIGURA 13: Modelado de negocio actual consulta documentación cursantes**



Fuente: *Elaboración propia, 2020*

**FIGURA 14: Modelado de negocio actual consulta documentación docentes**



Fuente: Elaboración propia, 2020

### 3.2.3. Identificación de deficiencias

**TABLA 5: Identificación de deficiencias**

PROCESO	CARACTERISTICAS	DEFICIENCIAS
Admisión de documentación de cursantes y docentes de diferentes programas de la dirección de posgrado	<ul style="list-style-type: none"> <li>• El cursante debe entregar toda la documentación requerida después de haber realizado el depósito bancario para su inscripción</li> <li>• Los documentos requeridos son: formularios de admisión, contrato compromiso de pago, certificado de nacimiento original, fotocopia simple de cédula de identidad, fotocopia de factura de luz, croquis de domicilio, currículum vitae documentado, copia legalizada de título en provisión nacional, copia legalizada del título de maestría (para doctorado)</li> <li>• La secretaría revisa que la documentación entregada esté completa, si la documentación no está completa se recepciona, y se solicita que en los próximos días complete la documentación</li> <li>• La secretaría entrega un registro de file de ingreso de cursantes, en el que se selecciona qué documentos están presentes (Ver anexo "D")</li> <li>• La secretaría guarda los files de los cursantes en los estantes</li> </ul>	<ul style="list-style-type: none"> <li>• Al aceptar files con ausencia de algunos documentos, con el tiempo los cursantes no entregan los documentos faltantes</li> <li>• Muchas veces cuando la secretaría no se encuentra otro funcionario de posgrado de buena voluntad recepciona estos documentos, al momento de que la secretaría evalúa los documentos entregados, muchas veces algunos están ausentes causando un problema para la secretaría</li> <li>• El manejo de información académica con datos sensibles no es el más óptimo por lo que mucho de estos pueden ser hurtados, la garantía de confidencialidad no es la mejor</li> <li>• El registro de la documentación entregada por los docentes se las realizan de manera manual</li> <li>• Los comprobantes de entrega de documentos muchas veces son perdidos</li> <li>• </li> </ul>

PROCESO	CARACTERISTICAS	DEFICIENCIAS
Búsqueda de información de docentes o cursantes	<ul style="list-style-type: none"> <li>• La dirección de posgrado regularmente es evaluada por el director o personal de acreditación académica con el fin de que los procesos que se realizan internamente sean los mas correctos posibles</li> <li>• Se evalúan las carpetas de los docentes y cursantes</li> <li>• Cada programa de posgrado debe tener incluido los currículums y títulos académicos de los docentes que imparten los diferentes modelos de los distintos programas</li> <li>• Se consulta las carpetas de los docentes para encontrar cierta información requerida</li> <li>• Las consultas se las hacen de manera física, extrayendo las carpetas o files de los estantes que los contienen</li> <li>• Cualquier empleado de posgrado tiene acceso a estos documentos</li> </ul>	<ul style="list-style-type: none"> <li>• La búsqueda de información de un docente o estudiante demanda un tiempo considerable</li> <li>• Al realizarse las consultas de manera física, estos documentos están expuestos al entrepapelado con otros documentos que el personal de posgrado maneja</li> <li>• El extravío de documentation academica esta latente</li> <li>• Cuando no se encuentra la documentación se revisa exhaustivamente toda la documentación disponible</li> <li>• Percatarse de que los documentos están entrepapelados o extraviados es una tarea muy difícil</li> <li>• Al no encontrar la documentación requerida se solicita al cursante o docente que este vuelve a entregar dicha documentación</li> </ul>

Fuente: *Elaboracion propia 2020*

### 3.2.4. Identificación de componentes workflow flujo de trabajo

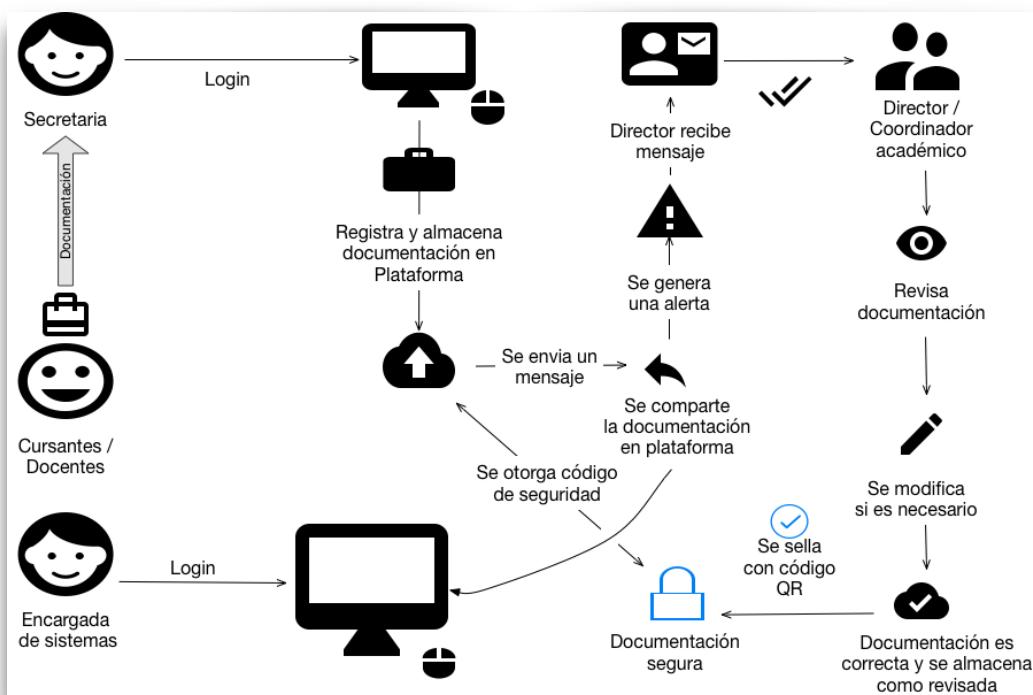
FIGURA 15: Identificación de componentes workflow



Fuente: Elaboración propia 2020

En base a la figura anterior se definen los componentes de workflow para el proyecto, el mismo se detalla a continuación.

FIGURA 16: Definición de componentes workflow



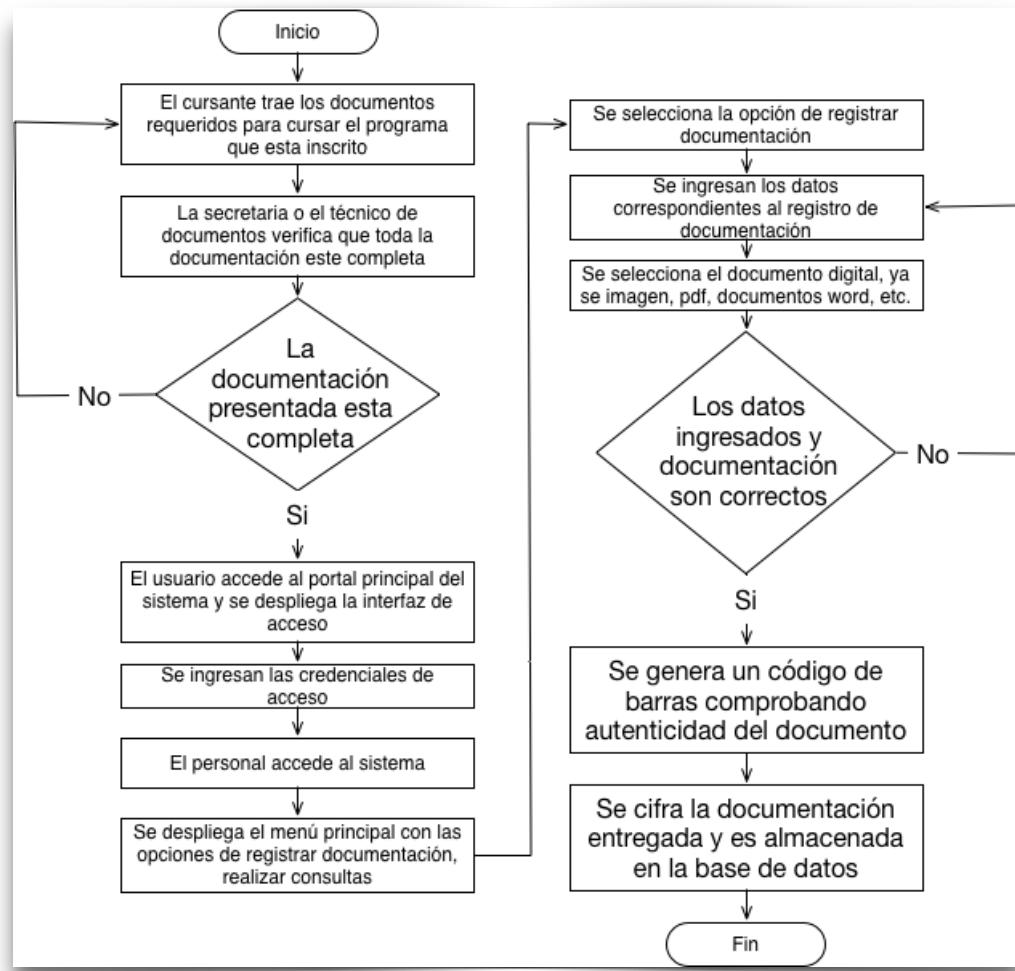
Fuente: Elaboración propia 2020

### 3.2.5. Modelado del diagrama de flujo alternativo.

En base a la observación realizada internamente dentro de la dirección de posgrado y un previo análisis al modelo de proceso actual, se plantea el siguiente flujo alternativo partiendo de la base de las características de la arquitectura orientada a servicios y sistemas workflow.

#### 3.2.5.1. Modelado del diagrama de flujo alternativo entrega documentación de cursantes.

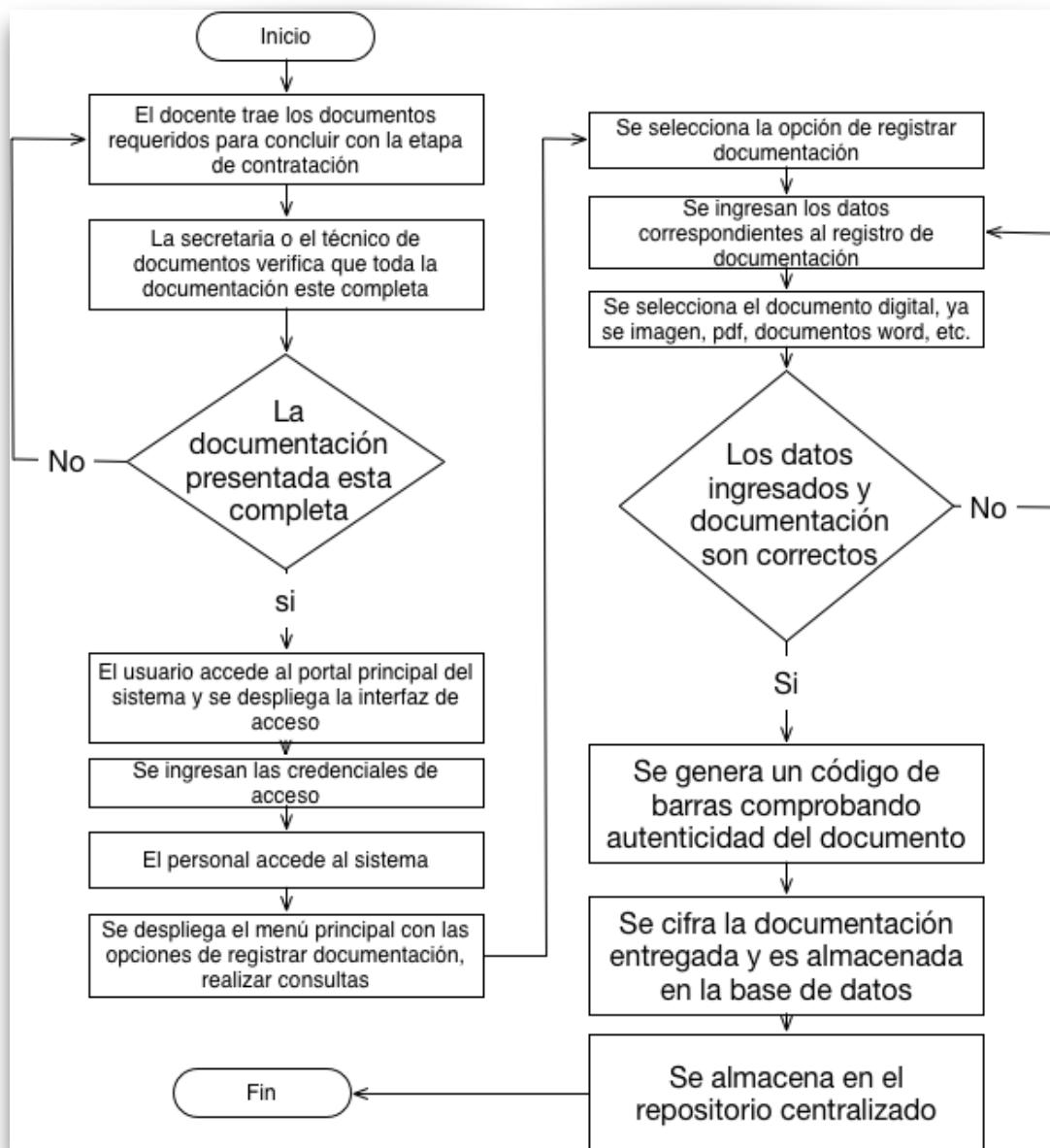
FIGURA 17: Proceso alternativo de entrega de documentación de cursantes



Fuente: Elaboración propia 2020

### 3.2.5.2. Modelado del diagrama de flujo alternativo entrega documentación de docentes.

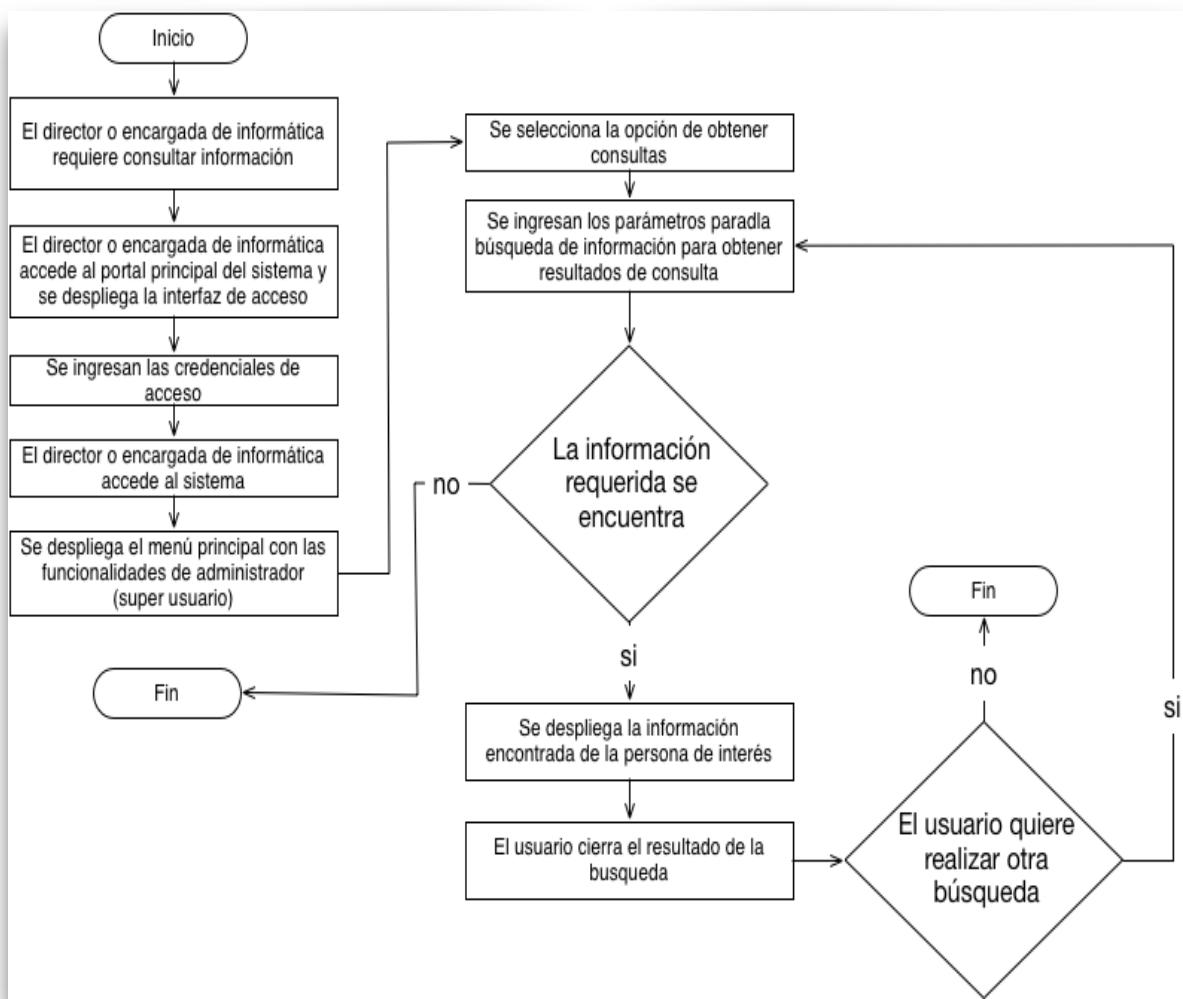
**FIGURA 18: Proceso alternativo de entrega de documentación de docentes**



Fuente: Elaboración propia 2020

### 3.2.5.3. Modelado del diagrama de flujo alternativo generación de reportes.

**FIGURA 19: Proceso alternativo de generación de reportes**



Fuente: Elaboración propia 2020

### 3.2.6. Selección de metodología de desarrollo

Después de haber analizado los diferentes modelos de desarrollo de software identificados en el marco teórico, a continuación se muestra una comparación de ventajas y desventajas que presenta cada uno de los modelos:

**TABLA 6: Comparativa de modelos de desarrollo de software**

MODELO	VENTAJAS	DESVENTAJAS
<b>Scrum</b>	<ul style="list-style-type: none"> <li>• Los clientes pueden participar en cada una de las iteraciones y proponer soluciones. De hecho, el proceso en su conjunto está pensado para un tipo de evaluación conjunta.</li> <li>• Cada iteración arroja una serie de resultados. No es necesario, por tanto, que el cliente espere hasta el final para ver el producto.</li> <li>• Se adapta a cualquier contexto, área o sector de la gestión. No es una técnica exclusiva de ninguna disciplina.</li> <li>• Del mismo modo, los riesgos que pueden afectar a un proyecto son gestionados en el mismo momento de su aparición. La intervención de los equipos de trabajo es inmediata.</li> </ul>	<ul style="list-style-type: none"> <li>• No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia.</li> </ul>
<b>Extreme Programming XP</b>	<ul style="list-style-type: none"> <li>• Da lugar a una programación sumamente organizada.</li> <li>• Ocasiona eficiencias en el proceso de planificación y pruebas.</li> <li>• Cuenta con una tasa de errores muy pequeña.</li> <li>• Propicia la satisfacción del programador.</li> <li>• Fomenta la comunicación entre los clientes y los desarrolladores.</li> <li>• Facilita los cambios.</li> <li>• Permite ahorrar mucho tiempo y dinero.</li> <li>• Puede ser aplicada a cualquier lenguaje de programación.</li> <li>• El cliente tiene el control sobre las prioridades.</li> <li>• Se hacen pruebas continuas durante el proyecto.</li> <li>• La XP es mejor utilizada en la implementación de nuevas tecnologías</li> <li>• </li> </ul>	<ul style="list-style-type: none"> <li>• Es recomendable emplearla solo en proyectos a corto plazo.</li> <li>• En caso de fallar, las comisiones son muy altas.</li> <li>• Requiere de un rígido ajuste a los principios de XP.</li> <li>• Puede no siempre ser más fácil que el desarrollo tradicional.</li> <li>• </li> </ul>

MODELO	VENTAJAS	DESVENTAJAS
Kanban	<ul style="list-style-type: none"> <li>• Facilidad de entendimiento y exposición de información a todos los miembros involucrados.</li> <li>• Facilidad de integración con metodologías ágiles (Scrumban).</li> <li>• El más adecuado para proyectos que se encuentran en mejora continua.</li> <li>• No todas las desventajas de Kanban surgidas en manufactura aplican en la industria del software puesto que la implementación interna es diferente.</li> <li>• Acepta el ingreso de cambios a último momento con facilidad.</li> </ul>	<ul style="list-style-type: none"> <li>• Menor efectividad en situaciones de Recursos Compartidos: Las órdenes no frecuentes vuelven inefficientes a kanban ya que se tiene que asegurar una producción suficiente por parte de un proceso mientras que a su vez el proceso que es no-frecuente es ejecutado.</li> <li>• Kanban asume sistemas de producción repetitivos dada la naturaleza de su creación en el área de manufactura.</li> <li>• Calidad: El sistema Kanban lleva los niveles de inventario cerca de 0 lo que en caso de alta incertidumbre e interrupciones en la red de transporte representa un peligro ya que eso significa que los clientes se quedan sin suministro de partes.</li> </ul>

Fuente: Elaboración propia, 2020

Se optará por utilizar la metodología de desarrollo de software de programación extrema XP (Xtreme programing) debido a que las ventajas que presenta con respecto a la metodología Scrum son: cuenta con una tasa de errores pequeños, fomenta la comunicación entre clientes y desarrolladores, facilita los cambios en cualquier etapa del proyecto, puede ser aplicado a cualquier lenguaje de programación, la programación extrema es excelente para la aplicación de nuevas tecnologías. Los requisitos del proyecto podrán cambiar por lo cual se debe ser flexible en cuanto a cambios en los requerimientos, el acceso a la información que

brinda la empresa, el proyecto se dividirá en entregables, ademas de que las pruebas serán realizadas en cada iteración, las actividades como la programación en parejas propias de la metodología XP se descarta debido a que solo es una persona quien desarrollara el proyecto.

Se descarta utilizar la metodología de desarrollo Scrum debido a que esta metodología demanda para la buena aplicación de la misma, experiencia en los equipos de trabajo, por lo cual grupos junior que estén en proceso de formación aplicando esta metodología no se obtendrá el mejor resultado esperado, gran parte del éxito de la aplicación de la metodología Scrum radica en la experiencia que aportan los profesionales de los equipos de trabajo.

### **3.2.7. Planificación de metodología**

#### **3.2.7.1. Planificación de los incrementos del sistema en base al método de desarrollo seleccionado**

En base a la metodología seleccionada se plantea las siguientes historias de usuarios:

#### **3.2.7.2. Identificación de Usuarios**

- **Director:** Involucra al director de la dirección de posgrado, el cual revisa la documentación de los docentes potencialmente a impartir clases dentro de posgrado, ademas de verificar eventualmente la documentación de algún cursante del cual este tenga interés
- **Coordinador Académico:** Es quien hace de conexión entre el personal civil y el director para los distintos procesos administrativos dentro posgrado, eventualmente consulta información de docentes o cursantes
- **Encargada de sistemas:** Es quien supervisa las distintas aperturas de programas, hace seguimiento a los docentes que imparten los distintos modelos, ademas de hacer seguimiento a cursantes, constantemente requiere

reportes de docentes y estudiantes, ademas de ordenar la documentación académica de cursantes y docentes

- **Secretaria:** Es quien recepciona la documentación de docentes potenciales a impartir clases, derivar estos documentos al director para que este los revise, ademas de recepcionar la documentación de los cursantes, haciendo seguimiento de que estos cursantes entreguen sus documentos a cabalidad, registrandolos en el “Registro: Files de ingreso de cursantes” (Ver Anexo D).

**FIGURA 20: Identificación de Usuarios**



Fuente: Elaboración propia, 2020

### 3.2.7.3. Historias de Usuario

**TABLA 7: Tabla de descripción de historias de usuarios**

ID	QUIEN	NECESITO	PARA
1	Director	Crear Usuarios	Poder registrar usuarios, para que estos puedan acceder al sistema, ademas de que cuando elimine un usuario no se elimine completamente de la base de datos, si no que este registrado pero ya no tenga acceso a la plataforma, ademas de que cada usuario tenga credenciales de acceso.
2	Director	Otorgar roles y privilegios	Los usuarios ya registrados no deben tener acceso todo el sistema, solo deberían poder ver e interactuar con los módulos que a cada empleado le compete.
3	Coordinador académico	Crear Usuarios	Poder registrar usuarios, para que estos puedan acceder al sistema, ademas de que cuando elimine un usuario no se elimine completamente de la base de datos, si no que este registrado pero ya no tenga acceso a la plataforma, ademas de que cada usuario tenga credenciales de acceso.
4	Coordinador académico	Otorgar roles y privilegios	Los usuarios ya registrados no deben tener acceso todo el sistema, solo deberían poder ver e interactuar con los módulos que a cada empleado le compete.

ID	QUIEN	NECESITO	PARA
5	Director	Admisión de documentación de cursantes y docentes	Ya dentro del sistema se debe poder desplegar un menú donde pueda registrar la información del acreedor de la información ya sea del cursante o docente, especificando la fecha y hora del registro.
6	Director	Admisión de documentación de cursantes, docentes y almacenamiento	Después de haber introducido la información del acreedor de la documentación debo poder almacenar el file en la base de datos del sistema.
7	Encargada de sistemas	Admisión de documentación de cursantes y docentes	Ya dentro del sistema se debe poder desplegar un menú donde pueda registrar la información del acreedor de la información ya sea del cursante o docente, especificando la fecha y hora del registro.
8	Encargada de sistemas	Admisión de documentación de cursantes, docentes y almacenamiento	Después de haber introducido la información del acreedor de la documentación debo poder almacenar el file en la base de datos del sistema.
9	Secretaria	Admisión de documentación de cursantes y docentes	Ya dentro del sistema se debe poder desplegar un menú donde pueda registrar la información del acreedor de la información ya sea del cursante o docente, especificando la fecha y hora del registro.
10	Secretaria	Admisión de documentación de cursantes, docentes y almacenamiento	Después de haber introducido la información del acreedor de la documentación debo poder almacenar el file en la base de datos del sistema.
11	Secretaria	Generación de código único	Después de haber registrado los datos del acreedor y almacenar el file del acreedor de la documentación, se debe generar un código de seguridad asignado al registro de la documentación.
12	Director	Inclusión de código QR a la documentación presentada	Después de haberse registrado los datos del acreedor y almacenar el file del acreedor de la documentación, la documentación debe derivarse a otro menú para que esta documentación sea revisada, posterior a revisarse y que todo este a cabalidad, se incrustara el código QR parametrizado con el código único, siendo este un aval de garantía de veracidad de los datos y documentación.
13	Coordinador académico	Inclusión de código QR a la documentación presentada	Después de haberse registrado los datos del acreedor y almacenar el file del acreedor de la documentación, la documentación debe derivarse a otro menú para que esta documentación sea revisada, posterior a revisarse y que todo este a cabalidad, se incrustara el código QR parametrizado con el código único, siendo este un aval de garantía de veracidad de los datos y documentación.

ID	QUIEN	NECESITO	PARA
14	Director	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual "Revisada" o "No revisada", ademas de desplegar la documentación Revisada en un menú y la No Revisada en otro menú, ademas de poder obtener un reporte de los usuarios registrados, sus roles y privilegios respectivos.
15	Coordinador académico	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual "Revisada" o "No revisada", ademas de desplegar la documentación Revisada en un menú y la No Revisada en otro menú, ademas de poder obtener un reporte de los usuarios registrados, sus roles y privilegios respectivos.
16	Encargada de sistemas	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual "Revisada" o "No revisada"
17	Secretaria	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual "Revisada" o "No revisada"

Fuente: *Elaboración propia, 2020*

### 3.2.7.2. Realelease del plan XP

**TABLA 8: Realelease plan XP**

Entregable	PRIORIDAD	Quien	Necesito	Para	EDSUFRAZON		Fecha entregable		
					E	D	S	U	F
1. Registro de usuarios con roles y privilegios	1 ALTA 3 ALTA	Director, coordinador académico	Crear Usuarios	Poder registrar usuarios, para que estos puedan acceder al sistema, ademas de que cuando elimine un usuario no se elimine completamente de la base de datos, si no que este registrado pero ya no tenga acceso a la plataforma, ademas de que cada usuario tenga credenciales de acceso.		5	1	15 de febrero	

Entregable	PRIORIDAD	Quien	Necesito	Para	EDISON	Fecha entregable
1. Registro de usuarios con roles y privilegios	2 ALTA	Director, coordinador académico	Otorgar roles y privilegios	Los usuarios ya registrados no deben tener acceso todo el sistema, solo deberían poder ver e interactuar con los módulos que a cada empleado le compete.	51	15 de febrero
2. Admisión de documentación de cursantes y docentes	5 ALTA 7 ALTA 9 ALTA	Director, encargada de sistemas y secretaria	Admisión de documentación de cursantes y docentes	Ya dentro del sistema se debe poder desplegar un menú donde pueda registrar la información del acreedor de la información ya sea del cursante o docente, especificando la fecha y hora del registro.	73	4 de marzo
	6 ALTA 8 ALTA 1 ALTA 0 ALTA	Director, encargada de sistemas y secretaria	Admisión de documentación de cursantes, docentes y almacenamiento	Después de haber introducido la información del acreedor de la documentación debo poder almacenar el file en la base de datos del sistema.	71	4 de marzo
3. Generación de código único	1 ALTA 1 T A	Secretaria	Generación de código único	Después de haber registrado los datos del acreedor y almacenar el file del acreedor de la documentación, se debe generar un código de seguridad asignado al registro de la documentación.	72	18 de marzo
4. Inclusion de código QR	1 M 2 E D 1 I 3 A	Director, coordinador académico	Inclusión de código QR a la documentación presentada	Después de haberse registrado los datos del acreedor y almacenar el file del acreedor de la documentación, la documentación debe derivarse a otro menú para que esta documentación sea revisada, posterior a revisarse y que todo este a cabalidad, se incrustara el código QR parametrizado con el código único, siendo este un aval de garantía de veracidad de los datos y documentación.	62	1 de abril

Entregable	P R I O R D A D	Quien	Necesito	Para	E D S U F R U A E C R I Z O N	Fecha entregable
5. Reportes en base a modulos implementados	1 4 M E D I A 5	Director, coordinador académico	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual “Revisada” o “No revisada”, ademas de desplegar la documentación Revisada en un menú y la No Revisada en otro menú, ademas de poder obtener un reporte de los usuarios registrados, sus roles y privilegios respectivos.	8 3	22 de abril
5. Reportes en base a modulos implementados	1 6 1 7	Encargada de sistemas, secretaria	Generación de reportes en base a módulos implementados	Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual “Revisada” o “No revisada”	8 3	22 de abril

Fuente: Elaboración propia, 2020

### 3.2.7.2. Planificación de las iteraciones del proyecto

**TABLA 9: Planificación de iteraciones del proyecto**

ITERACIÓN		ENTREGABLES		DURACIÓN
Iteración 1	Registro de usuarios con roles y privilegios respectivos	Entregable 1	Registro de usuarios con roles y privilegios respectivos	2 Semanas
Iteración 2	Admisión de documentación de cursantes y docentes	Entregable 2	Admisión de documentación de cursantes y docentes	3 Semanas
Iteración 3	Generación de código único	Entregable 3	Generación de código único	2 Semanas
Iteración 4	Inclusión de código Qr	Entregable 4	Inclusión de código Qr	2 Semanas

ITERACIÓN		ENTREGABLES		DURACIÓN
Iteración 5	Reportes en base a módulos implementados	Entregable 5	Reportes en base a módulos implementados	3 Semanas

*Fuente: Elaboración propia, 2020*

### 3.2.7.2. Relación entre los entregables y objetivos específicos del proyecto

**TABLA 10: Relación de los entregables y objetivos**

OBJETIVOS ESPECIFICOS		ENTREGABLES	
Desarrollar el módulo de registro de usuarios con roles y privilegios respectivos	Entregable 1	Gestion de usuarios y roles de la plataforma	
Desarrollar el módulo de admisión de documentación de cursantes y docentes	Entregable 2	Admisión de documentación de cursantes y docentes	
Desarrollar el módulo de generación de código único para cada documento presentado	Entregable 3	Generación de código único para cada documento	
Desarrollar el módulo de inclusión de código QR a la documentación presentada	Entregable 4	Inclusión de código QR a los documentos presentados	
Desarrollar el módulo de generación de reportes en base a módulos implementados	Entregable 5	Reportes en base a módulos implementados	

*Fuente: Elaboración propia, 2020*

### **3.3. DESARROLLO DEL MÓDULO DE REGISTRO DE USUARIOS CON ROLES Y PRIVILEGIOS RESPECTIVOS.**

#### **3.3.1. Selección de un lenguaje de programación.**

**TABLA 11: Tabla comparativa de lenguajes de programación**

LENGUAJE	VENTAJAS	DESVENTAJAS
PHP	<ul style="list-style-type: none"><li>•Es un lenguaje multiplataforma.</li><li>•Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.</li><li>•El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.</li><li>•Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.</li><li>•Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.</li></ul>	<ul style="list-style-type: none"><li>•Como es un lenguaje que se interpreta en ejecución para ciertos usos, puede resultar un inconveniente que el código fuente no pueda ser ocultado.</li><li>•La ofuscación es una técnica que puede dificultar la lectura del código pero no lo impide y, en ciertos casos, representa un costo en tiempos de ejecución</li></ul>
JAVA	<ul style="list-style-type: none"><li>•Manejo automático de la memoria.</li><li>•Lenguaje Multi-plataforma: El código escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma.</li><li>•Programación Orientada a Objetos: Paradigma muy utilizado hoy en día que facilita y organiza mucho la programación.</li><li>•Puede correr en el explorador y en dispositivos móviles.</li><li>•Sintaxis similar a C/C++, pero más simple.</li><li>•Fácil de aprender.</li></ul>	<ul style="list-style-type: none"><li>•Requiere un intérprete.</li><li>•Algunas implementaciones y librerías pueden tener código rebuscado.</li><li>•Una mala implementación de un programa en java, puede resultar en algo muy lento.</li><li>•Algunas herramientas tienen un costo adicional.</li></ul>

LENGUAJE	VENTAJAS	DESVENTAJAS
Python	<ul style="list-style-type: none"> <li>• Python es un lenguaje de programación poderoso y fácil de aprender.</li> <li>• Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos</li> <li>• La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.</li> <li>• El interprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C)</li> <li>• Python es más fácil de usar, está disponible para sistemas operativos Windows, Mac OS X y Unix.</li> <li>• Python ofrece mucho más chequeo de error que C, y siendo un lenguaje de muy alto nivel, tiene tipos de datos de alto nivel incorporados como arreglos de tamaño flexible y diccionarios.</li> </ul>	<ul style="list-style-type: none"> <li>• Los scripts se lucen para mover archivos de un lado a otro y para modificar datos de texto, no para aplicaciones con interfaz de usuario o juegos</li> <li>• Se puede escribir un programa en C/C++/Java, pero puede tomar mucho tiempo de desarrollo obtener al menos un primer borrador del programa</li> </ul>

Fuente: *Elaboración propia, 2020*

Una vez realizado el análisis de las ventajas y desventajas que presenta cada lenguaje de programación, se ha decidido elegir el lenguaje de programación PHP debido a que es un lenguaje exclusivamente orientado a desarrollo web dinámico, es muy completo y permite desarrollar aplicaciones de alto rendimiento basado en servicios, siendo un lenguaje muy sólido y confiable también se vio la posibilidad de utilizar JAVA, pero se descartó debido a q PHP tiene mejores librerías que permitirán el desarrollo del proyecto, en este sentido se descartaron la otra opción propuesta ya que visual C# es un lenguaje más orientado a realizar aplicaciones de escritorio.

### 3.3.2. Selección de framework adecuado.

**TABLA 12: Tabla comparativa de framework**

FRAMEWORK	VENTAJAS	DESVENTAJAS
Laravel	<ul style="list-style-type: none"> <li>• Framework destinado a PHP.</li> <li>• Reducción de costos y tiempos en el desarrollo y mantenimiento.</li> <li>• Flexible y adaptable no solo al MVC Tradicional (Modelo vista controlador) sino que para reducir código propone usar “Routes with closures”</li> <li>• Es modular y con una amplio sistemas de paquetes y drivers con el que se puede extender la funcionalidad de forma fácil, robusta y segura.</li> <li>• Hace que el manejo de los datos en Laravel no sea complejo.</li> <li>• Es compatible con la gran mayoría de las bases de datos del mercado actual facilitando la migración de los datos de una forma fácil y segura.</li> <li>• Permite la creación de consultas robustas y complejas.</li> <li>• Incluye un sistema de cache que hace que las plantillas sean más rápidas, lo que mejora el rendimiento de la aplicación.</li> </ul>	<ul style="list-style-type: none"> <li>• No existe suficiente documentación bibliográfica de referencia.</li> <li>• Solamente trabaja en entornos UNIX, pero si se puede trabajar en Windows.</li> <li>• Trabaja modo consola para levantar el servidor.</li> <li>• Necesita un excesivo uso de roles y permisos de usuario.</li> </ul>

FRAMEWORK	VENTAJAS	DESVENTAJAS
Django	<ul style="list-style-type: none"> <li>• Es lo mejor de todo y también es lo que se lleva “casi” todo el crédito a la definición de Django. “Django es un framework web de alto nivel” eso está muy claro en la definición, “Django es un framework web de alto nivel escrito en Python”. Gracias a esto Django hereda todas las características y facilidades que nos da Python, entre ellas escribir código bastante fácil de entender, y sobre todo te permite desarrollar aplicaciones muy rápidas y potentes.</li> <li>• Rapidez: Django nació en un ambiente periodístico, donde se subían noticias muy rápido, y como los desarrolladores no pudieron estar a ese ritmo decidieron crear algo que sí lo haga, y así fue como nace Django , es por eso que ha sido estructurado de tal manera que tus aplicaciones web se crean muy rápidas.</li> <li>• DRY: Django utiliza esta filosofía para no crear bloques de código iguales y fomentar la reutilización del mismo.</li> <li>• Admin: Django es el único framework que “por defecto” viene con un sistema de administración activo, listo para ser utilizado sin ningún tipo de configuración.</li> <li>• ORM: Para resumir esto, es una herramienta que te permite realizar consultas SQL a la Base de Datos, sin utilizar SQL.</li> </ul>	<ul style="list-style-type: none"> <li>• Es un poco mas difícil conseguir hosting.</li> <li>• La compatibilidad Su puesta en producción es un tanto mas complicado.</li> <li>• Si se viene programando en php a código limpio, la filosofía MVC complicara las cosas mientras la aprendes bien.</li> <li>• Toda la documentación oficial de <a href="https://code.djangoproject.com">https://code.djangoproject.com</a> es extremadamente buena , pero en ocasiones el ingles no genera una buena respuesta y los traductores no son muy buenos. Pero esta la comunidad que apoya mucho en google y <code>duckduckgo</code> se encuentra muy buenas referencias</li> </ul>

Fuente: Elaboración propia, 2019

Una vez realizado el análisis de las ventajas y desventajas que presenta cada lenguaje de programación, se ha decidido elegir el lenguaje de programación PHP debido a que es un lenguaje exclusivamente orientado a desarrollo web dinámico, es muy completo y permite desarrollar aplicaciones de alto rendimiento basado en servicios, siendo un lenguaje muy sólido y confiable también se vio la posibilidad de utilizar JAVA, pero se descartó debido a q PHP tiene mejores librerías que

permitirán el desarrollo del proyecto, en este sentido se descartaron la otra opción propuesta ya que visual C# es un lenguaje más orientado a realizar aplicaciones de escritorio.

### **3.3.3. Selección de un gestor de base de datos.**

A continuación se presenta las ventajas y desventajas de los Gestores de Bases de Datos seleccionados en el marco teórico para el desarrollo del presente proyecto, de aquí se seleccionara uno que esté acorde al proyecto y brinde mejores beneficios.

**TABLA 13: Tabla comparativa de Gestores de Base de Datos**

GESTOR DE BD	VENTAJAS	DESVENTAJAS
SQL Server	<ul style="list-style-type: none"> <li>• Facilidad de instalación, distribución y utilización.</li> <li>• Almacenamiento de datos.</li> <li>• Seguridad de alto nivel</li> <li>• Integración con internet</li> <li>• Incluye un potente entorno gráfico de administración</li> </ul>	<ul style="list-style-type: none"> <li>• Sus versiones difieren en los requerimientos para su instalación según la versión del sistema operativo Windows.</li> <li>• No tiene soporte en otras plataformas, como ser Linux, Mac OS, Solaris, etc.</li> </ul>
MySQL	<ul style="list-style-type: none"> <li>• La cualidad más destacada es su velocidad y así cómo el software fue diseñado desde un principio, pensando principalmente en la rapidez.</li> <li>• MySQL es gratuito.</li> <li>• Fácil de usar, siguiendo simples reglas en el lenguaje SQL que es el lenguaje estándar para la comunicación con los sistemas de gestión de bases de datos (RDBMS).</li> <li>• MySQL corre en muchos sistemas operativos, Windows, Linux, Mac OS, la mayoría de las variedades de Unix (incluyendo Solaris y AIX).</li> </ul>	<ul style="list-style-type: none"> <li>• Un gran porcentaje de las utilidades de MySQL no están documentadas.</li> <li>• No es intuitivo, como otros programas (ACCESS).</li> </ul>

GESTOR DE BD	VENTAJAS	DESVENTAJAS
PostgreSQL	<ul style="list-style-type: none"> <li>• Fácil de Administrar.</li> <li>• Su sintaxis SQL es estándar y fácil de aprender.</li> <li>• Footprint bajo de memoria, bastante poderoso con una configuración adecuada.</li> <li>• Multiplataforma.</li> <li>• Capacidades de replicación de datos.</li> <li>• Soporte empresarial disponible.</li> </ul>	<ul style="list-style-type: none"> <li>• En comparación con MySQL es más lento en inserciones y actualizaciones.</li> <li>• Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.</li> <li>• Consumo más recursos que MySQL.</li> <li>• La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.</li> </ul>

Fuente: Elaboración propia, 2019

Después de realizar el análisis de las distintas ventajas y desventajas que presentan los diferentes gestores de base de datos contemplados para el análisis y habiendo decidido trabajar con el lenguaje de programación PHP y Laravel también se tomó la decisión de trabajar con el gestor de base de datos MySQL, ya que por un lado PHP destaca su conectividad con MYSQL y por otro lado es uno de los gestores con más facilidad de conexión, configuración e instalación, también cabe resaltar que es un gestor que se desacata principalmente por su velocidad en la realización de operaciones, brindar seguridad en los datos almacenados y soportar bases de datos de gran tamaño.

### 3.3.5. Análisis de requerimientos.

Para el primer incremento se establecerán los requerimientos para el módulo y los actores que intervendrán en el sistema, los cuales se detallan a continuación:

#### 3.3.5.1. Requerimientos funcionales

- El módulo debe permitir registrar la información de nuevos usuarios que tendrán acceso al sistema.

- El módulo debe permitir buscar a un usuario para poder modificar o eliminar sus datos del sistema.
- El módulo debe permitir asignar un cargo a un usuario administrativo.
- El módulo debe permitir crear cuentas de usuario para los usuarios registrados.

### **3.3.5.2. Requerimientos no funcionales**

- **Disponibilidad:** el sistema debe estar disponible para su uso las 24 horas para que los coordinadores puedan tener acceso en cualquier momento.
- **Extensibilidad:** debido a que el sistema está diseñado bajo la Arquitectura Orientada a Servicios podrá integrarse de manera más sencilla con sistemas paralelos de la institución e implementar otras funcionalidades.
- **Mantenibilidad:** se puede agregar nuevas funcionalidades o mejoras al proyecto ya que está siendo desarrollado con metodología ágil Extreme Programing y Arquitectura Orientada a Servicios.
- **Seguridad:** para poder acceder al sistema se necesita estar registrado como usuario, tener una cuenta de usuario y realizar su autenticación correspondiente por cada usuario.
- **Usabilidad:** el sistema está diseñado de manera amigable para el usuario que no necesita conocimientos avanzados para poder utilizar los servicios que éste ofrece.

### **3.3.5.3. Tarjetas de historias de usuarios**

A continuación se detallan las historias de usuario contempladas en las tarjetas de historias de usuario siguientes.

**TABLA 14: Historia de usuario 1**

<b>HISTORIA DE USUARIO</b>	
Numero 1	Nombre: Crear Usuarios
Usuario: Admin	
Modificación de historia Número:	Entregable 1

HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Puntos Estimados: 5
Riesgo en desarrollo	Puntos reales: 3
Descripción: Poder registrar usuarios, para que estos puedan acceder al sistema, ademas de que cuando elimine un usuario no se elimine completamente de la base de datos, si no que este registrado pero ya no tenga acceso a la plataforma, ademas de que cada usuario tenga credenciales de acceso.	
Observaciones: No eliminar de la base de datos a los usuarios, solo darlos de baja.	

*Fuente: Elaboración propia, 2020*

**TABLA 15: Historia de usuario 2**

HISTORIA DE USUARIO	
Numero 2	Nombre: Otorgar Roles y Privilegios
Usuario: Administrador o Director	
Modificación de historia Número:	Entregable 1
Prioridad en negocio: Alta	Puntos Estimados: 5
Riesgo en desarrollo	Puntos reales: 3
Descripción: Los usuarios ya registrados no deben tener acceso todo el sistema, solo deberían poder ver e interactuar con los módulos que a cada empleado le compete.	
Observaciones	

*Fuente: Elaboración propia, 2020*

**TABLA 16: Historia de usuario 3**

HISTORIA DE USUARIO	
Numero 3	Nombre: Crear Usuarios, roles y privilegios
Usuario: Coordinador académico	
Modificación de historia Número:	Entregable 1
Prioridad en negocio: Alta	Puntos Estimados: 5
Riesgo en desarrollo	Puntos reales: 3
Descripción: Tener acceso a las mismas funcionalidades del director, debido a que cuando el se ausenta sus atribuciones, le competen debido a ser el segundo al mando	
Observaciones Crear usuarios, roles y privilegios son acciones que competen a este actor.	

*Fuente: Elaboración propia, 2020*

**TABLA 17: Historia de usuario 4**

HISTORIA DE USUARIO	
Numero 4	Nombre: Crear Usuarios
Usuario: Encargada de sistemas	
Modificación de historia Número:	Entregable 1
Prioridad en negocio: Alta	Puntos Estimados: 3
Riesgo en desarrollo	Puntos reales: 2
Descripción: Poder acceder al sistema mediante credenciales de acceso y poder interactuar con el sistema en las operaciones que competen a este actor.	
Observaciones: El acceso a funcionalidades no esta definido en el entregable 1	

*Fuente: Elaboración propia, 2020*

**TABLA 18: Historia de usuario 5**

HISTORIA DE USUARIO	
Numero 5	Nombre: Crear Usuarios
Usuario: Secretaria	
Modificación de historia Número:	Entregable 1
Prioridad en negocio: Alta	Puntos Estimados: 5
Riesgo en desarrollo	Puntos reales: 3
Descripción: Poder acceder al sistema mediante credenciales de acceso y poder interactuar con el sistema en las operaciones que competen a este actor.	
Observaciones: El acceso a funcionalidades no esta definido en el entregable 1	

*Fuente: Elaboración propia, 2020*

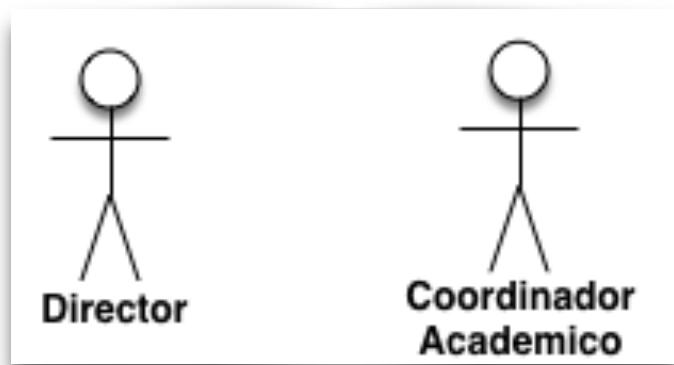
### **3.3.6. Diagramas UML para el módulo de registro de usuarios.**

#### **3.3.6.1. Diagramas de caso de uso para el módulo de registro de usuarios.**

- **Actores**

Para desarrollar el módulo de gestión de usuarios se identificaron 2 actores los cuales tendrán la opción de administrar los usuarios.

**FIGURA 21: Actores del módulo de gestión de usuarios**



*Fuente: Elaboración propia, 2020*

- **Descripción de actores**

- **Director de posgrado:** Este usuario es el encargado de la administración de toda la institución, el cual tiene acceso a todo el sistema podrá realizar la administración de usuarios y todos los procesos, teniendo la posibilidad de generar cualquier tipo de reportes acorde a sus necesidades.
- **Coordinador académico:** Este usuario es una persona del área de informática y tiene amplio conocimiento sobre el manejo de sistemas, tiene acceso a todos los procesos que realiza el sistema, se encarga de administrar las cuentas de usuario del sistema y asignarlas a cada uno de ellos, así mismo puede generar cualquier tipo de reportes acorde a sus necesidades.

#### **3.3.6.2. Diagrama de caso de uso general**

**FIGURA 22: Diagrama de casos de uso general de gestión de usuarios**



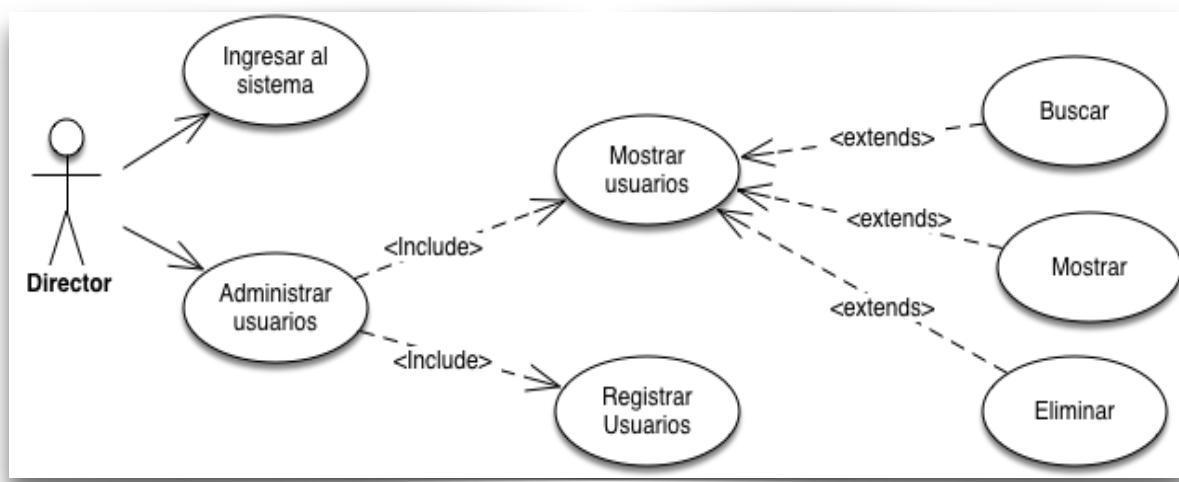
*Fuente: Elaboración Propia, 2020*

### 3.3.6.3. Diagramas de caso de uso por actor

Los casos de uso por actor permitirán visualizar de manera más detallada las diferentes actividades, funcionalidades y tareas que tiene cada actor en el sistema, en este sentido se realizaran los diferentes casos de uso por actor, los cuales intervendrán en el primer incremento.

**Actor:** Coordinador académico

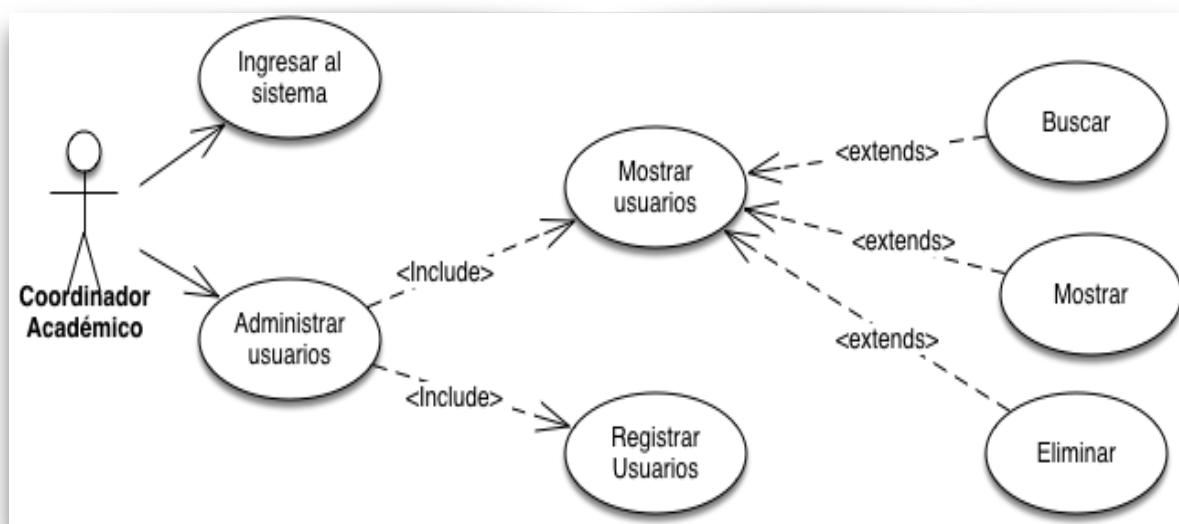
FIGURA 23: Diagrama de caso de uso de director



Fuente: Elaboración Propia, 2020

**Actor:** Director de Posgrado

FIGURA 24: Diagrama de caso de uso del Director de Posgrado



Fuente: Elaboración Propia, 2020

### 3.3.6.4. Diagrama de clases para el módulo de gestión de usuarios.

FIGURA 25: Diagrama de clases gestión de usuarios

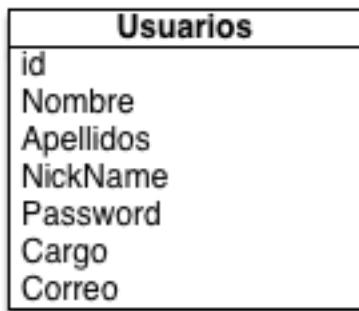


Fuente: Elaboración Propia, 2020

### 3.3.7. Diseño de la base de datos

#### 3.3.7.1. Diagrama de base de datos

FIGURA 26: Diagrama de base de datos gestión de usuarios



Fuente: Elaboración Propia, 2020

### 3.3.7. Implementar módulo de registro de usuarios

A continuación se muestran las partes mas importantes de código de la iteración.

#### 3.3.7.1. Código de modelo de usuarios

```
class Usuario(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField('Nombres',max_length=50, blank= True)
    apellidos = models.CharField('Apellidos',max_length=50, blank=True)
```

```

        sexo = models.CharField('Sexo',max_length=30,
                               choices=((('MASCULINO','Masculino'),('FEMENINO','Femenino'),
                                         ('PREFIERO NO DECIRLO','Prefiero no decirlo')),default='MASCULINO')
                               correo = models.EmailField('Correo',max_length=30, blank=True)
                               estado = models.BooleanField('Estado', default=True)
                               imagen = models.ImageField(upload_to='fotos',null=True,blank=True)
                               comentarios = models.TextField('Comentario',blank=True,null=True)

```

Para la gestione usuarios se denotan las lineas de medis.py debido a que estos modelos serán los campos almacenados en la base de datos.

### 3.3.7.2. Código crear usuarios

```

def crearUsuario(request):
    if request.method == 'POST':
        usuario_form = usuarioForms(request.POST)
        if usuario_form.is_valid():
            usuario_form.save()
            return redirect('usuario:listar_usuarios')
    else:
        docente_form = DocenteForms()
    return render(request,'usuario/crear_usuarios.html',
{'usuario_form':docente_form})

```

Para crear usuarios se deben implementar en las vistas para que este interactúe con los modelos, en los cuales están implementados la petición POST, para posteriormente renderizarlo en los templete, según el modelo vista templete que implementa django.

### 3.3.8. Pruebas de aceptación

**TABLA 19: Prueba 1**

Caso de prueba	
Numero de caso de prueba 1	Numero de historia de usuario: 1,3
Nombre de caso de prueba: Prueba a la creación de usuarios	
Descripción: Registro de usuarios con nick name y contraseña, ademas de registrar sus nombres	
Entradas: Ingreso de datos al formulario de registro de usuarios, tales como nickname, password, registro de nombre del usuario, apellidos, correo, etc.	
Resultado esperado: Registro satisfactorio de los usuarios	

### Caso de prueba

Evaluación: Registro satisfactorio, con contraseña cifrada en base de datos

*Fuente: Elaboración propia, 2020*

**TABLA 20: Prueba 2**

Caso de prueba	
Numero de caso de prueba 2	Numero de historia de usuario: 2,4
Nombre de caso de prueba: Prueba a la otorgaron de roles y privilegios	
Descripción: Asignación de accesibilidad de los usuarios registrados a las ciertas funcionalidades del sistema que correspondan según el rol que desempeñan dentro la organización	
Entradas: Asignación del usuario a roles registrados: Director, coordinador académico, encargada de sistemas y secretaria	
Resultado esperado: Asignación satisfactoria de los usuarios a roles respectivos	
Evaluación: Asignación de roles satisfactoria	

*Fuente: Elaboración propia, 2020*

**TABLA 21: Pruebas de aceptación**

ID	Nombre de la prueba	Historia de usuario	Criterio de aceptación	Resultado obtenido
1	Prueba a la creación de usuarios	Crear Usuarios	Registro de usuarios con nick name y contraseña, ademas de registrar sus nombres	Satisfactorio
2	Prueba a la otorgaron de roles y privilegios	Otorgar roles y privilegios	Asignación de accesibilidad de los usuarios registrados a las ciertas funcionalidades del sistema que correspondan según el rol que desempeñan dentro la organización	Satisfactorio

*Fuente: Elaboración Propia, 2020*

## 3.4. DESARROLLO DEL MÓDULO DE ADMISIÓN DE DOCUMENTACIÓN DOCENTE, DE CURSANTES Y ALMACENAMIENTO.

### 3.4.1. Analizar requerimientos

Para el segundo incremento se establecen requerimientos que se detallan a continuación:

### **3.4.1.1. Requerimientos funcionales del primer incremento**

- El modulo debe permitir ingresar datos de la documentación que se almacenara en el repositorio de documentos
- El modulo debe poder almacenar cualquier documento digital contemplando imágenes en cualquiera de sus extensiones (Jpg, png, jpeg, etc.)
- Al modulo debe poder acceder director de posgrado, encargada de sistemas y secretaria para el registro de documentación

### **3.4.1.2. Tarjetas de historias de usuario del módulo admisión de documentación**

**TABLA 22: Historia de usuario 6**

<b>HISTORIA DE USUARIO</b>	
Numero 6	Nombre: Admisión de documentación de cursantes y docentes
Usuario: Secretaria	
Modificación de historia Número:	Entregable 2
Prioridad en negocio: Alta	Puntos Estimados: 7
Riesgo en desarrollo	Puntos reales: 5
Descripción: Ya dentro del sistema se debe poder desplegar un menú donde pueda registrar la información del acreedor de la información ya sea del cursante o docente, especificando la fecha y hora del registro.	
Observaciones:	

*Fuente: Elaboración propia, 2020*

**TABLA 23: Historia de usuario 7**

<b>HISTORIA DE USUARIO</b>	
Numero 7	Nombre: Admisión de documentación de cursantes y docentes y almacenamiento
Usuario: Secretaria	
Modificación de historia Número:	Entregable 2
Prioridad en negocio: Alta	Puntos Estimados: 7

HISTORIA DE USUARIO	
Riesgo en desarrollo	Puntos reales:5
Descripción: Después de haber introducido la información del acreedor de la documentación debo poder almacenar el file en la base de datos del sistema.	
Observaciones: La documentación digital puede no introducirse	

*Fuente: Elaboración propia, 2020*

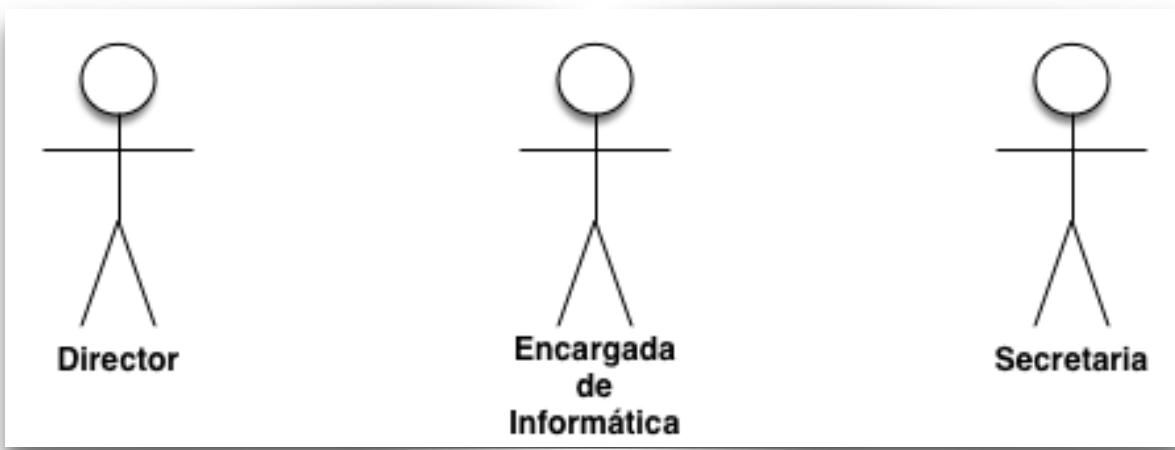
### 3.4.2. Diagramas UML para el módulo de admisión de documentación

#### 3.4.2.1. Diagramas de casos de uso para el módulo de admisión de documentación

##### a) Actores

Para desarrollar el módulo se identificaron los siguientes actores de incidencia directa con el mismo

**FIGURA 27: Actores del módulo de admisión de documentación**



*Fuente: Elaboración Propia, 2020*

##### b) Descripción de actores

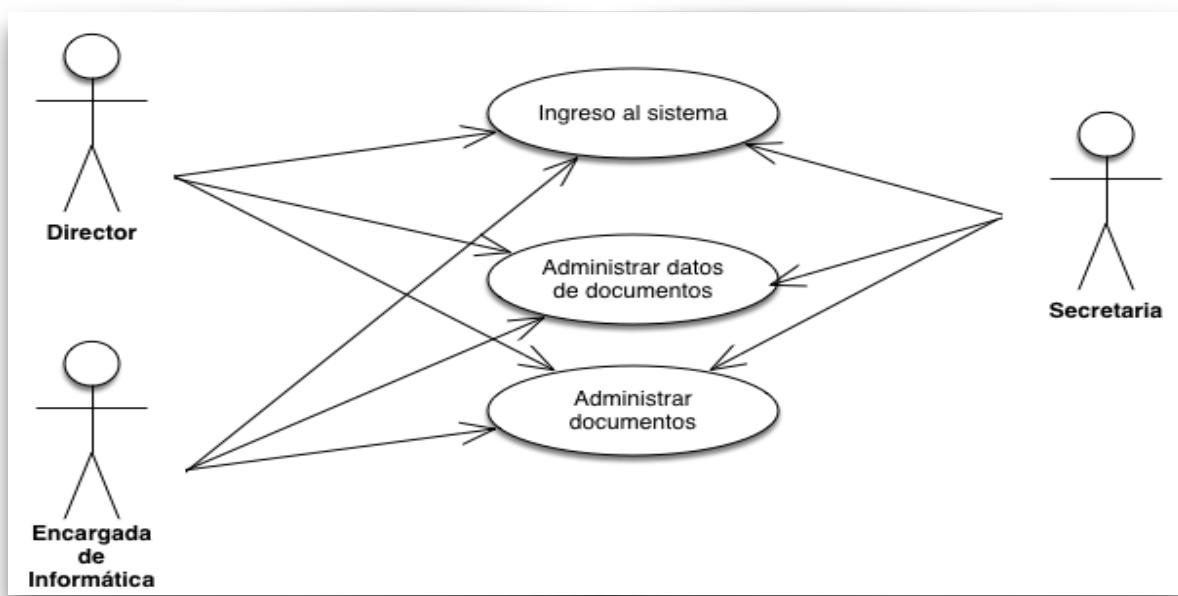
- **Director**: El director al ser un super usuario puede acceder a esta funcionalidad del recepción de documentación de docentes o cursantes ademas de registrar los datos que este modulo requiere.

- **Encargada de informática:** La encargada de informática también al ser super usuario tiene la posibilidad de registrar datos de la recepción de documentos y almacenarlos en el sistema
- **Secretaria:** La secretaria entre una de sus funciones actuales es la de recepciones la documentación de cursantes o docentes según sea el caso, ademas de registrar los datos de los documentos que se almacenaran

### 3.4.2.2. Diagramas de casos de uso del módulo de recepción de documentos

### 3.4.2.3. Diagrama de casos de uso general

FIGURA 28: Actores del módulo de admisión de documentación



Fuente: Elaboración Propia, 2020

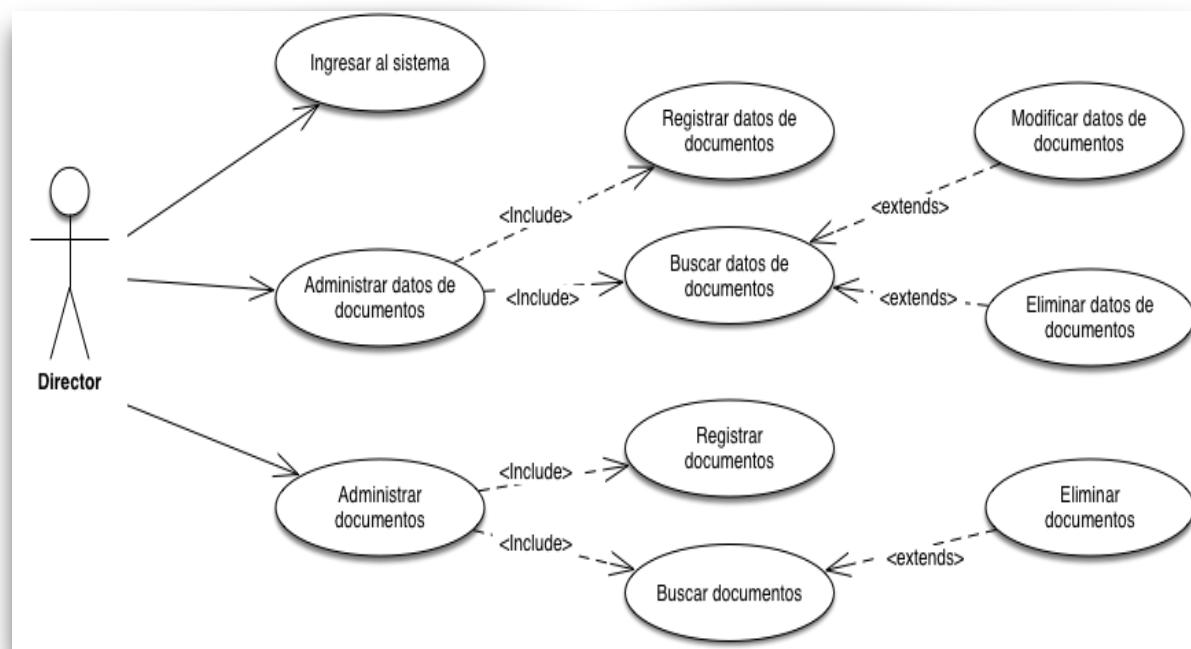
El director, encargada de informática y la secretaria, debe ingresar las funcionalidades del sistema mediante el ingreso al sistema con la respectiva validación de sus credenciales de ingreso y permisos respectivos, los mismos interactuarán con la administración de datos de los documentos y la administración de documentación.

### 3.4.2.4. Diagrama de casos de uso por actor

A continuación se describen casos de uso por actor los cual coadyuvara a visualizar de manera mas minuciosa las diferentes actividades de los actores involucrados en el módulo.

#### a) Director

FIGURA 29: Actores del módulo de admisión de documentación

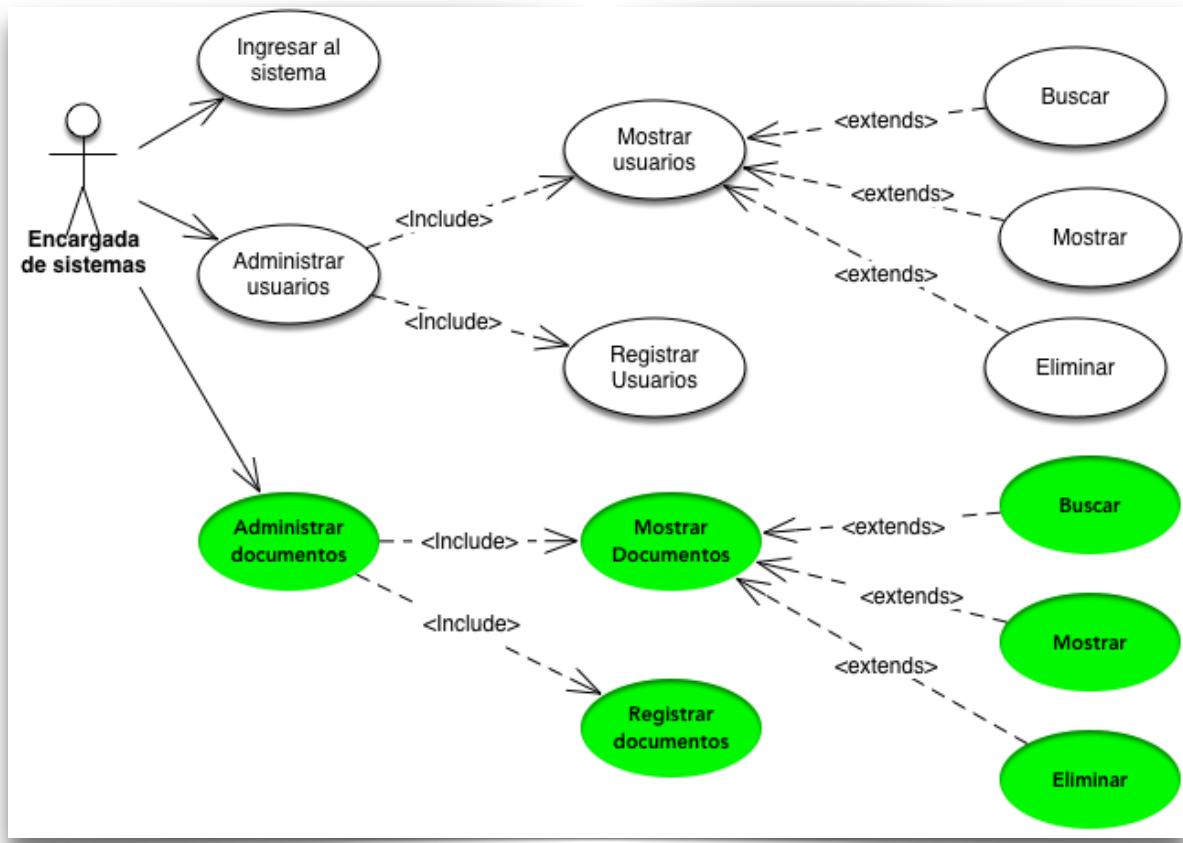


Fuente: Elaboración Propia, 2020

El director tendrá acceso a la administración de datos de los documentos pudiendo realizar operación cruz, tambien podrá administrar la documentación almacenada en la plataforma, registrando documentos, búsqueda de documentos y eliminar documentos.

## b) Encargada de informática

FIGURA 30: Actor encargada de informática

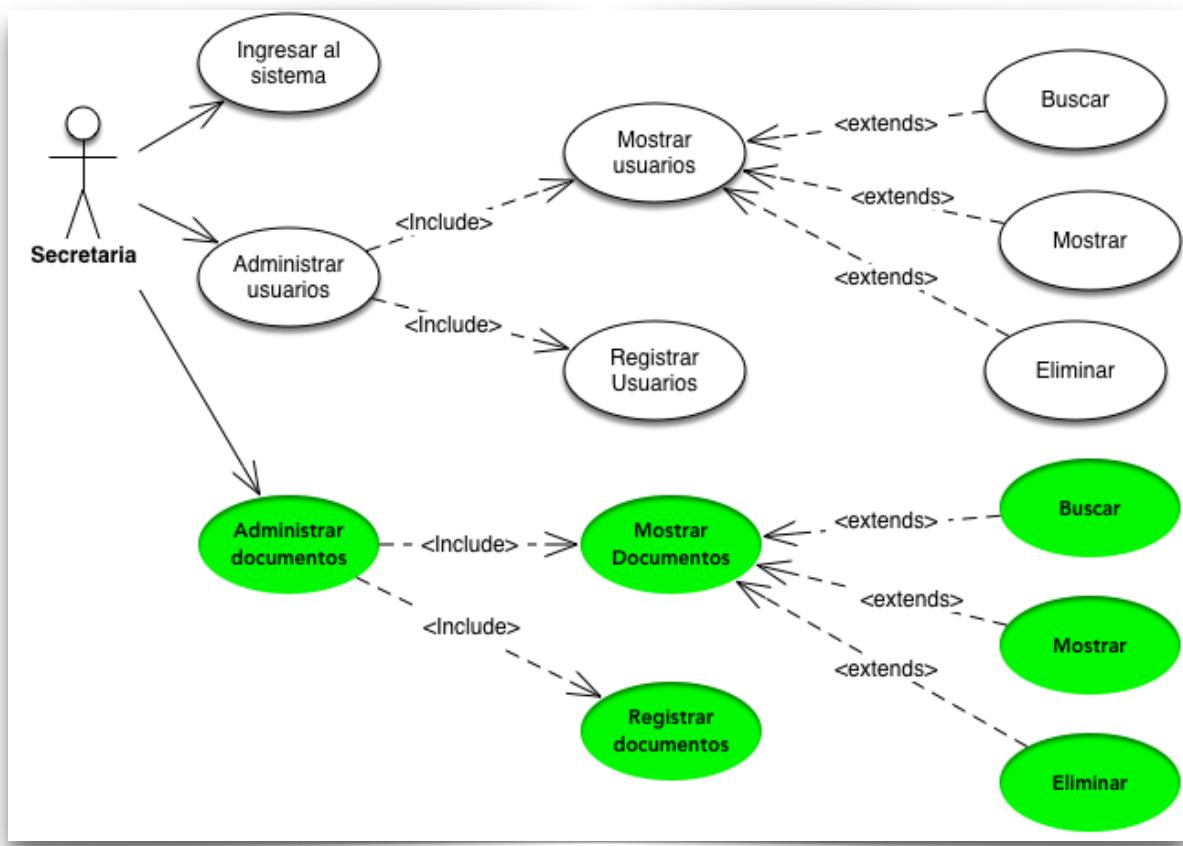


Fuente: Elaboración Propia, 2020

La encargada de informática tendrá acceso a la administración de datos de la documentación pudiendo realizar operaciones CRUD, así mismo podrá acceder a la gestión de documentos, pudiendo registrar documentos, buscar documentos y eliminarlos.

### c) Secretaria

FIGURA 31: Actor secretaria

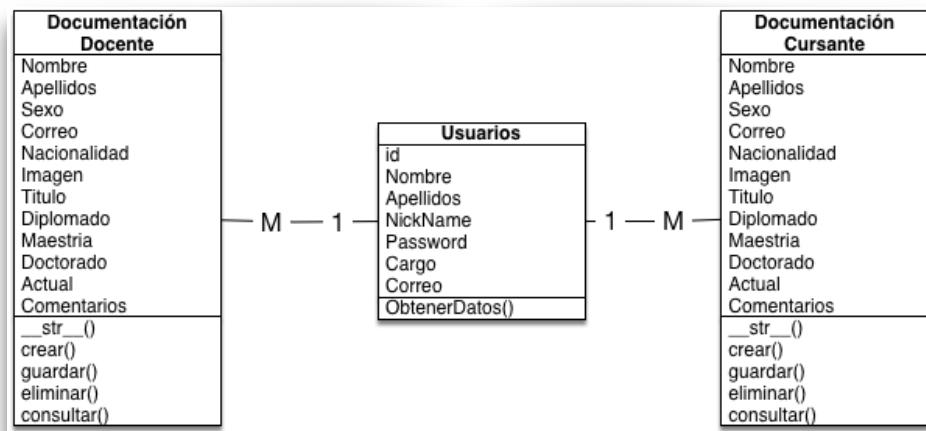


Fuente: Elaboración Propia, 2020

La secretaria tendrá acceso a la administración de datos de documentos, pudiendo registrar, buscar y modificar datos, así mismo se le otorgara acceso a la administración de documentos pudiendo atribuirse el registro y búsqueda como permisos a este actor.

### 3.4.2.5. Diagramas de clases módulo recepción de documentación

FIGURA 32: Diagrama de clases



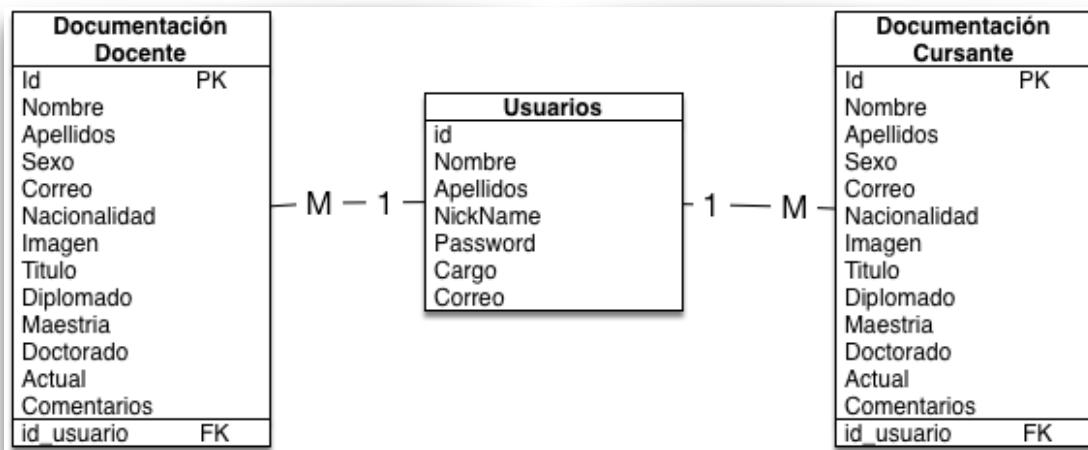
Fuente: Elaboración Propia, 2020

### 3.4.3. Actualizar base de datos

Para la representación de la actualización de la base de datos se representa con un diagrama de base de datos el cual se detalla a continuación.

#### 3.4.3.1. Diagramas de bases de datos módulo recepción de documentación

FIGURA 33: Actor del módulo de recepción de documentación



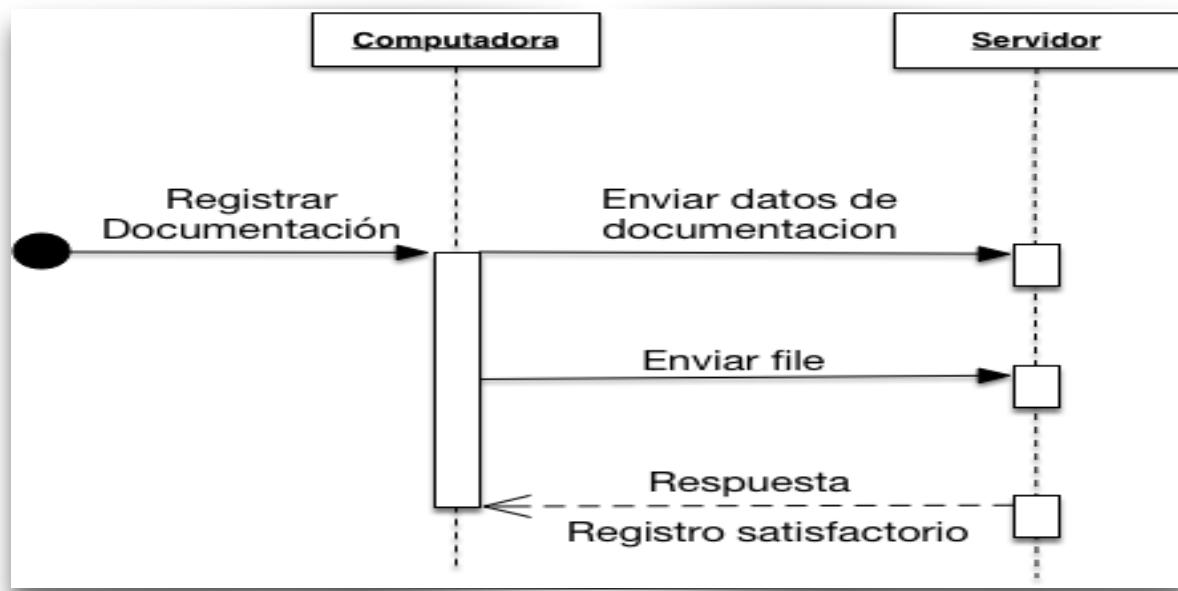
Fuente: Elaboración Propia, 2020

### 3.4.4. Diseño de servicios de la api REST

Para describir el diseño de la api Restful se utiliza el diagrama de secuencia debido a que es el mas adecuado para documentar la interacción de la misma.

#### 3.4.4.1. Diagrama de secuencia para el diseño de la api REST del módulo de admisión de documentación.

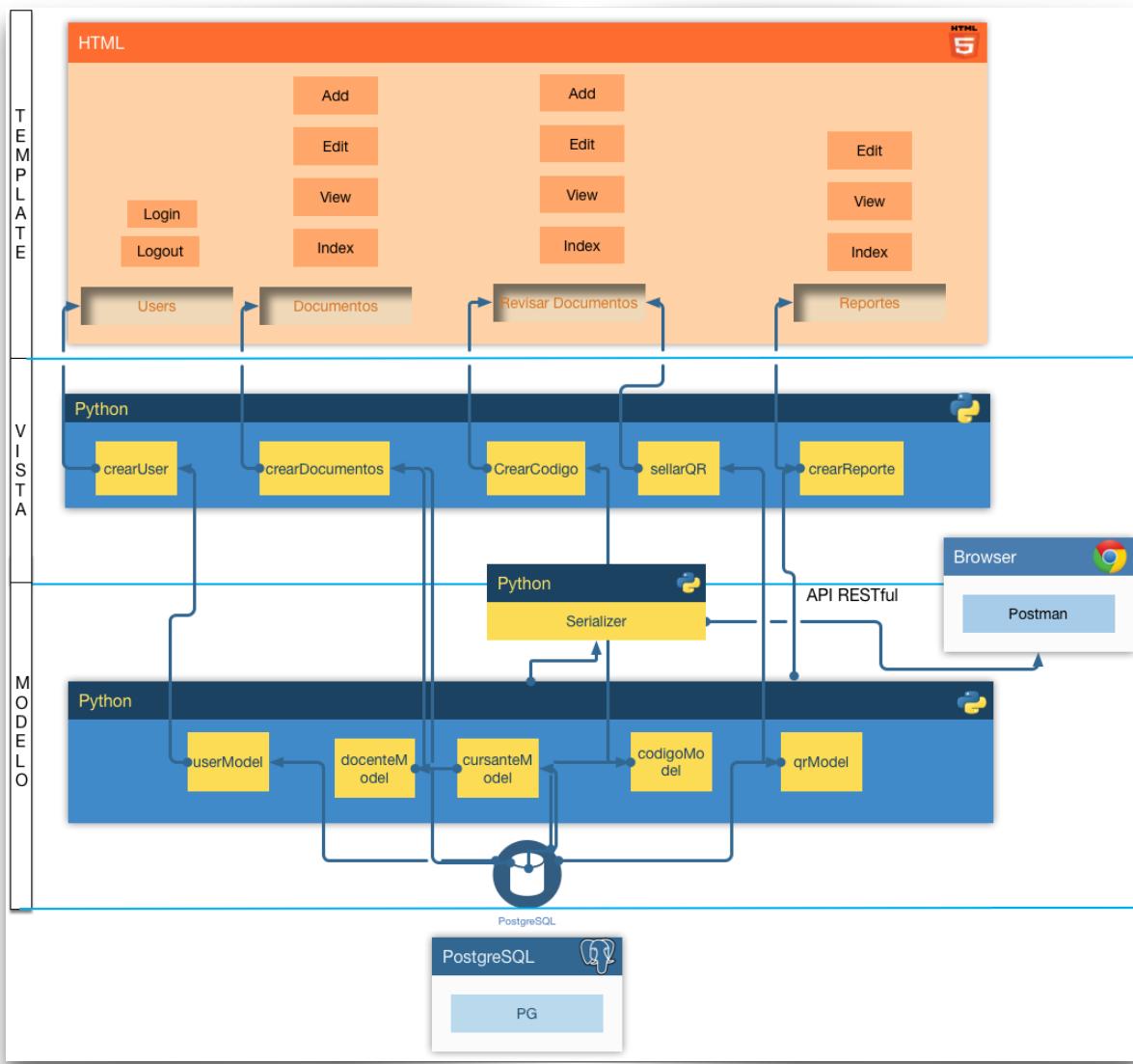
FIGURA 34: Diagrama de estados api REST



Fuente: Elaboración Propia, 2020

### 3.4.4.2. Diagrama de por capas con la arquitectura MVT

FIGURA 35: Diagrama de capas arquitectura MVT



Fuente: Elaboración Propia, 2020

### 3.4.5. Implementar el módulo para la admisión de documentación

A continuación se muestran las partes más importantes de código de la iteración.

### 3.4.5.1. Modelo Docentes

```
class docente(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField('Nombres',max_length=50, blank= True)
    apellidos = models.CharField('Apellidos',max_length=50, blank=True)
    sexo = models.CharField('Sexo',max_length=30,
                           choices=((('MASCULINO','Masculino'),
                                     ('FEMENINO','Femenino'), ('PREFIERO NO DECIRLO','Prefiero no
decirlo')),default='MASCULINO')
    correo = models.EmailField('Correo',max_length=30, blank=True)
    nacionalidad = models.CharField('Nacionalidad',max_length=30,
                                    blank=True)
    estado = models.BooleanField('Estado', default=True)
    imagen = models.ImageField(upload_to='fotos',null=True,blank=True)
    titulo = models.FileField('Titulo en provision',
nacional',upload_to='docs',blank=True)
    diplomado = models.FileField('Titulo de
diplomado',upload_to='docs',blank=True)
    maestria = models.FileField('Titulo de
maestria',upload_to='docs',blank=True)
    doctorado = models.FileField('Titulo de
doctorado',upload_to='docs',blank=True)
    actual = [
models.DateTimeField('Actual',default=datetime.now,blank=False)
    comentarios = models.TextField('Comentario',blank=True,null=True)
    codigoS = UUIDField(default=uuid.uuid4,editable=False)
```

### 3.4.5.2. Modelo Cursantes

```
class Cursante(models.Model):
    id = models.AutoField(primary_key=True)
    nombre = models.CharField('Nombres',max_length=50, blank= True)
    apellidos = models.CharField('Apellidos',max_length=50, blank=True)
    sexo = models.CharField('Sexo',max_length=30,
                           choices=((('MASCULINO','Masculino'),
                                     ('FEMENINO','Femenino'), ('PREFIERO NO DECIRLO','Prefiero no
decirlo')),default='MASCULINO')
    correo = models.EmailField('Correo',max_length=30, blank=True)
    nacionalidad = models.CharField('Nacionalidad',max_length=30,
                                    blank=True)
```

```

estado = models.BooleanField('Estado', default=True)
imagen = models.ImageField(upload_to='fotos', null=True, blank=True)
titulo = models.FileField('Titulo en provision')
nacional', upload_to='docs', blank=True)
diplomado = models.FileField('Titulo de')
diplomado', upload_to='docs', blank=True)
maestria = models.FileField('Titulo de')
maestria', upload_to='docs', blank=True)
doctorado = models.FileField('Titulo de')
doctorado', upload_to='docs', blank=True)
actual =
models.DateTimeField('Actual', default=datetime.now, blank=False)
comentarios = models.TextField('Comentario', blank=True, null=True)
codigoS = UUIDField(default=uuid.uuid4, editable=False)

```

### 3.4.5.3. Servicio de registro de la api REST

Para realizar la prueba de la api REST se utilizaron los servicios de cliente de Postman, para poder probar que la aplicación puede interactuar con otras aplicaciones se realizo el registro de la documentación y el file correspondiente mediante JSON, que se describe a continuación.

**FIGURA 36: Servicio de registro de api REST**

Key	Value	Description
file	Escanear.jpeg	
nombre	Panchito	
apellidos	Ibrahimovic	
sexo	MASCULINO	
correo	os@o.com	
nacionalidad	Peruano	

```

1  {
2      "id": 8,
3      "nombre": "Panchito",
4      "apellidos": "Ibrahimovic",
5      "sexo": "MASCULINO",
6      "correo": "os@o.com",
7      "nacionalidad": "Peruano",
8      "estado": true,
9      "codigoSecreto": "989d5258-d134-4dc1-8591-49a74b693475",
10     "file": "/fotos/Escanear.jpeg"
11 }

```

Fuente: Elaboración Propia, 2020

Realizado este proceso se pudo constatar que el documento ademas de los datos del acreedor de la documentación se almacenaron correctamente en la base de datos.

### 3.4.6. Pruebas al módulo

**TABLA 24: Prueba 3**

Caso de prueba	
Numero de caso de prueba 3	Numero de historia de usuario: 5,7,9
Nombre de caso de prueba: Prueba de admisión de documentación de cursantes y docentes	
Descripción: Registro de datos del acreedor de documentación tales como nombre, apellidos, correo, sexo, fuerza, etc.	
Entradas: Ingreso de datos al formulario de registro de documentación nombre, apellidos, correo, sexo, fuerza, etc.	
Resultado esperado: Registro satisfactorio de la documentación.	
Evaluación: Registro satisfactorio de documentación en la base de datos	

*Fuente: Elaboración propia, 2020*

**TABLA 25: Prueba 4**

Caso de prueba	
Numero de caso de prueba 4	Numero de historia de usuario: 6,8,10
Nombre de caso de prueba: Prueba de admisión de documentación de cursantes, docentes y almacenamiento	
Descripción: Carga de la documentación digital de docentes o cursantes y almacenaje de la documentación en la base de datos.	
Entradas: Carga de documentación digital correspondiente	
Resultado esperado: Almacenaje satisfactorio de documentación digital a la base de datos	
Evaluación: Almacenaje satisfactorio	

*Fuente: Elaboración propia, 2020*

**TABLA 26: Pruebas de aceptación**

ID	NOMBRE DE LA PRUEBA	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	RESULTADO OBTENIDO
3	Prueba de admisión de documentación de cursantes y docentes	Admisión de documentación de cursantes y docentes	Registro de datos del acreedor de documentación tales como nombre, apellidos, correo, sexo, fuerza, etc.	Satisfactorio

ID	NOMBRE DE LA PRUEBA	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	RESULTADO OBTENIDO
5	Prueba de admisión de documentación de cursantes y docentes y almacenamiento	Admisión de documentación de cursantes, docentes y almacenamiento	Carga de la documentación digital de docentes o cursantes y almacenaje de la documentación en la base de datos.	Satisfactorio

Fuente: *Elaboración Propia, 2020*

### 3.5. DESARROLLO DEL MÓDULO DE GENERACIÓN DE CÓDIGO ÚNICO PARA CADA DOCUMENTO PRESENTADO.

#### 3.5.1. Análisis de requerimientos

##### 3.5.1.1. Requerimientos funcionales

- El módulo debe permitir implementar algoritmo de técnicas criptográficas simétricas
- El módulo debe generar códigos únicos para cada documento presentado
- El módulo debe poder agregar fecha y hora de la subida de cada documento a la plataforma, al código único.

##### 3.5.1.2. Requerimientos no funcionales

- **Disponibilidad:** El sistema debe estar disponible las 24 horas para que los usuarios puedan tener acceso en cualquier momento.
- **Extensibilidad:** En base a las características de la arquitectura orientada a servicios, la plataforma podrá integrarse a otras aplicaciones dentro la organización.
- **Mantenibilidad:** Se puede agregar funcionalidades a la plataforma debido a que esta siendo desarrollado bajo la metodología de programación extrema, ademas de que esta siendo desarrollado bajo la arquitectura orientada a servicios.

- **Seguridad:** El sistema está siendo desarrollado para ser utilizado de una manera simple y fácil de aprender para el usuario final debido a la implementación de workflow en el proyecto.

### **3.5.1.3. Tarjetas de historias de usuario del modulo de generación de código único.**

**TABLA 27: Historia de usuario 8**

<b>HISTORIA DE USUARIO</b>	
Numero 8	Nombre: Generación de código único
Usuario: Secretaria	
Modificación de historia Número:	Entregable 3
Prioridad en negocio: Alta	Puntos Estimados: 7
Riesgo en desarrollo	Puntos reales: 5
Descripción: Después de haber registrado los datos del acreedor y almacenar el file del acreedor de la documentación, se debe generar un código de seguridad asignado al registro de la documentación.	
Observaciones: La inclusión del código único es obligatorio	

*Fuente: Elaboración propia, 2020*

### **3.5.2. Selección de algoritmos de generación código único**

**TABLA 28: Selección de algoritmo de generación de código único**

<b>NOMBRE</b>	<b>Ventajas</b>	<b>Desventajas</b>
<b>Random Code</b>	<ul style="list-style-type: none"> <li>- Ampliamente utilizado en diversas aplicaciones, como métodos de monte carlo, muestreo, análisis numérico, programación.</li> <li>- Al ser simétrico su implementación no es compleja.</li> <li>- No requieren grandes recursos de hardware para ser implementada.</li> <li>- Ampliamente difundido.</li> </ul>	<ul style="list-style-type: none"> <li>- Vulnerabilidad latente</li> <li>- Al no ser asimétrico, se pueden descifrar con poco esfuerzo</li> <li>- Recomendable para entornos locales ejemplo: Intranets.</li> </ul>

NOMBRE	Ventajas	Desventajas
DES	<ul style="list-style-type: none"> <li>- Es uno de los sistemas mas empleados y extendidos, por tanto es de los mas probados.</li> <li>- Implementación sencilla.</li> </ul>	<ul style="list-style-type: none"> <li>- No es seguro</li> <li>- No se permite una clave de longitud variable, es decir no se puede aumentar el tamaño del código para aumentar la seguridad</li> <li>- Es vulnerable a criptoanalisis.</li> <li>- La longitud de clave de 56 bits es demasiado corta, y por tanto vulnerable. Actualmente DES ya no es un estándar, debido a que en 1999 fue roto. (Johannes, 2004)</li> </ul>
UUID	<ul style="list-style-type: none"> <li>- Es uno de los sistemas de identificadores utilizados en la industria de la banca</li> <li>- Un UUID puede ser utilizado como un identificador específico y ser utilizado en varias ocasiones para identificar el mismo objeto en diferentes contextos</li> <li>- UUID esta estandarizado por la Open Software Foundation (SOF)</li> <li>- Se documenta como parte de la norma ISO / IEC 11578</li> <li>- La probabilidad de que se genere un código similar es muy cercano a cero</li> </ul>	<ul style="list-style-type: none"> <li>- Mayor uso del espacio en la base de datos</li> <li>- No puede realizarse el order by ID para conseguir la inserción de la orden</li> <li>- No recomendable su aplicación en aplicaciones de gran tamaño</li> </ul>

Fuente: Elaboración Propia, 2020

Debido a un análisis previo de las ventajas y desventajas de los algoritmos se opta por utilizar UUID Universally Unique Identifier, debido a las ventajas que ofrece, tales como su uso en diferentes aplicaciones tales como es la industria de la banca, ademas de que esta normado bajo ISO / IEC ademas de que la implementación en el framework django esta bastante extendido. Se descarta DES debido a que no es seguro, no se permite ampliar la longitud de variables para volverlo mas seguro, vulnerable a criptoanalisis, ya no es un estándar y Random Code por la vulnerabilidad latente que este presenta.

### 3.5.3. Diagramas UML para el módulo de generación de código único

#### 3.5.3.1. Diagramas de casos de uso para el módulo de generación de código único

#### 3.5.3.2. Actores

Para desarrollar el módulo se identificaron los siguientes actores de incidencia directa con el mismo

FIGURA 37: Actores del módulo



Fuente: *Elaboración Propia, 2020*

#### 3.5.3.3. Descripción de actores

- **Secretaria:** Este usuario al cargar documentación a la plataforma, esta generara automáticamente un código único para cada documento presentado, combinando la fecha y hora del momento en que se esta almacenando la documentación.
- **Encargada de sistemas** Este usuario al cargar documentación a la plataforma, esta generara automáticamente un código único para cada documento presentado, combinando la fecha y hora del momento en que se esta almacenando la documentación.

### 3.5.3.4. Diagramas de caso de uso del módulo

FIGURA 38: Diagrama de caso de uso del módulo

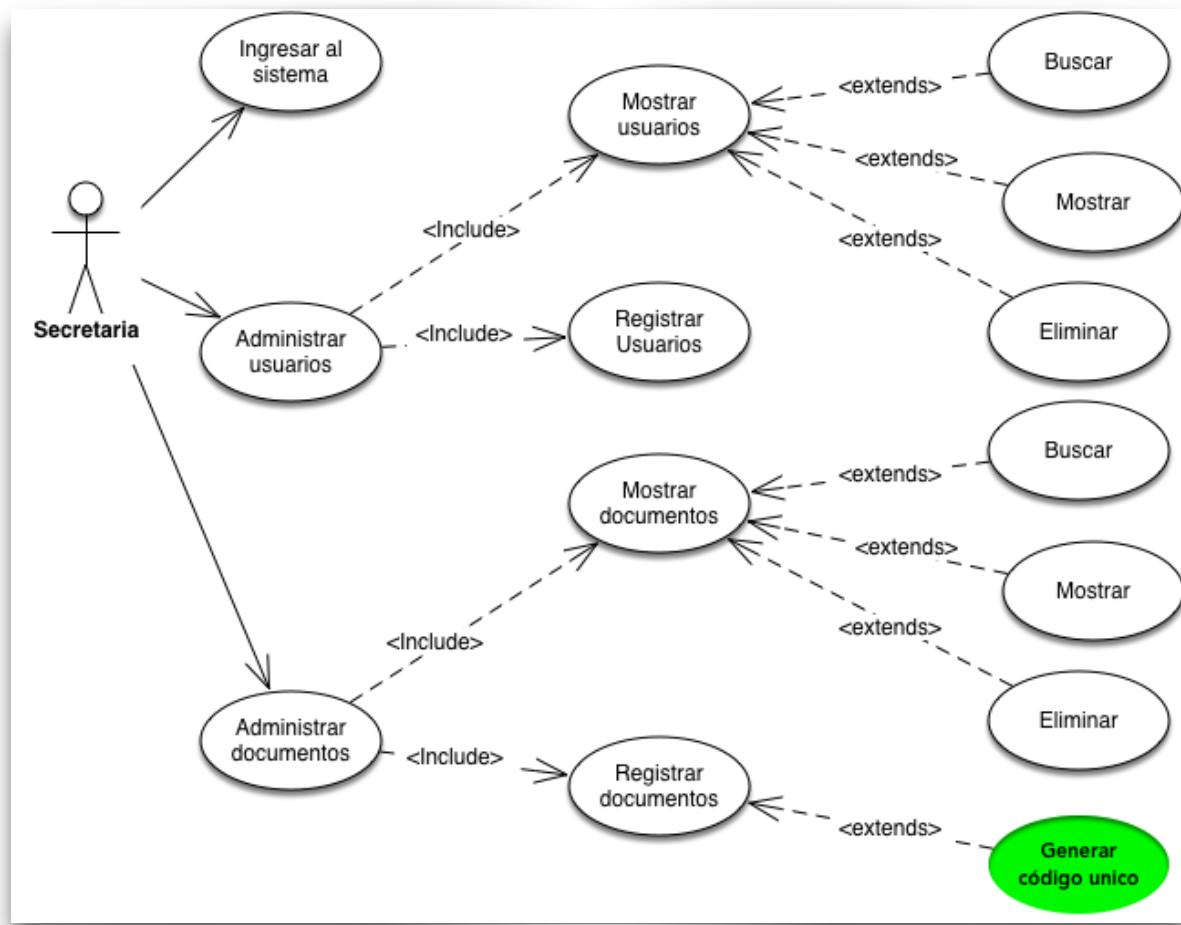


Fuente: Elaboración Propia, 2020

Para desarrollar La secretaria tendrá acceso a la administración de documentos, registrando los datos y subiendo a la plataforma la documentación digital de docentes o cursantes, la plataforma generará automáticamente un código único para cada documento almacenado, esto se lo realizara en base a la implementación del algoritmo Random Code, almacenando los datos e información de los documentos con un código único otorgado.

### 3.5.3.5. Diagramas de caso de uso de secretaria

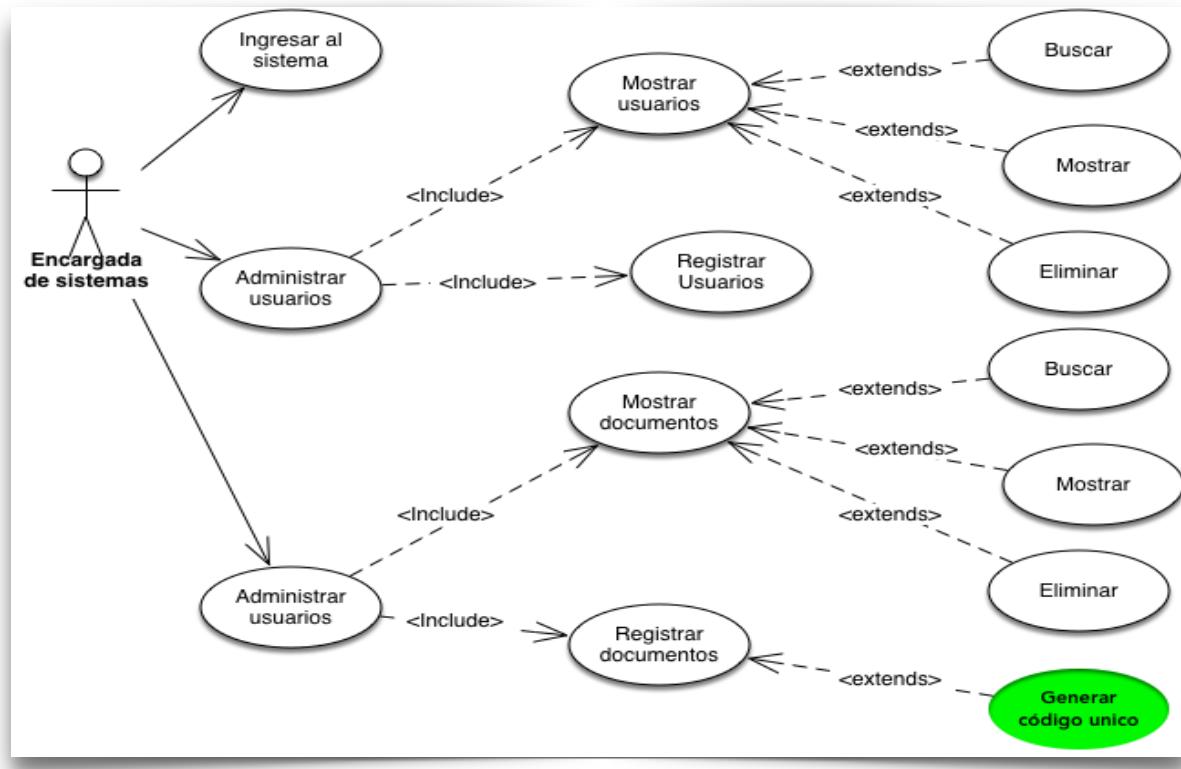
FIGURA 39: Actor secretaria



Fuente: Elaboración Propia, 2020

### 3.5.3.6. Diagramas de caso de uso de encargada de sistemas

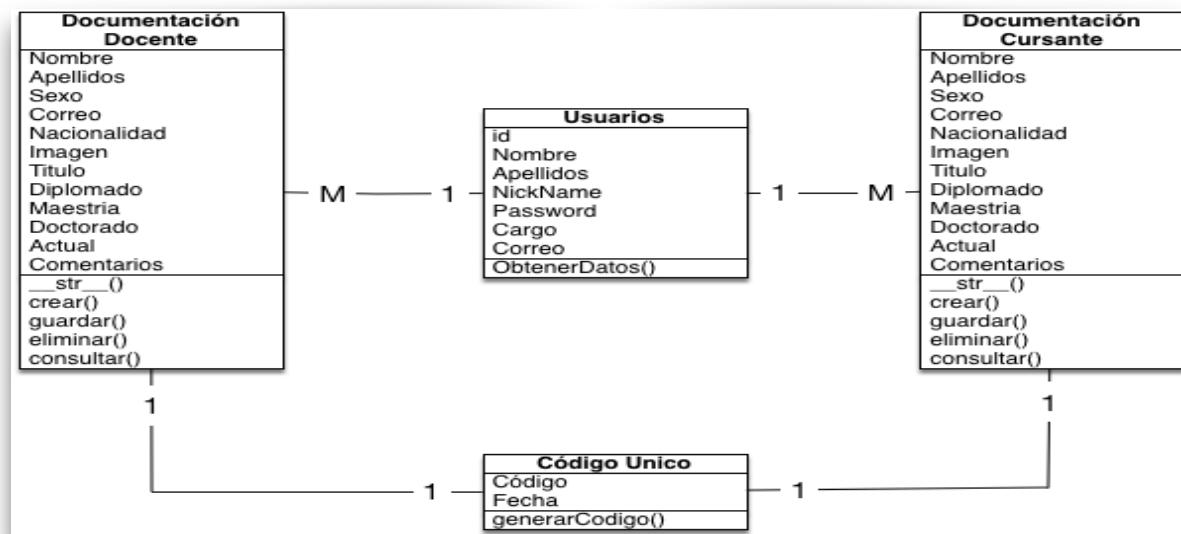
FIGURA 40: Actor encargada de sistemas



Fuente: Elaboración Propia, 2020

### 3.5.3.7. Diagrama de clases del módulo de generación de código único

FIGURA 41: Diagrama de clases

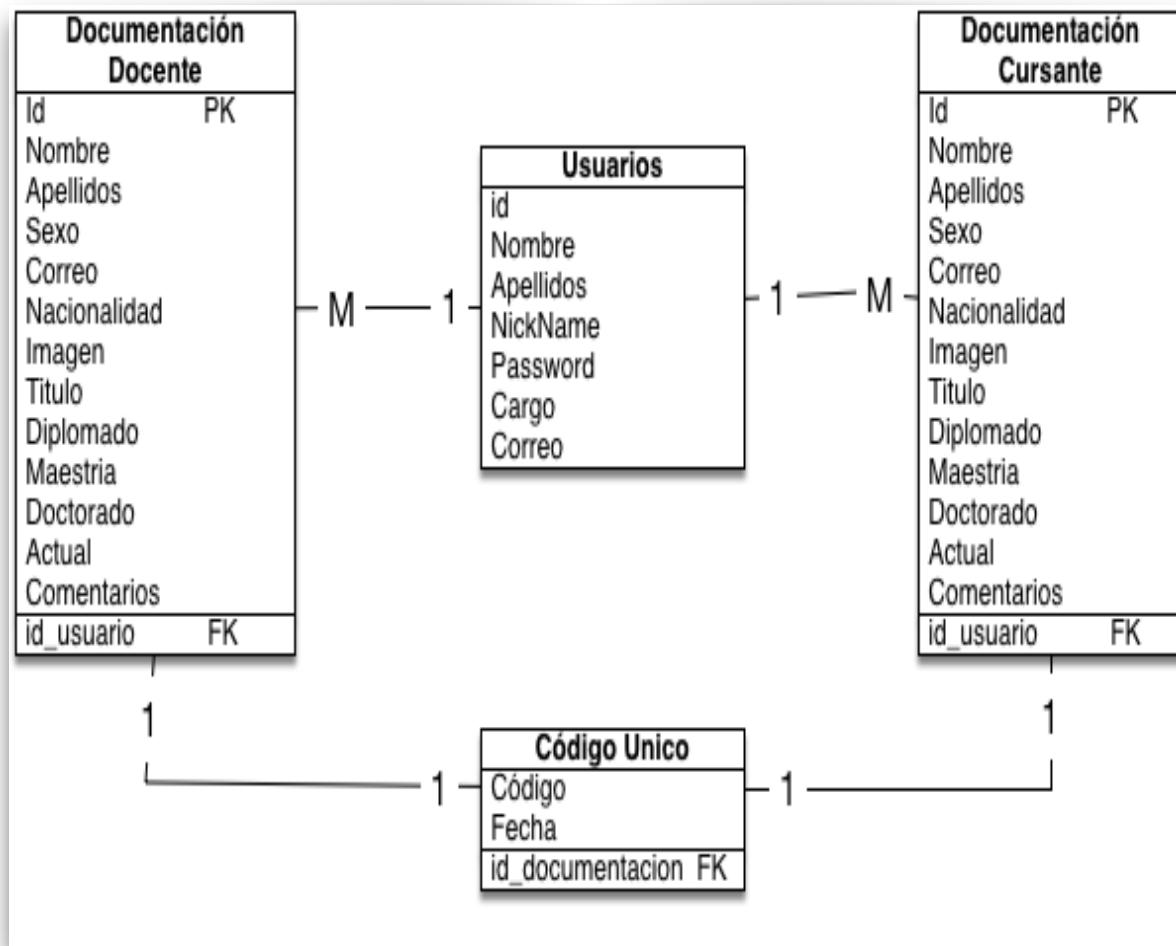


Fuente: Elaboración Propia, 2020

### 3.5.4. Actualizar la base de datos

#### 3.5.4.1. Diagrama de base de datos módulo de generación de código único

FIGURA 42: Diagrama de base de datos



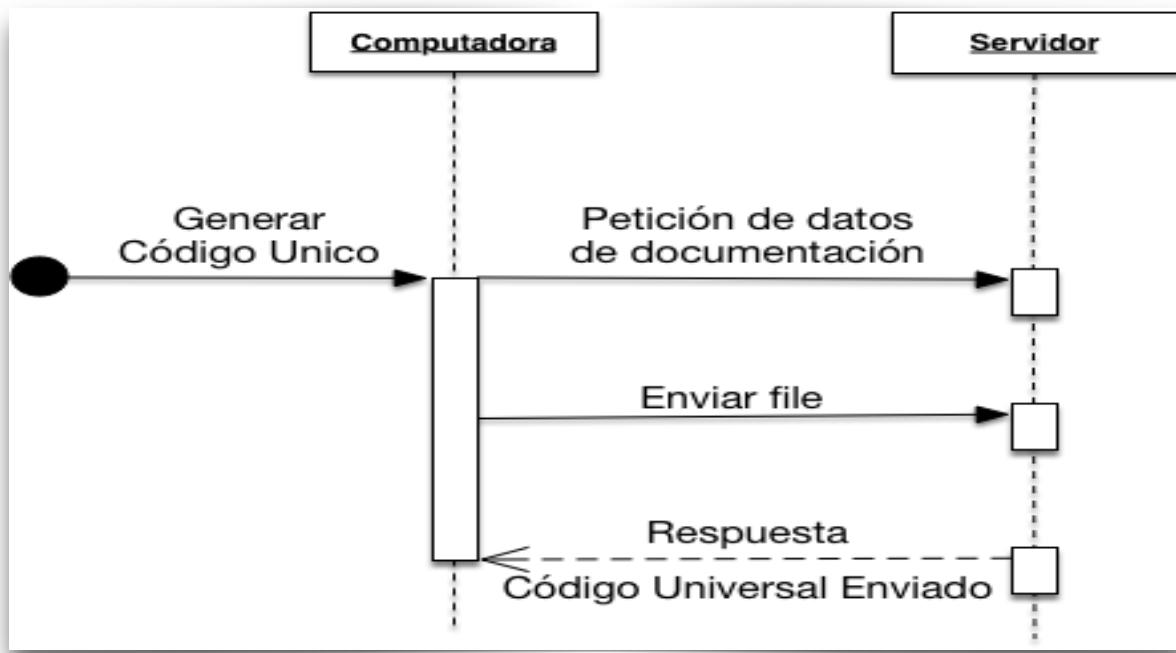
Fuente: Elaboración Propia, 2020

### 3.5.5. Diseño de la api REST para el módulo de generación de código único

Para describir el diseño de la api Restful se utiliza el diagrama de secuencia debido a que es el mas adecuado para documentar la interacción de la misma.

### 3.5.5.1. Diagrama de secuencia para el módulo de generación de código único

FIGURA 43: Diagrama de estados api REST



Fuente: Elaboración Propia, 2020

### 3.5.6. Implementar código para el módulo de generación de código único

A continuación se muestran las partes del código mas importantes de la codificación del módulo.

#### 3.5.6.1. Generación de código unico

```
const buffer = new Array();
uuidv4(null, buffer, 0); // =>
  [
    155, 29, 235, 77, 59,
    125, 75, 173, 155, 221,
    43, 13, 123, 61, 203,
    109
  ]
uuidv4(null, buffer, 16); // =>
```

```

// [
//   155, 29, 235, 77, 59, 125, 75, 173,
//   155, 221, 43, 13, 123, 61, 203, 109,
//   27, 157, 107, 205, 187, 253, 75, 45,
//   155, 93, 171, 141, 251, 189, 75, 237
// ]

```

### 3.5.6.2. Implementación en el modelo

```

import uuid

codigoS = UUIDField(default=uuid.uuid4, editable=False)

```

### 3.5.7. Realizar pruebas

**TABLA 29: Prueba 5**

Caso de prueba	
Numero de caso de prueba 5	Numero de historia de usuario:11
Nombre de caso de prueba: Prueba de generación de código único	
Descripción: Al Almacenar la documentación de docentes o cursantes se generara un código único.	
Entradas: Carga de documentación simultáneamente generando el código unico	
Resultado esperado: Generación de código único	
Evaluación: Generación satisfactoria	

*Fuente: Elaboración propia, 2020*

**TABLA 30: Pruebas de aceptación**

ID	NOMBRE DE LA PRUEBA	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	RESULTADO OBTENIDO
5	Prueba de generación de código único	Generación de código unico	Al Almacenar la documentación de docentes o cursantes se generara un código único.	Satisfactorio

*Fuente: Elaboración Propria, 2020*

### **3.6. DESARROLLO DEL MÓDULO DE GENERACION DE CÓDIGO QR A LA DOCUMENTACIÓN PRESENTADA.**

#### **3.6.1. Analizar requerimientos**

##### **3.6.1.1. Requerimientos funcionales**

- El modulo generara código QR automaticamente
- El módulo parametrizará el código QR con el código único obtenido del modulo de generación de código único
- El modulo incrustara el código QR a la documentación ya revisada por el director

##### **3.6.1.2. Requerimientos no funcionales**

- **Disponibilidad:** El sistema debe estar disponible las 24 horas para que los usuarios puedan tener acceso en cualquier momento.
- **Extensibilidad:** En base a las características de la arquitectura orientada a servicios, la plataforma podrá integrarse a otras aplicaciones dentro la organización.
- **Mantenibilidad:** Se puede agregar funcionalidades a la plataforma debido a que esta siendo desarrollado bajo la metodología de programación extrema, ademas de que esta siendo desarrollado bajo la arquitectura orientada a servicios.
- **Seguridad:** El sistema esta siendo desarrollado para ser utilizado de una manera simple y fácil de aprender para el usuario final debido a la implementación de workflow en el proyecto.

### 3.6.1.3. Tarjetas de historias de usuario del módulo de inclusión de código QR a la documentación presentada

**TABLA 31: Historia de usuario 9**

<b>HISTORIA DE USUARIO</b>	
Numero 9	Nombre: Inclusión de código QR a la documentación presentada
Usuario: Director	
Modificación de historia Número:	Entregable 4
Prioridad en negocio: Media	Puntos Estimados: 4
Riesgo en desarrollo	Puntos reales: 6
Descripción: Después de haberse registrado los datos del acreedor y almacenar el file del acreedor de la documentación, la documentación debe derivarse a otro menú para que esta documentación sea revisada, posterior a revisarse y que todo este a cabalidad, se incrustara el código QR parametrizado con el código único, siendo este un aval de garantía de veracidad de los datos y documentación.	
Observaciones:	

*Fuente: Elaboración propia, 2020*

**TABLA 32: Historia de usuario 10**

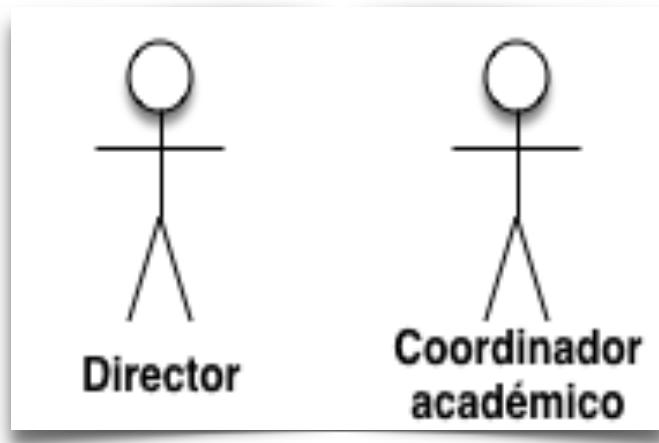
<b>HISTORIA DE USUARIO</b>	
Numero 10	Nombre: Inclusión de código QR a la documentación presentada
Usuario: Coordinador académico	
Modificación de historia Número:	Entregable 4
Prioridad en negocio: Media	Puntos Estimados: 4
Riesgo en desarrollo	Puntos reales: 6
Descripción: Después de haberse registrado los datos del acreedor y almacenar el file del acreedor de la documentación, la documentación debe derivarse a otro menú para que esta documentación sea revisada, posterior a revisarse y que todo este a cabalidad, se incrustara el código QR parametrizado con el código único, siendo este un aval de garantía de veracidad de los datos y documentación.	
Observaciones: El coordinador académico tiene la potestad de atribuirse estas tareas.	

*Fuente: Elaboración propia, 2020*

### 3.6.2. Diseño de diagramas UML para el módulo

#### 3.6.2.1. Identificación de actores

FIGURA 44: Actores del módulo



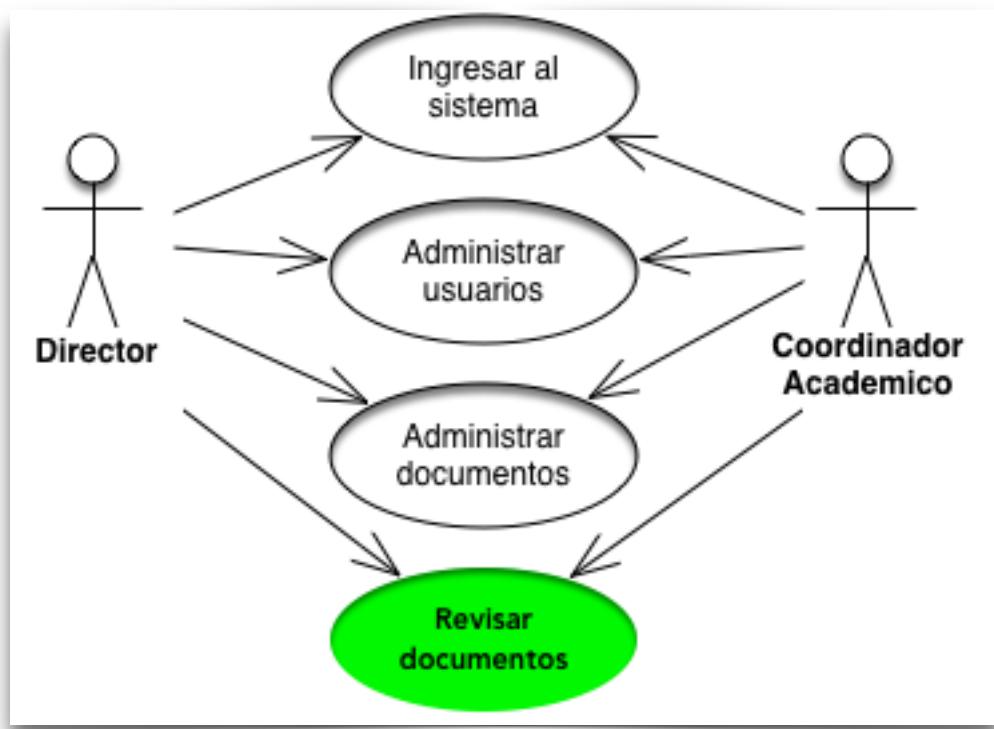
Fuente: *Elaboración Propia, 2020*

##### a) Descripción de actores

- **Director:** Este usuario al revisar la documentación pendiente, avalara que la documentación es satisfactoria y como prueba de ese aval la plataforma incrustara el código QR en la documentación revisada de modo que el QR estará parametrizado con el código único generado en el módulo de generación de código único.
- **Coordinador académico:** Ante la ausencia o imposibilidad del director de poder realizar la revisión de la documentación académica, el coordinador académico se atribuye esta responsabilidad como segundo al mando en la dirección de posgrado, permitiéndole revisar la documentación pendiente, avalara que la documentación es satisfactoria y como prueba de ese aval la plataforma incrustara el código QR en la documentación revisada de modo que el QR estará parametrizado con el código único generado en el módulo de generación de código único.

### 3.6.2.2. Diagramas de caso de uso

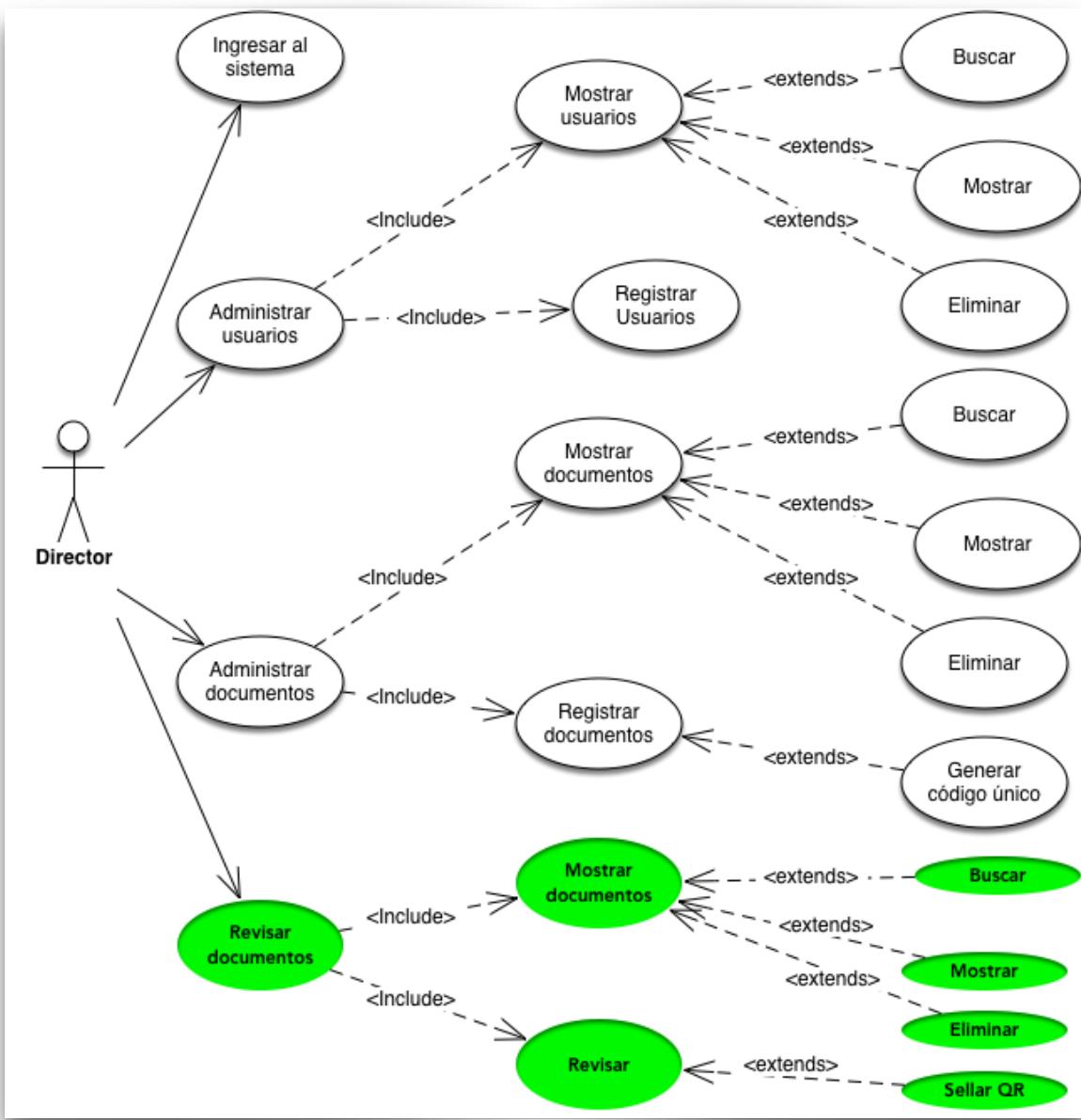
FIGURA 45: Diagrama de caso de uso



Fuente: Elaboración Propia, 2020

El director y el coordinador académico comparten los mismos permisos debido a que el coordinador académico es el segundo al mando y ante la ausencia del director este se atribuye como responsabilidad la revisión de documentos almacenados en la plataforma de cursantes o docentes.

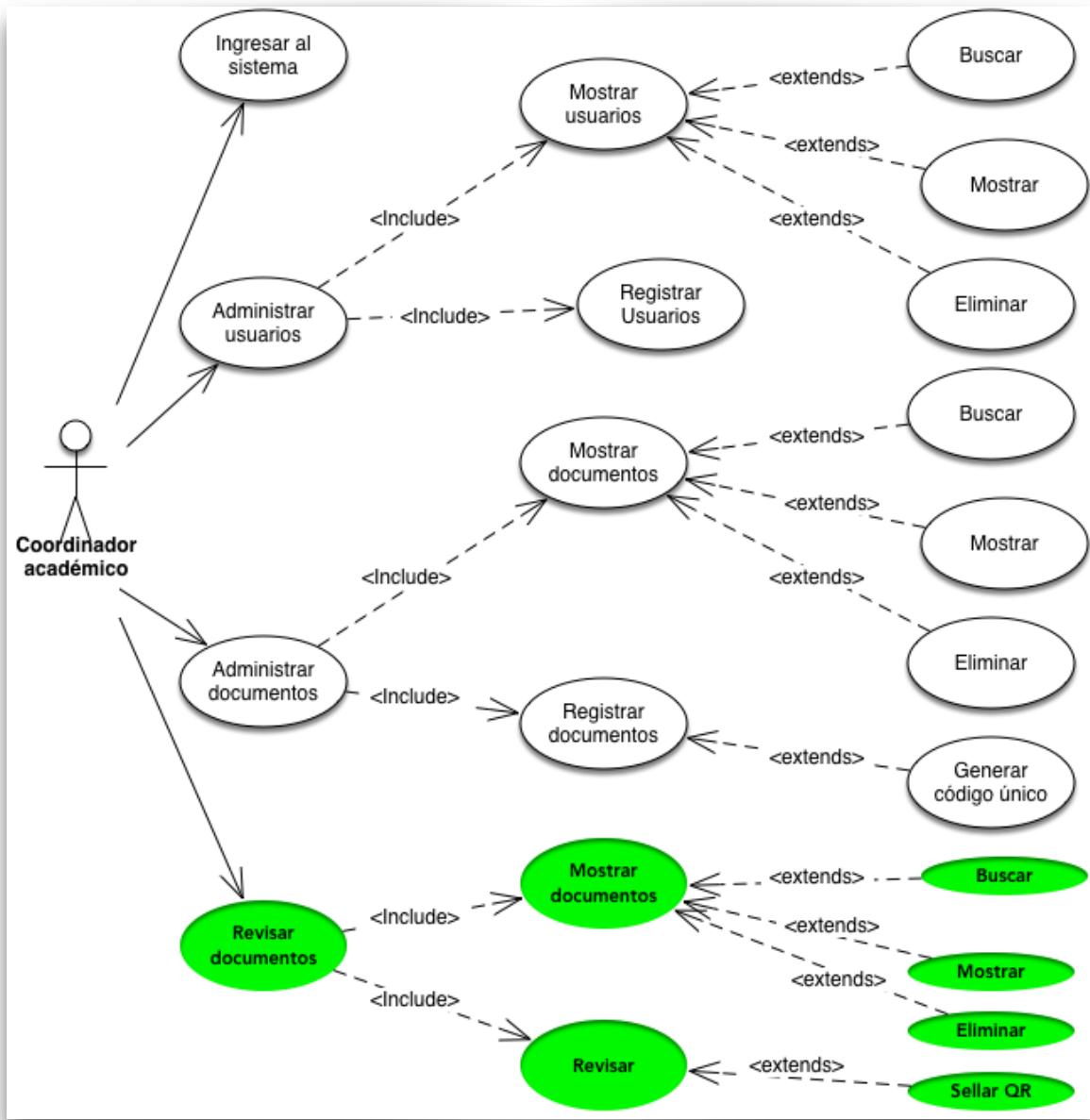
FIGURA 46: Actor director



Fuente: Elaboración Propia, 2020

El director revisa la documentación derivada por la secretaría para que este revise la documentación almacenada, este podrá buscar documentos y revisarlos, para cambiar el estado de los documentos de “No revisado” a “Revisado”, esta proceso se lo avalara mediante un marcado con un código QR que tendrá parametrizado el código único del documento presentado mas la fecha y hora del marcado con el código QR, cambiando el estado de dicha documentación a “revisado”.

**FIGURA 47: Actor coordinador académico**



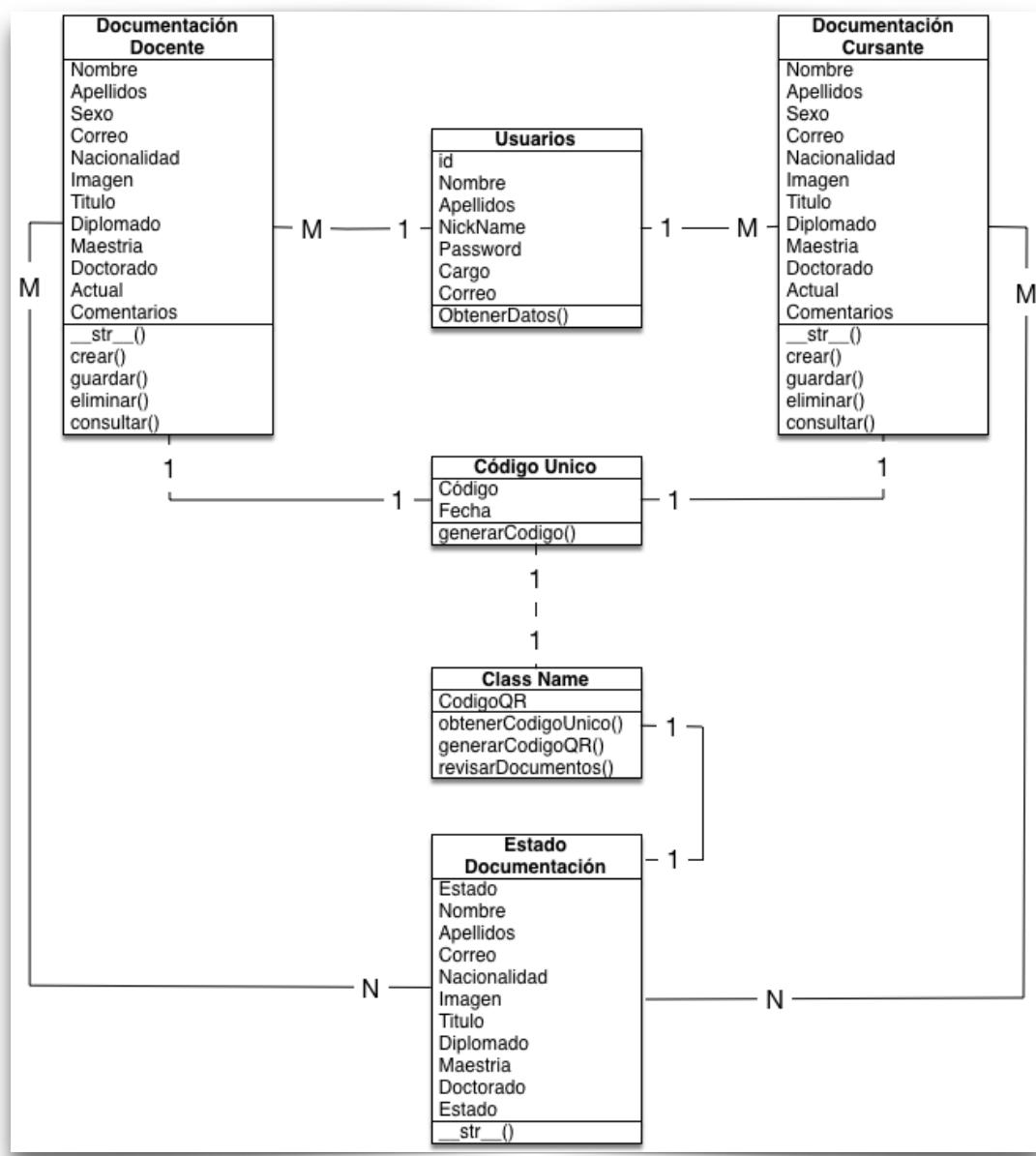
Fuente: Elaboración Propia, 2020

El coordinador académico ante ausencia o imposibilidad de que el director realice la revisión de documentos, este se atribuye realizar dicho proceso, revisa la documentación derivada por la secretaría para que este revise la documentación almacenada, este podrá buscar documentos y revisarlos, para cambiar el estado de los documentos de “No revisado” a “Revisado”, esta proceso se lo avalara mediante un marcado con un código QR que tendrá parametrizado el código único

del documento presentado mas la fecha y hora del marcado con el código QR, cambiando el estado de dicha documentación a “revisado”.

### 3.6.2.3. Diagrama de clases

FIGURA 48: Diagrama de clases

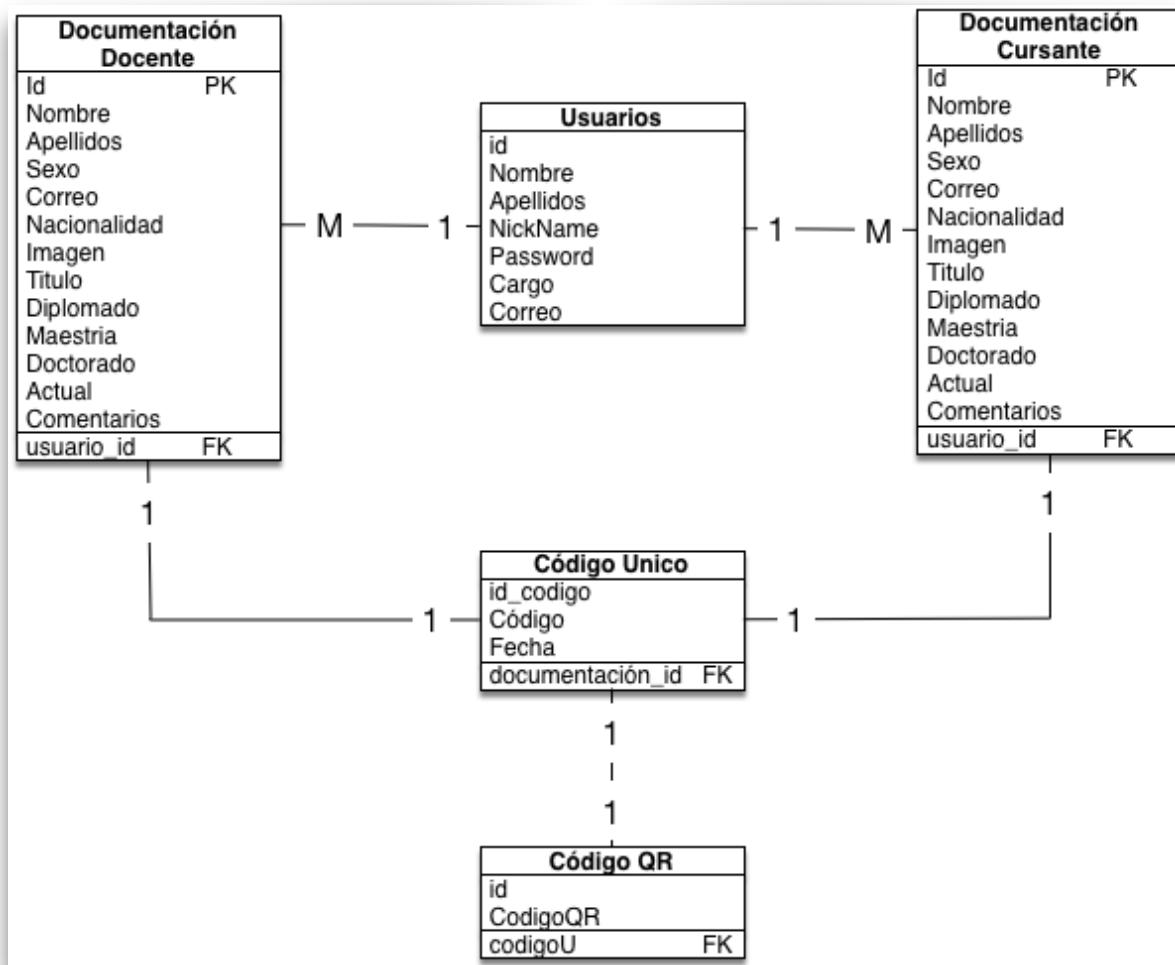


Fuente: Elaboración Propia, 2020

### 3.6.3. Actualizar la base de datos

#### 3.6.3.1. Diagrama de base de datos

FIGURA 49: Diagrama de base de datos

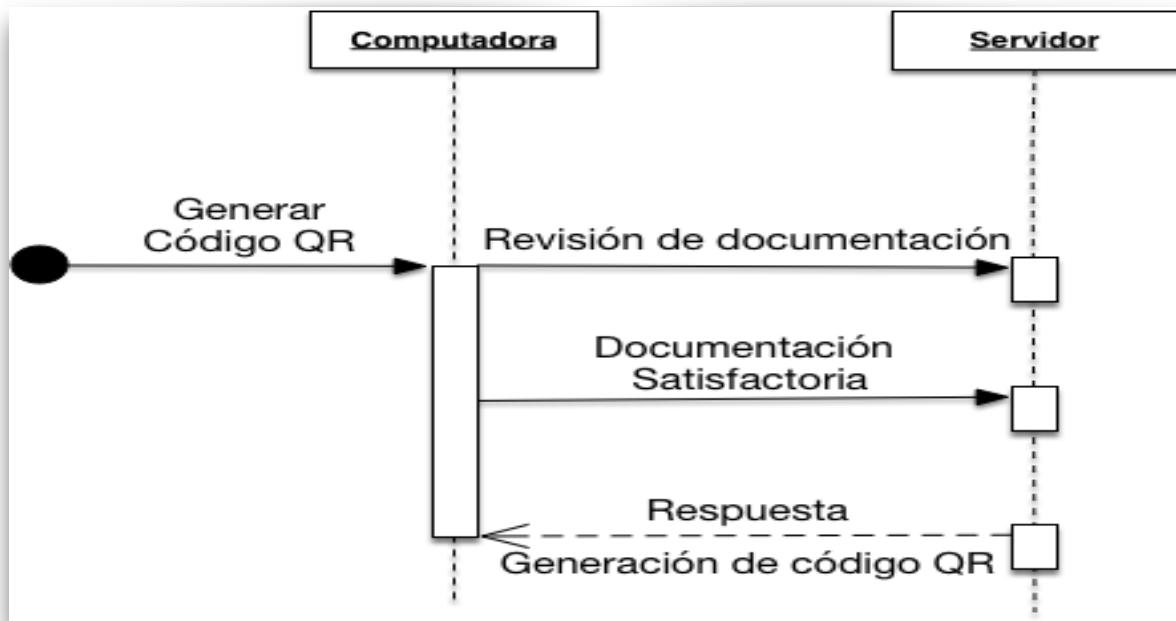


Fuente: Elaboración Propia, 2020

### 3.6.4. Diseño de servicios de la api REST

### 3.6.5. Diagrama de secuencia para el módulo de inclusión de código QR a la documentación presentada

FIGURA 50: Diagrama de estados api REST



Fuente: Elaboración Propia, 2020

### 3.6.6. Implementar código para el modulo de inclusión de código QR a la documentación presentada

A continuación se muestran las partes del código mas importantes de la codificación del módulo.

#### 3.6.6.1. Modelo de inclusion de QR

Se genera el modelo de QR en models.py para su creación siguiendo el modelo vista template.

```
class QR(models.Model):  
    name = 'qr_codigo'
```

```
verbose_name = 'QR'
```

### 3.6.6.2. Generar QR

Se define la clase QR funciona de QR en la dirección views.py para renderizarlo en las vistas del proyecto.

```
def QR(request):  
  
    context = dict(  
        docente=DocenteModel,  
        cursanteModel=cursanteModel,  
        Código=uuid.uuid4,  
    )  
    # Render the index page.  
    return render(request, 'qr_codigo/consultar.html', context=context)  
img = Image.open("/Users/oscarfrancisco/Downloads/docu.jpg")  
#img = img.resize((2500, 2500), Image.ANTIALIAS)  
#imgAgua= Image.open("/Users/oscarfrancisco/Downloads/frame.png")  
imagen = imagen.resize((100, 100), Image.ANTIALIAS)  
img.paste(imagen, (305, 525))  
img.save("/Users/oscarfrancisco/Downloads/imagenFinal3.jpg")  
imagen = qrcode.make(codigoS) # Creamos un código a partir de una cadena  
de texto  
imagen.save("/Users/oscarfrancisco/Downloads/frame1.png")
```

### 3.6.7. Realizar pruebas

TABLA 33: Prueba 6

Caso de prueba	
Número de caso de prueba 6	Número de historia de usuario:12, 13
Nombre de caso de prueba: Prueba de generación de código único	
Descripción: Realizar un sellado con el código QR al documento revisado	
Entradas: Revisar documentación, cambiar su estado a revisado	
Resultado esperado: Inclusión de código QR en proceso	
Evaluación: Inclusión en proceso	

Fuente: Elaboración propia, 2020

**TABLA 34: Pruebas de aceptación**

ID	NOMBRE DE LA PRUEBA	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	RESULTADO OBTENIDO
6	Prueba a la inclusión de código QR a la documentación presentada	Inclusión de código QR a la documentación presentada	Realizar un sellado con el código QR al documento revisado	En proceso

*Fuente: Elaboración Propia, 2020*

### **3.7. GENERACION REPORTES EN BASE A MÓDULOS IMPLEMENTADOS**

#### **3.7.1. Analizar requerimientos**

##### **3.7.1.1. Requerimientos funcionales**

- El modulo debe ser capaz de realizar reportes acerca de los cursantes y docentes que presentaron documentación
- El módulo otorgara reportes del estado de la documentación, si fue revisada o si no fue revisada

##### **3.7.1.2. Requerimientos no funcionales**

- **Disponibilidad:** El sistema debe estar disponible las 24 horas para que los usuarios puedan tener acceso en cualquier momento.
- **Extensibilidad:** En base a las características de la arquitectura orientada a servicios, la plataforma podrá integrarse a otras aplicaciones dentro la organización.
- **Mantenibilidad:** Se puede agregar funcionalidades a la plataforma debido a que esta siendo desarrollado bajo la metodología de programación extrema, ademas de que esta siendo desarrollado bajo la arquitectura orientada a servicios.

- **Seguridad:** El sistema está siendo desarrollado para ser utilizado de una manera simple y fácil de aprender para el usuario final debido a la implementación de workflow en el proyecto.

### **3.7.1.3. Tarjetas de historias de usuario del módulo de generación de reportes en base a modelos implementados**

**TABLA 35: Historia de usuario 11**

<b>HISTORIA DE USUARIO</b>	
Numero 11	Nombre: Generación de reportes en base a módulos implementados
Usuario: Director	
Modificación de historia Número:	Entregable 5
Prioridad en negocio: Media	Puntos Estimados: 8
Riesgo en desarrollo	Puntos reales: 7
Descripción: Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual “Revisada” o “No revisada”, ademas de desplegar la documentación Revisada en un menú y la No Revisada en otro menú, ademas de poder obtener un reporte de los usuarios registrados, sus roles y privilegios respectivos.	
Observaciones:	

*Fuente: Elaboración propia, 2020*

**TABLA 36: Historia de usuario 12**

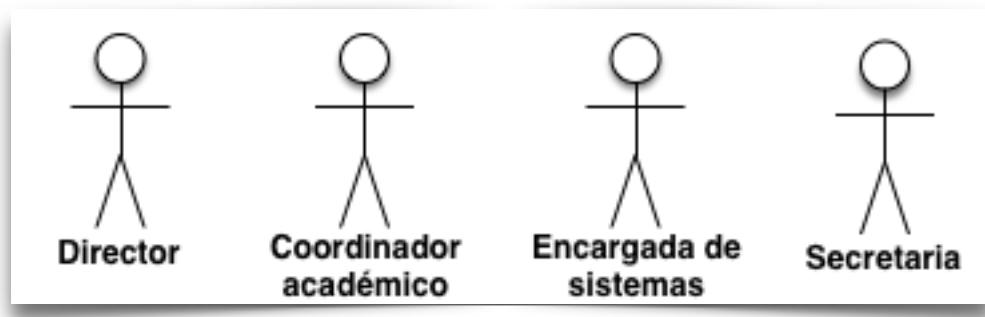
<b>HISTORIA DE USUARIO</b>	
Numero 12	Nombre: Generación de reportes en base a módulos implementados
Usuario: Coordinador académico	
Modificación de historia Número:	Entregable 5
Prioridad en negocio: Media	Puntos Estimados: 8
Riesgo en desarrollo	Puntos reales: 7
Descripción: Se podrá tener acceso a toda la documentación y datos almacenados, pudiendo recibir reportes de la documentación actual “Revisada” o “No revisada”, ademas de desplegar la documentación Revisada en un menú y la No Revisada en otro menú, ademas de poder obtener un reporte de los usuarios registrados, sus roles y privilegios respectivos.	
Observaciones: El coordinador académico tiene la potestad de atribuirse estas tareas.	

*Fuente: Elaboración propia, 2020*

### 3.7.2. Casos de uso del módulo de generación de reportes

#### 3.7.2.1. Identificación de actores

FIGURA 51: Actores del módulo



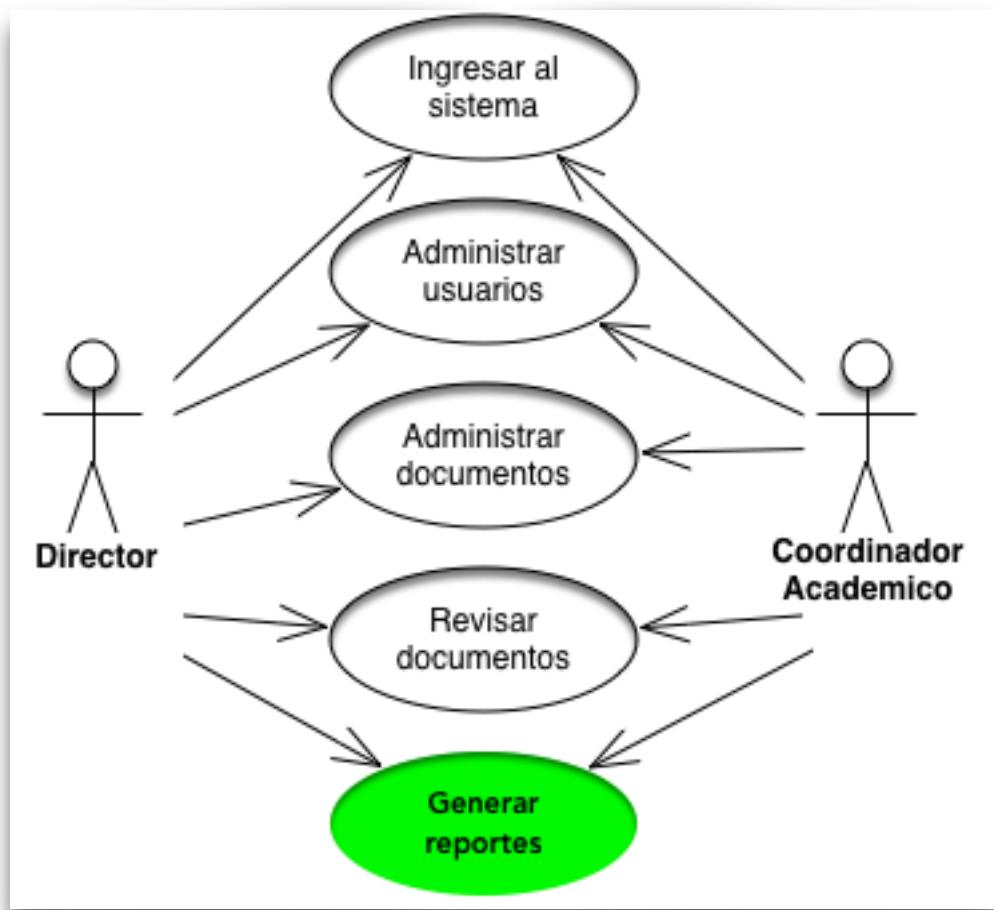
Fuente: *Elaboración Propia, 2020*

##### a) Descripción de actores

- **Director:** Este usuario obtendrá reportes de la documentación no revisada, así mismo de la documentación revisada, listado de todos los cursantes y docentes que presentaron documentación
- **Coordinador académico:** Obtendrá reportes de la documentación no revisada, así mismo de la documentación revisada, listado de todos los cursantes y docentes que presentaron documentación
- **Encargada de sistemas:** Podrá obtener reportes de la documentación presentada por cursantes o docentes.
- **Secretaria:** Obtendrá reportes del estado de la documentación almacenada

### 3.7.2.2. Casos de uso

FIGURA 52: Diagrama de casos de uso

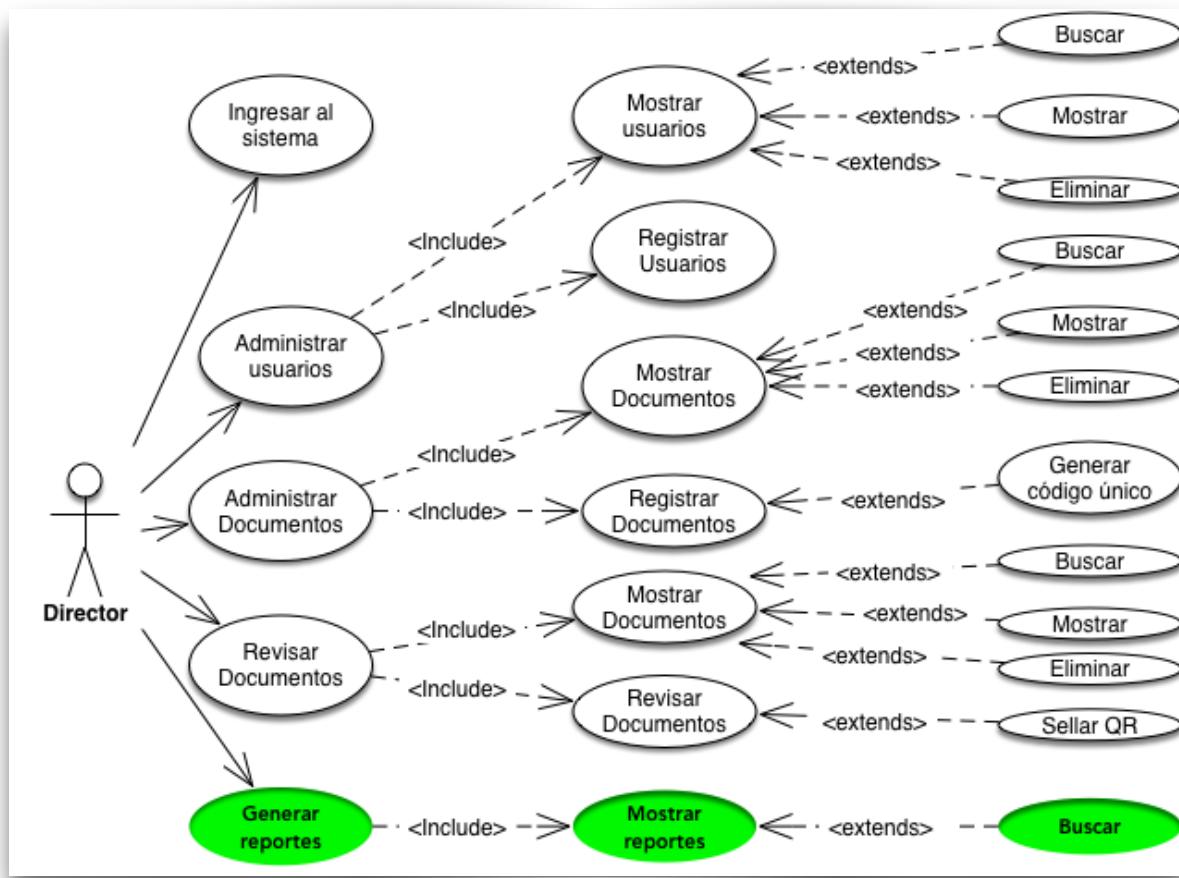


Fuente: Elaboración Propia, 2020

En este módulo todos los actores se involucran en las consultas, unos con permisos mas amplios con respecto a otros dentro de la plataforma, se detallan a continuación.

El director tiene acceso a los reportes, pudiendo buscar documentos revisados, documentos de docentes y cursantes, ademas de buscar documentos no revisados.

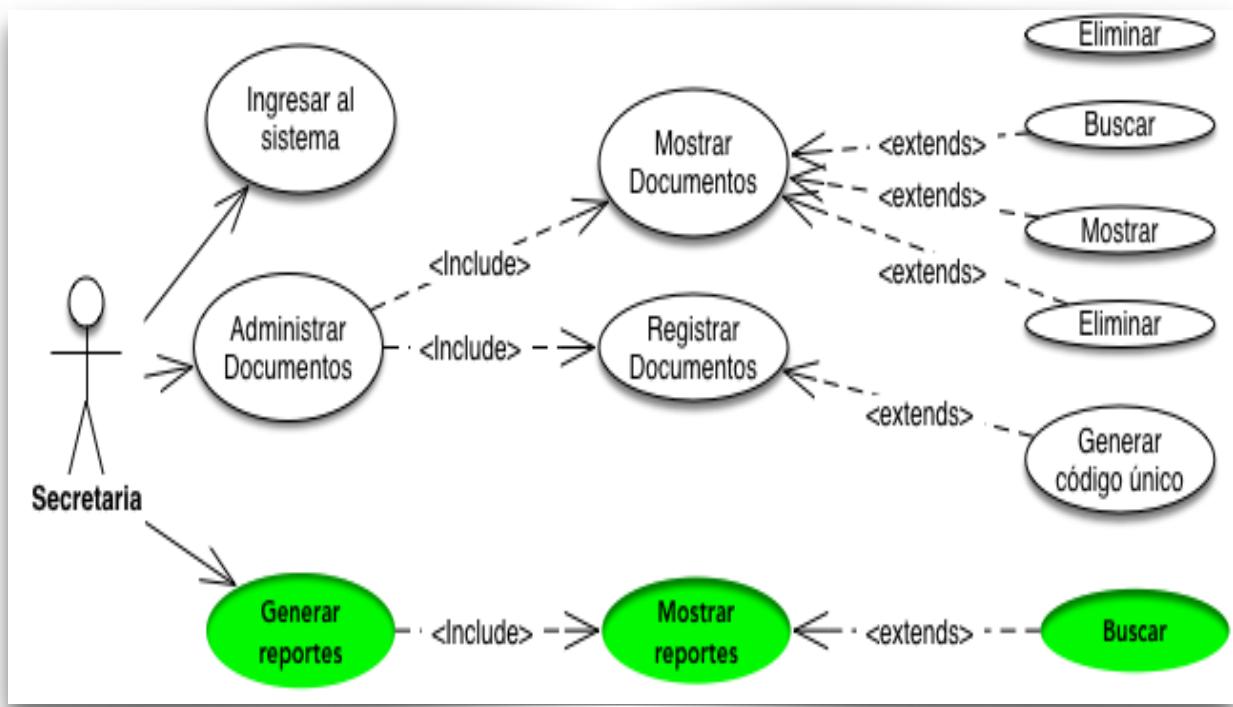
**FIGURA 53: Actor director**



Fuente: Elaboración Propia, 2020

La secretaria tiene acceso a los reportes, pudiendo buscar documentos, también puede revisar documentos de docentes o cursantes individualmente.

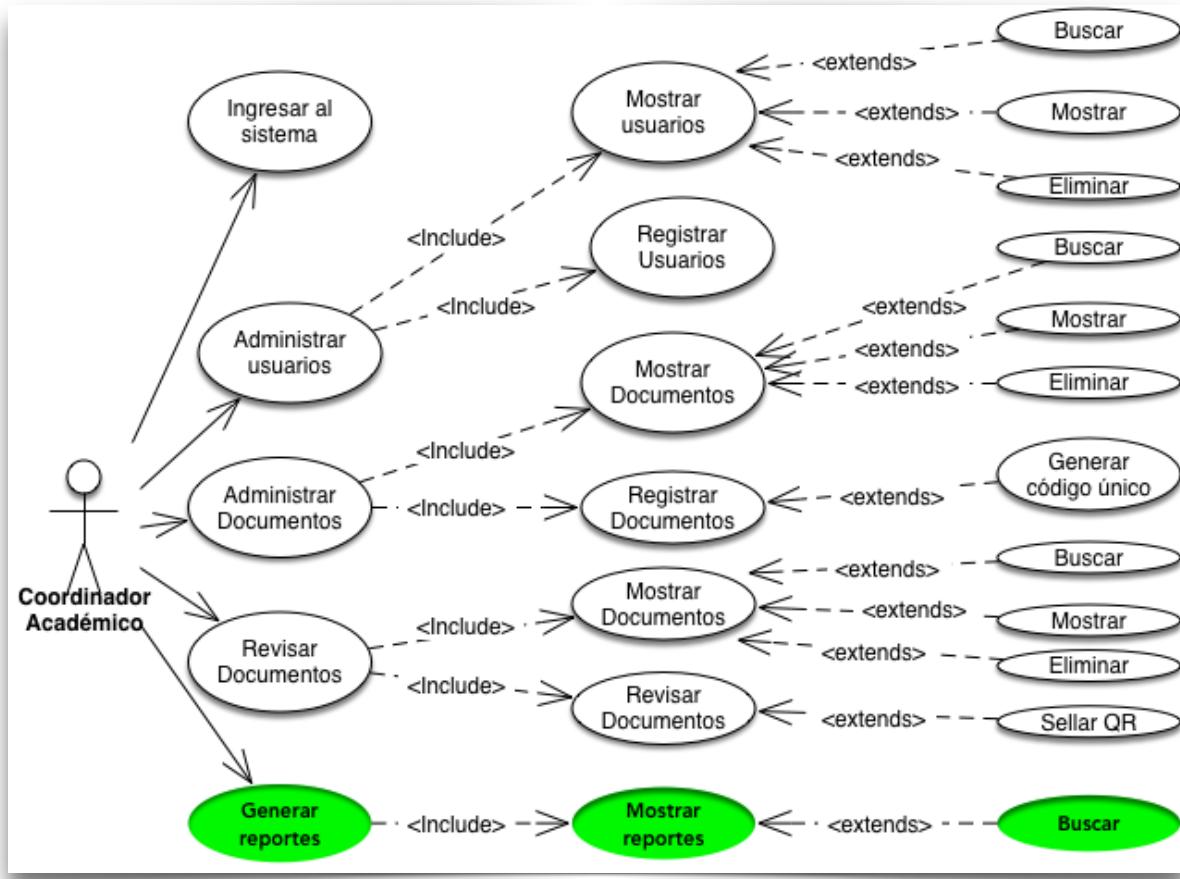
**FIGURA 54: Actor secretaria**



Fuente: Elaboración Propia, 2020

El coordinador académico tiene acceso a los reportes, pudiendo buscar documentos revisados, documentos de docentes y cursantes, ademas de buscar documentos no revisados.

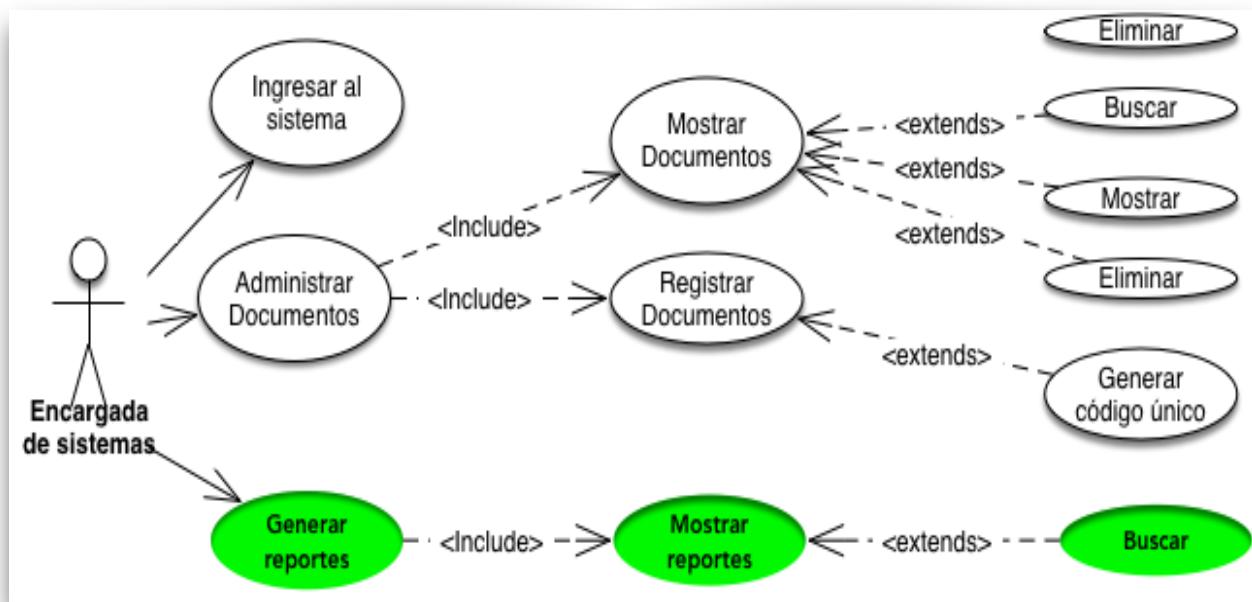
**FIGURA 55: Actor coordinador academico**



Fuente: Elaboración Propia, 2020

La encargada de sistemas tiene acceso a los reportes, pudiendo buscar documentos, también puede revisar documentos de docentes o cursantes individualmente.

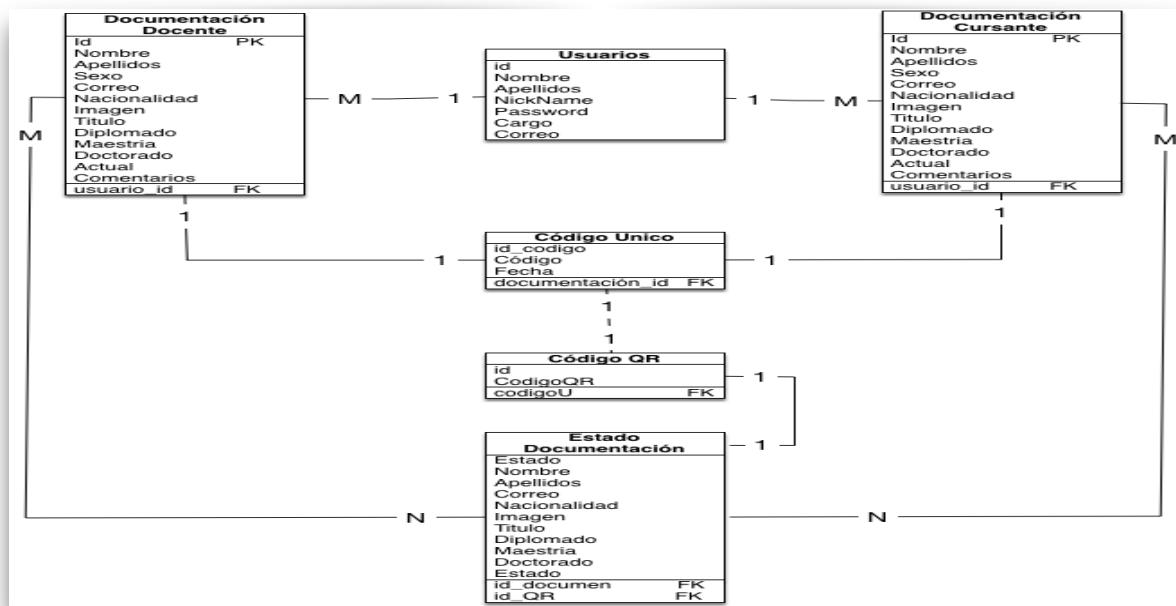
**FIGURA 56: Actor encargada de sistemas**



Fuente: Elaboración Propia, 2020

### 3.7.3. Diagrama de base de datos del módulo de generación de reportes

**FIGURA 57: Diagrama de base de datos**

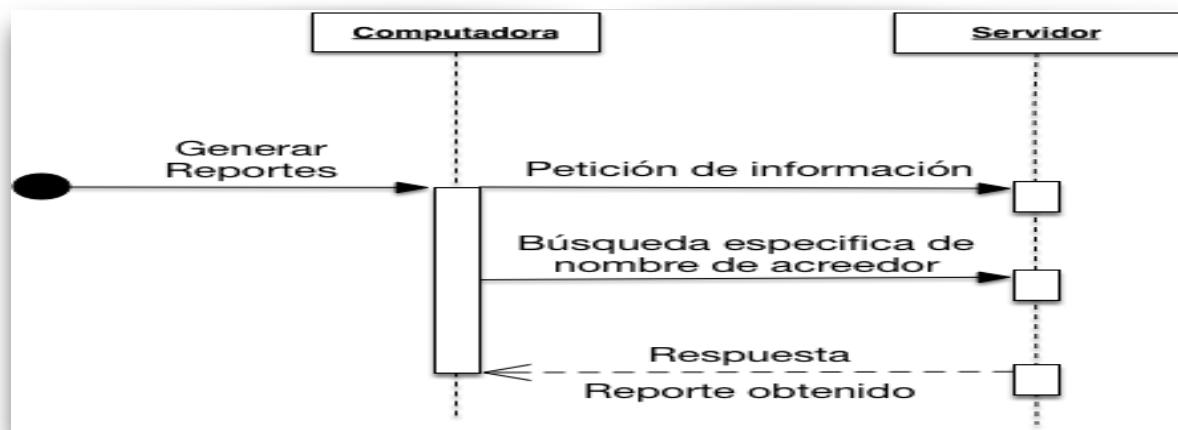


Fuente: Elaboración Propia, 2020

### 3.7.4. Diseño de servicios de la api REST

#### 3.7.4.1. Diagrama de secuencia para servicios de la api REST del módulo de generación de reportes

FIGURA 58: Diagrama de estados api REST



Fuente: Elaboración Propia, 2020

#### 3.7.4.2. Realizar pruebas al módulo

TABLA 37: Prueba 7

Caso de prueba	
Numero de caso de prueba 7	Numero de historia de usuario:14,15
Nombre de caso de prueba: Prueba a la generación de reportes simple	
Descripción: Desplegar un formulario de toda la documentación recepcionada	
Entradas: Ingreso al menu de reportes	
Resultado esperado: Desplegar formulario de toda la documentación excepcionada	
Evaluación: En proceso	

Fuente: Elaboración propia, 2020

TABLA 38: Prueba 8

Caso de prueba	
Numero de caso de prueba 8	Numero de historia de usuario:16, 17

Caso de prueba
Nombre de caso de prueba: Prueba a la generación de reportes
Descripción: Desplegar un formulario con toda la documentación sin revisar
Entradas: Ingreso al menú de documentación sin revisar
Resultado esperado: Desplegar formulario de toda la documentación no revisada
Evaluación: En proceso

Fuente: *Elaboración propia, 2020*

**TABLA 39: Pruebas de aceptación de módulo de reportes**

ID	NOMBRE DE LA PRUEBA	HISTORIA DE USUARIO	CRITERIO DE ACEPTACIÓN	RESULTADO OBTENIDO
7	Prueba a la generación de reportes simple	Generación de reportes en base a módulos implementados	Desplegar un formulario de toda la documentación recepcionada	En proceso
8	Prueba a la generación de reportes	Generación de reportes en base a módulos implementados	Desplegar un formulario de toda la documentación recepcionada	En proceso
			Desplegar un formulario con toda la documentación sin revisar	En proceso

Fuente: *Elaboración Propia, 2020*

### 3.8. REALIZAR PRUEBAS AL SISTEMA CONCLUIDO

#### 3.8.1. Selección de pruebas adecuadas al proyecto

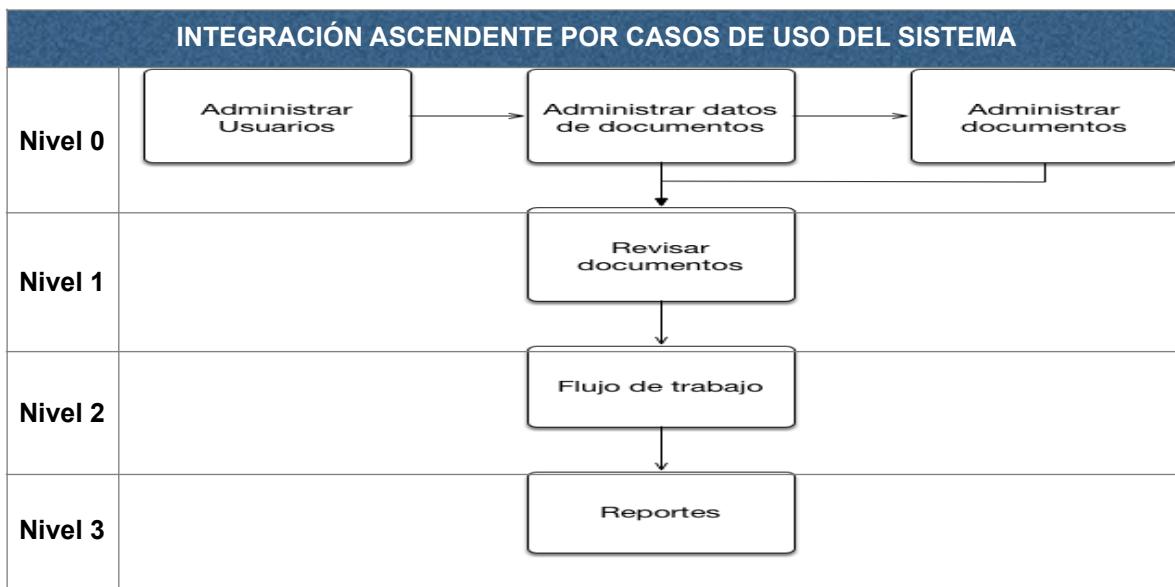
Debido a que la plataforma desarrollada es de gestión documental ECM (por sus siglas en inglés) es de suma importancia realizar pruebas de integridad que permitirán demostrar que los datos almacenados en la plataforma son coherentes. Por otro lado para verificar constantemente que los datos introducidos cumplen con formatos definidos y a la vez evitar en lo posible los errores de transcripción que puede cometer el usuario, se llevaron a cabo pruebas de validación de campos. A continuación se describen los resultados de pruebas obtenidas.

### **3.8.2. Documentación de pruebas realizadas**

A continuación se documentan los detalles de las pruebas realizadas

### **3.8.3. Pruebas de integración**

**TABLA 40: Pruebas de integración**



*Fuente: Elaboración Propia, 2020*

#### **3.8.3.1. Nivel 0**

Para el nivel 0 solo la administración de usuarios y administración de datos de documentos son casos de uso que no dependen de otros módulos, el caso de uso de administración de documentos no es independiente y funciona con los datos que se derivan de la administración de datos de documentos.

Como administrador se deberá tener acceso a la información actualizada de los módulos de administrar usuarios, administrar datos de documentos y administrar documentos.

**TABLA 41: Pruebas de integración nivel 0**

CASO DE USO	CASO DE USO ANTECESOR	PRUEBAS	ESTADO
<b>Administrar usuarios</b>	Ninguno	Verificar usuarios registrados en la base de datos.	Satisfactorio
<b>Administrar datos de documentos</b>	Ninguno	Verificar registro de datos de documentos, tales como dueño del acreedor de los documentos, fecha de registro, etc.	Satisfactorio
<b>Administrar documentos</b>	Ninguno	Verificar que la documentación se haya almacenado correctamente en la base de datos	Satisfactorio

*Fuente: Elaboración Propia, 2020*

### 3.8.3.2. Nivel 1

Para el nivel 1 se debe enfatizar que este depende del nivel 0, para poder revisar los documentos se deben haber previamente realizado el registro de datos de los documentos y haber cargado a la plataforma los documentos.

Como administrador de debe tener acceso a la documentación no revisada actualizada para poder realizar la revisión correspondiente.

**TABLA 42: Pruebas de integración nivel 1**

CASO DE USO	CASO DE USO ANTECESOR	PRUEBAS	ESTADO
<b>Revisar documentos</b>	Administrar documentos, administrar datos de documentos	Visualizar la documentación almacenada en el sistema, clasificada como no revisados, poder marcar la documentación para que esta incluya un código QR y cambie de estado a “revisado”.	En proceso

*Fuente: Elaboración Propia, 2020*

### **3.8.3.3. Nivel 2**

Para el nivel 2 se recalca que este depende de los niveles 0 y 1, para realizar las acciones de cada módulo.

Como administrador debería tener acceso a la información actualizada de cada modulo, administrar usuarios, administrar datos de documentos, administrar documentos, revisar documentos, usando la técnica de los escenarios.

**TABLA 43: Pruebas de integración nivel 2**

CASO DE USO	CASO DE USO ANTECESOR	PRUEBAS	ESTADO
Flujo de trabajo	Revisar documentos	Verificar los estados en el que se encuentra la documentación	En proceso

*Fuente: Elaboración Propia, 2020*

### **3.8.3.4. Nivel 3**

Para el nivel 3, para obtener reportes actualizados de los diferentes módulos implementados.

**TABLA 44: Pruebas de integración nivel 3**

CASO DE USO	CASO DE USO ANTECESOR	PRUEBAS	ESTADO
Obtener reportes	Flujo de trabajo	Realizar consultas a la base de datos para realizar reportes correspondientes	Listado de las consultas realizadas a la base de datos

*Fuente: Elaboración Propia, 2020*

Con el resultado presentado en los diferentes niveles de las tablas anteriores se ha demostrado que los resultados son los esperados en cuestión de validaciones en los campos.

## 4. ANALISIS DE VIABILIDAD

### 4.1. VIABILIDAD TÉCNICA

Para la viabilidad técnica se analizan los requerimientos mínimos de hardware y software necesarios para la correcta funcionalidad del sistema, ademas se proponen algunos requerimientos ideales para el funcionamiento optimo del sistema.

**TABLA 45: Requerimiento mínimo e ideal de hardware**

NOMBRE	REQUERIMIENTO MÍNIMO	PRECIO Bs	REQUERIMIENTO IDEAL	PRECIO Bs
Hardware	Procesador core i3	769	Procesador core i7	2498
	Memoria Ram 4 Gb	257	Memoria Ram 8 Gb	1045
	Disco duro 500 Gb	264	Disco duro 1 Tb	1392
	Tarjeta de video AMD radeon RX 570	417	Tarjeta de video NVIDE GEFORCE GTX 1050Ti	1879
	Teclado y raton	150	Teclado y raton inalambricos	220
<b>Costo total</b>		1964		7264

Fuente: Elaboración Propia, 2020

**TABLA 46: Requerimiento mínimo e ideal de software**

NOMBRE	REQUERIMIENTO MÍNIMO	PRECIO Bs	REQUERIMIENTO IDEAL	PRECIO Bs
Hardware	Sistema operativo Linux ubuntu	Gratis	Sistema operativo mac OS High Sierra	2498
	Navegador Internet explorer, firefox.	Gratis	Navegador: Safari, Google Chrome, Firefox,Opera.	Gratis
	Lenguaje de programación Python 3.7	Gratis	Lenguaje de programación Python 3.7	Gratis

NOMBRE	REQUERIMIENTO MÍNIMO	PRECIO Bs	REQUERIMIENTO IDEAL	PRECIO Bs
Hardware	Gestor de base de datos mySQL	Gratis	PostgreSQL	Gratis
Costo total	0		2498	

*Fuente: Elaboración Propia, 2020*

Según el modelo de COCOMO de estimación de costos se diferencian tres tipos de proyectos de software para la realización del cálculo de costo y duración.

**Modo orgánico:** Cuando el ambiente es desarrollado en un ambiente familiar y estable y en el que el producto es similar a otros proyectos desarrollados previamente. Son además productos pequeños y poco novedosos, con unos requerimientos definidos y poco rígidos.

**Modo semiacoplado:** Es un modelo para proyectos que se presentan características intermedias entre el orgánico y el empotrado, con un equipo formado por miembros de distintos niveles de experiencia que trabajan sobre un conjunto de requisitos, más o menos flexibles.

**Modo empotrado:** para proyectos caracterizados por unos requerimientos y restricciones poco flexibles, que requieren un gran esfuerzo de innovación, además de poseer un elevado nivel de complejidad en software.

**TABLA 47: Constante por modo de desarrollo**

MODO	A	B	C	D
ORGANICO	3.2	1.05	2.5	0.38
SEMIACOPLADO	3.0	1.12	2.5	0.35
EMPOTRADO	2.8	1.2	2.5	0.32

*Fuente: Elaboración Propia, 2020*

El modo que se utilizará en el proyecto, será el modo semiacoplado debido a que este es un proyecto que cuenta con miembros en diferentes niveles de experiencia en los distintos lenguajes de programación que se está desarrollando, además de ser un sistema complejo y los requisitos son poco flexibles para su desarrollo.

COCOMO esta definido en términos de tres modelos que reflejan el nivel de detalle considerado a lo hora de realizar costos.

**TABLA 48: Ecuaciones básicas en COCOMO**

MODO	Ecuaciones básicas del modelo
<b>ORGANICO</b>	Esfuerzo = EAF * 3.2 * KSLOC
<b>SEMIACOPLADO</b>	Esfuerzo = EAF * 3.0 * KSLOC
<b>EMPOTRADO</b>	Esfuerzo = EAF * 2.8 * KSLOC

*Fuente: Elaboración Propia, 2020*

**TABLA 49: Ecuaciones básicas en COCOMO intermedio**

MODO	Ecuaciones básicas del modelo
<b>ORGANICO</b>	Tiempo de desarrollo = 2.5 * Esfuerzo
<b>SEMIACOPLADO</b>	Tiempo de desarrollo = 2.5 * Esfuerzo
<b>EMPOTRADO</b>	Tiempo de desarrollo = 2.5 * Esfuerzo

*Fuente: Elaboración Propia, 2020*

A continuación se describen las variables y sus unidades de anexiones para su uso.

E = Esfuerzo en unidades de [Personas/mes]

KSLOC = Kilolineas o miles de lineas de código

EAF = Multiplicacion de 15 factores de costo

T = Tiempo de duración del desarrollo en unidades de meses

Se utilizo el modelo intermedio porque establece que 15 factores de costo, que determinan la duración y el costo, de esta manera se obtiene mejores resultados.

**TABLA 50: Factores de riesgo en COCOMO**

	Factores de costo		Muy bajo	Bajo	Medio	Alto	Muy alto	Extra alto
<b>Atributos del producto</b>	F i a b l i d a d requerida del software	RELY	0.75	0.88	1	1.15	1.4	-
	Tamaño de la base de datos	DATA	-	0.94	1	1.08	1.16	-
	Complejidad del producto	CPLX	0.75	0.85	1	1.15	1.3	1.65

	Factores de costo		Muy bajo	Bajo	Medio	Alto	Muy alto	Extra alto
<b>Atributos del computador</b>	Restricciones del tiempo en ejecucion	TIME	-	-	1	1.11	1.3	1.66
	Restricciones del almacenamiento principal	STOR	-	-	1	1.06	1.21	1.56
	Volatilidad de la maquina virtual	VIRT	-	0.87	1	1.15	1.3	-
	Tiempo de respuesta del ordenador	TURN	-	0.87	1	1.07	1.15	-
<b>Atributos del personal</b>	Capacidad del analista	ACAP	1.46	1.19	1	0.86	0.71	-
	Experiencia en aplicaciones	AEXP	1.29	1.13	1	0.91	0.82	-
	Capacidad de los programadores	PCAP	1.42	1.17	1	0.86	0.7	-
	Experiencia en sistema operativo utilizado	VEXP	1.21	1.1.	1	0.95	-	-
	Experiencia con el lenguaje de programación	LEXP	1.14	1.07	1	0.95	-	-
<b>Atributos del proyecto</b>	Practicas de programación modernas	MODP	1.24	1.1	1	0.91	0.82	-
	Utilización de herramientas de software	TOOL	1.24	1.1	1	0.91	0.83	-
	Limitaciones de planificación del proyecto	SCED	1.23	1.08	1	1.04	1.1	-

Fuente: Elaboración propia, 2020

Nota: Los valores marcados con azul se consideran seleccionados para la estimación de costo.

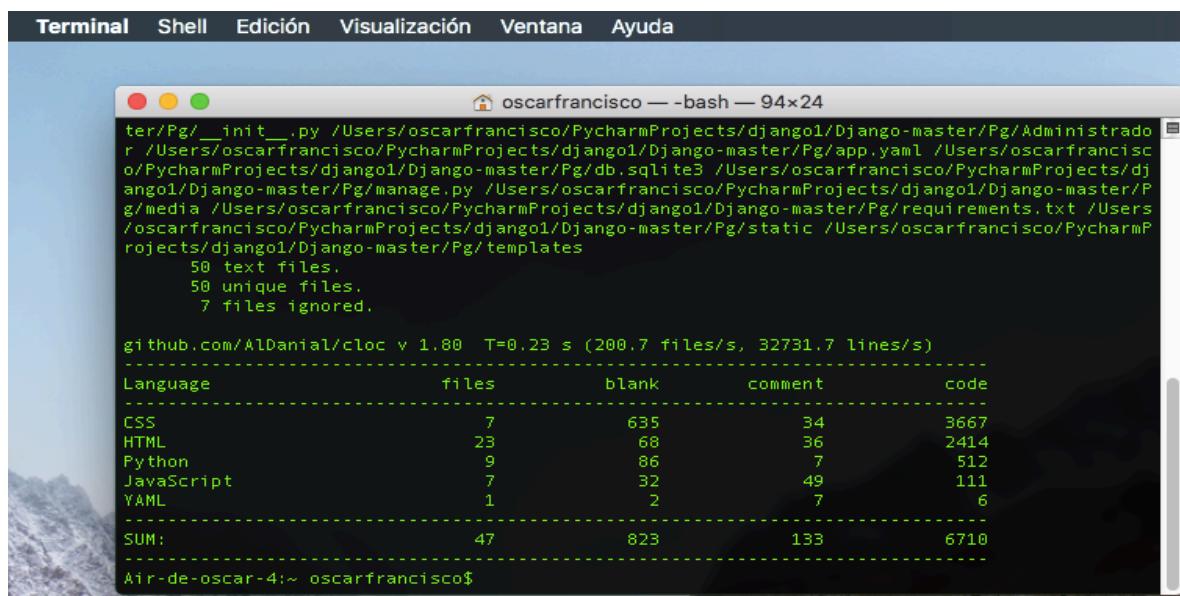
En la anterior tabla se seleccionaron los valores para cada uno de los 15 factores de costos, los cuales se multiplicaran y se obtendrá el valor de AEF.

$$AEF = 1 * 1 * 1.15 * 1 * 1 * 0.87 * 1.15 * 1 * 1 * 1.17 * 1 * 0.95 * 1 * 1 * 1$$

$$AEF = 1.27886411$$

Para saber la cantidad de lineas de código se utilizo CLOC

**FIGURA 59: Cantidad de lineas de código utilizando CLOC**



A screenshot of a Mac OS X terminal window titled "oscarfrancisco — bash — 94x24". The window displays the output of the "cloc" command on a Django project. The output shows the total number of files (50), unique files (50), and ignored files (7). It then provides a detailed breakdown by language:

Language	files	blank	comment	code
CSS	7	635	34	3667
HTML	23	68	36	2414
Python	9	86	7	512
JavaScript	7	32	49	111
YAML	1	2	7	6
SUM:	47	823	133	6710

The terminal prompt at the bottom is "Air-de-oscar-4:~ oscarfrancisco\$".

Fuente: Elaboración propia, 2020

En la contado de lineas de código se evidencio que el proyecto tiene 6710 lineas de código, estas lineas de código se obtuvieron sin contar las lineas comentadas y lineas vacias,

A continuación se deduce que:

$$SLOC = 6710 \text{ KSLOC} = 6.710$$

Por lo tanto encontrados EAF y KSLOC se procede a calcular el esfuerzo.

$$E = EAF * 3.0 * KSLOC^{1.12}$$

$$E = 1.27886411 * 3.0 * (6.710)^{1.12}$$

$$\text{Esfuerzo} = 32.350 \text{ [Personas / Mes]}$$

Obtenido el valor del esfuerzo se procede a calcular el tiempo de desarrollo.

$$T = 2.5 * \text{Esfuerzo}^{0.38}$$

$$T = 2.5 * (32.350)^{0.38} \quad T = 9.36897 \text{ [Meses]}$$

Ahora se procede a calcular el personal promedio requerido para el desarrollo del proyecto.

$$\text{Personas} = 32.350 / 9.36897 \quad \text{Personal} = \text{Esfuerzo} / \text{Tiempo de desarrollo}$$

$$\text{Personal} = 3.4528 \text{ [Personas]}$$

Interpretando los valores, se determina que para el proyecto se han escrito 6710 líneas de código. Lo que se traduce un esfuerzo de 32 personas por mes, el proyecto se ha desarrollado en 9 meses y lo ideal para desarrollar el proyecto es contar con 3 programadores.

Pero la situación en este caso, es que el proyecto ha sido desarrollado por una persona en un lapso de 7 meses, así que los costos de desarrollo se calcularan considerando esta situación.

El desarrollador ha trabajado 5 horas al dia, quien ha realizado las tareas de análisis, diseño y codificación del proyecto. Se ha considerado el trabajo en días laborales de lunes a sábado, cinco horas al dia, lo que supone 100 horas al mes y según el tiempo estimado multiplicado por la cantidad de horas al mes se obtiene un total de horas de trabajo de 936,897

$$\text{Total, horas de trabajo} = 100 * 9.36897 = 936.897$$

Para determinar una aproximación del costo de desarrollo del proyecto se ha averiguado que el sueldo para el área de programación en Bolivia es de 30 bs/hora.

Por lo tanto se procede a calcular el costo estimado del proyecto, multiplicando la cantidad de horas a trabajar en el lapso estimado.

$$\text{Costo del proyecto} = 936.897 * 30 \text{ bs/hora}$$

$$\text{Costo del proyecto} = 28106.91$$

Obteniendo el resultado de la ecuación se determina que el costo estimado del software desarrollado es de 28106.91 bs

### **Costos en base a los requerimientos mínimos**

**TABLA 51: Costos en base a los requerimientos mínimos**

DESCRIPCION	COSTO
<b>Recursos de hardware y software</b>	1964
<b>Desarrollo del proyecto</b>	28106.91
<b>Total bolivianos</b>	30070.91

*Fuente: Elaboración Propia, 2020*

### **Costos en base a los requerimientos ideales**

**TABLA 52: Costos en base a los requerimientos ideales**

DESCRIPCION	COSTO
<b>Recursos de hardware y software</b>	9762
<b>Desarrollo del proyecto</b>	28106.91
<b>Total bolivianos</b>	39772.91

*Fuente: Elaboración Propia, 2020*

A continuación se determina el análisis beneficio / costo, donde los cálculos realizados son en base a un estimado.

### **Costos en base a los requerimientos ideales**

**TABLA 53: Costos en base a los requerimientos ideales**

DESCRIPCION	SIN SISTEMA (AÑO)	CON SISTEMA (MES)
<b>Hojas</b>	45	0
<b>Lapiceros</b>	30	0
<b>Carpetas y cuadernos para registro</b>	400	100
<b>Total</b>	5700	1200

*Fuente: Elaboración Propia, 2020*

Para el cálculo de la estimación Beneficio/Costo se tiene:

Si  $B/C > 1 \rightarrow$  el proyecto es viable económico ya que los beneficios superan a los costos.

Si  $B/C < 1 \rightarrow$  el proyecto no es viable económico ya que los beneficios son superados.

$Y = \text{Beneficio neto}$

n = tiempo de vida de duración del proyecto (5 años es el ciclo de vida de la aplicación ya que se menciono en el capitulo generalidades, donde se determino en la delimitación temporal)

Y = Beneficio donde  $Y = 5700 - 1200 = 4500$

$C_n$  = Cotos donde  $C_n = 1200$

n = Tiempo de vida de duración del proyecto donde n = 5 años

i = Tasa de interes = 2.31

Donde reemplazando en la formula se tienen los valores de

B/C = 1.4583

Después de aplicar la formula se obtuvo por el cual el resultado es mayor a 1 se puede concluir que los beneficios que se obtiene con la implantación del proyecto son justificables respecto al costo invertido para su desarrollo, por lo tanto el proyecto es económicoamente viable.

#### **4.2. VIABILIDAD OPERATIVA**

La plataforma permite al personal de la dirección de posgrado conceder permisos de usuario como administrador, coordinador académico, encargada de sistemas y secretaria.

El sistema es bastante sencillo de entender ya que presenta una interfaz Web muy intuitiva que solo requerirá de conceptos y conocimientos previos y estar familiarizado con el manejo de una PC y navegación en Internet, la mayoría de las ventanas ofrece básicamente botones para facilitar el trabajo de los usuarios que trabajan en la dirección, evitando de esta forma que el usuario pueda cometer errores y por otro lado facilitando la selección de opciones.

También cabe mencionar que se requerirá capacitación y charlas informativas a los diferentes actores para que estos se familiaricen con la plataforma y así puedan trabajar con el mismo, de una forma más eficiente y sin cometer errores.

Las funcionalidades del personal administrativo, en el caso del “Administrador”, es el que tiene más funcionalidades y tiene acceso a todo el sistema. En el caso del “secretaria” solo registra la información de la documentación y carga esta a la

plataforma. En el caso del “coordinador académico”, Tiene las mismas facultades del administrador. En el caso de “encargada de sistemas” solo realiza consultas. El nivel de complejidad en base a las funcionalidades que se realizan en el “uso” del sistema, además de los conocimientos sobre el área de ingeniería de sistemas, se detalla en el siguiente orden:

**TABLA 54: Nivel de complejidad del cargo**

CARGO	COMPLEJIDAD
Administrador	Alto
Coordinador académico	Alto
Encargada de sistemas	Medio
Secretaria	Bajo

*Fuente: Elaboración Propia, 2020*

El Administrador del Sistema tiene un nivel de complejidad alto ya que se encarga de registrar usuarios, asignar cuentas de usuarios y tener acceso a todo el sistema debe estar al tanto de esta información existente y las modificaciones que se realizan a dicha información.

El coordinador académico también tiene las mismas facultades del administrador. La secretaria tiene un nivel de complejidad bajo ya que se encarga de registrar la documentación de docentes y cursantes ademas de cargarla a la plataforma, deberá tener los conocimientos de Ofimática para Windows y saber manejar un navegador con la versión más reciente.

La encargada de sistemas tiene un nivel de complejidad bajo ya que solo puede acceder a consultas de la información requerida.

Teniendo en cuenta los beneficios del sistema, y los resultados que brindan al administrador para la toma de decisiones, se puede decir que el proyecto es operativamente viable.

## **5. CONCLUSIONES Y RECOMENDACIONES**

### **5.1. CONCLUSIONES**

- Se estableció el modelado de negocio actual efectuando el método de observación y entrevistas al personal involucrado para determinar cómo se realiza este proceso el cual fue reflejado con un diagrama de flujo para su mejor entendimiento. Seguidamente de este se realizó la evaluación de las diferentes deficiencias existentes en el proceso actual para elaborar una propuesta de modelado de negocio alternativo solucionando estos problemas identificados que se plasmó en un diagrama de flujo para su mejor entendimiento. Estableciendo el flujo de la información para cada uno de los actores que participan.
- Se selecciono el lenguaje de programación Python que gracias a sus diferentes librerías y líneas de comandos que puede ser usada en aplicaciones gráficas independientes fue posible el desarrollo del workflow el armado y secuencia de los estados y así mismo facilitó ver los estados y continuar un proceso de adopción, se utilizo el framework Django que ayudó en el desarrollo y organizar toda la estructura que facilito las labores del desarrollo del sistema, gestor de base de datos MsQL que ayudó en el almacenamiento de datos del sistema para mostrar los registros, las solicitudes de adopción, reportes, etc, gracias a su capacidad de almacenamiento y a la obtención de los mismos, y la metodología de desarrollo de software los cuales ayudó en el desarrollo del sistema propuesto, en base a la metodología de programación extrema o extreme programing (por su significado en inglés) se organizo el desarrollo del proyecto en entregables en base a historias de usuario y pruebas de aceptación, esta metodología permitió enfocarse mas en la parte de programación que en la documentación como tal
- La implementación del modulo de gestión de usuarios se lo realizo previo a un análisis de los requerimientos de la organización, con la respectiva

- identificación de actores los cuales tienen involucrados al director y al coordinador académico, elaboración de diagramas de caso de uso y plasmado de la codificación y las respectivas pruebas de aceptación tal y como lo establece la aplicación de la metodología de programación extrema.
- La implementación del modulo de admisión de documentación se realizo en base a las actividades dando una explicación detallada de los actores que se involucran en este módulo tales como el director, la secretaria y la encargada de sistemas, posterior a ellos se desarrollaron los diagramas de caso de uso general y por actores, ademas de plasmar las partes mas importantes de la codificación, y las respectivas pruebas de aceptación requeridas por la metodología XP.
  - La implementación del modulo de generación de código único, se la realizo en base a actividades de análisis previas, ademas de una selección del algoritmo adecuado para ser implementado en dicho módulo, posterior a ello se identifico a los actores involucrados con el módulo que este caso es la secretaria, permitiendo registrar los documentos y la plataforma asignándole automáticamente un código único a cada documento registrado en la plataforma.
  - Para la implementación del módulo de inclusión de código QR en la documentación se interactuó con el módulo de generación de código único para poder parametrizar el código QR con dicho código único, se identificaron los actores involucrados, que en este caso es el director y el coordinador académico, con el fin de poder marcar la documentación revisada y dicha documentación cambie de estado de “no revisado” a “revisado”, realizadlo el diagrama de casos de uso general y por actor, ademas de plasmar las partes mas importantes de la codificación, concluyendo finalmente en las pruebas de aceptación necesarias.
  - Para el módulo de generación de reportes se realizo un previo análisis a los requerimientos de la organización plasmadas en historias de usuario para posterior a ello poder satisfacer las mismas, se identifico los actores involucrados que este caso son todos, director, coordinador académico,

secretaria y encargada de sistemas, elaborando el diagrama de casos de uso general y por actores respectivamente, concluyendo con las pruebas de aceptación como lo exige la metodología XP.

- Por ultimo se realizaron las pruebas finales al sistema concluido, haciendo hincapié en las pruebas de integración.

En base a todo lo anteriormente mencionado se desarrollo una plataforma para la gestión documental empleando técnicas criptográficas, empleando tecnología workflow y la arquitectura orientada a servicios, permitiendo otorgar mayor seguridad a los documentos garantizando de gran manera la integridad de los mismos y las características de la arquitectura orientada a servicios otorga a la aplicación poder interactuar con cualquier otra aplicación dentro de la organización y por ende una mayor expansión del proyecto, por ultimo la generación de reportes actualizados, permitirá a los empleados de la dirección de posgrado hacer las tareas de la gestión de documentos, mas simples y mas seguras. m

## **5.2. RECOMENDACIONES**

### **5.2.1. Recomendaciones de funcionamiento**

Las recomendaciones de funcionamiento son:

- Para garantizar la seguridad de las contraseñas se recomienda a los usuarios cambiarlos de forma periodica
- Para que la plataforma funcione de manera aceptable se recomienda que se cumplan con los requerimientos mínimos de software y hardware establecidos en el capitulo 4.
- Para proteger la información se recomienda realizar copias de respaldo para mantener la disponibilidad de la información.

### **5.2.2. Recomendaciones a futuro**

- Implementar el sistema a todas las direcciones de posgrado a nivel nacional
- Integración de la plataforma con otras aplicaciones dentro de la organización, debido a la arquitectura orientada a servicios que posee.
- Implementar sistemas de gestión de pagos enlazado a la plataforma

## BIBLIOGRAFÍA

### REFERENCIA DE LIBROS

- Abad Espinoza, M. (2014). Fundamentos de Ingeniería de Software 1ra Edición. Loja. Pag 17-23
- Alonso A. (2012), Responsive web design; interfaces web adaptables al dispositivo empleando html5 y css3 , Editorial, Universidad de alcalá, primera edición, pag. 5.
- Arfuch, L. (1995). La entrevista, una invención dialógica. Barcelona: Paidos. 1ra Edición. Pag 87-88
- Bernal, C. (2006). Metodología de la Investigación (2da. Edición). México: Ed. Pearson
- Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. Pag. 29-31
- Booch, G. Rumbaugh, J., & Jacobson, I. (2006). El lenguaje unificado de modelado. Madrid: Pearson Educación. S. A. pag 50
- Celma, M., & Casamayor, J. (2003). Bases De Datos Relacionales. Pearson Prentice Hall. Pag 44
- Cordova, F. G. (2004). Recomendaciones metodologicas para el diseño de cuestionarios. Mexico: Limusa. Pag. 102-107
- Deitel, P. J. (2004). Como programar en Java. Pearson Educación. Pag. 63
- Encinoza, L. B. (2008). Sistema de información Ed. Campo Santo. Pag. 221-223
- Gauchat J. (2012), *El gran libro de HTML5, CSS3 y Javascript* (1ra Edición) Barcelona, Ed. Marcombo
- Group, P. (2008). History of PHP and related projects. Pag. 51
- J., D. C. (2001). Introducción a los sistemas de bases de datos. Addison-Wesley Publishing Company, Ed. 7. Pag 73-74
- Johannes A. (2004), *Introducción a la criptografía* (2da Edición). Ed. Springer Verlag, Pag. 124-132
- Kendall y Kendall, K. J. (2011). Análisis y diseño de sistemas 8va Edicion. Mexico: PEARSON EDUCACIÓN. Pag. 5-7

- Pressman, R. S. (2010). Ingeniería del software un enfoque práctico, 7ma edición.  
Mexico. Pag. 45-47-61-70-312-313-319-351-552-353-394
- S, A. (2011). Psicología aplicada a la seguridad informática. Pag 40
- Someville, I. (2005). Ingeniería de Software 7ma Edición. Madrid: Pearson  
Educación S.A. Pag. 23
- Stroustrup, B. (1995). The C++ programming language. India: Pearson Education.  
Pag. 63-64

## REFERENCIAS DE SITIOS WEB

Aguirre, A. (2014). Sistema informático workflow. Recuperado el 24 de abril de 2015, de [http://Prezi:utm\\_campaign=share&utm\\_medium=copy&r=ex0share](http://Prezi:utm_campaign=share&utm_medium=copy&r=ex0share)

Billy C. (2004), Introducción a la arquitectura de software, recuperado 5 de abril de 2019 de: <http://carlosreyynoso.com.ar/archivos/arquitect.pdf>

Castañeda I.(2014), Administracion, recuperado 4 de abril de 2019 de: [http://dis.unal.edu.co/~icasta/ dir\\_doc/admon\\_bd/oracle\\_admon.pdf](http://dis.unal.edu.co/~icasta/ dir_doc/admon_bd/oracle_admon.pdf)

Carvallo S. (2004), Generación de números aleatorios, recuperado 23 de marzo 2020, de [https://www.dm.uba.ar/materias/probabilidades\\_estadistica\\_C/2004/1/PyEC08.pdf](https://www.dm.uba.ar/materias/probabilidades_estadistica_C/2004/1/PyEC08.pdf)

Ferrer J.(2004), Curso completo de html, Recuperado 30 de marzo de 2019 de <http://es.tldp.org/ Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>

Folgueiras P.(2016), *La entrevista*, Recuperado 1 de abril de 2019 de: <http://deposit.ub.edu/dspace/ bitstream/2445/99003/1/entrevista%20pf.pdf>

Fontela C. (2003), *Documentos y pruebas antes del paradigma de objetos*, Recuperado el 24 de marzo de 2020 de [http://materias.fi.uba.ar/7507/content/20101/lecturas/documentacion\\_pruebas.pdf](http://materias.fi.uba.ar/7507/content/20101/lecturas/documentacion_pruebas.pdf)

Fuentes C. 2011, Desarrollo ágil con kanban. Recuperado 11 de febrero 2020 de <http://umh2818.edu.umh.es/wp-content/uploads/sites/884/2016/03/DesÁ>

Gonzales J. 2009, *Inmersión en python 3*, Recuperado el 10 de abril de 2020 de <https://argentinaenpython.com/quiero-aprender-python/inmersion-en-python-3.0.11.pdf>

Hoyo C 2014, El código de barras, Recuperado el 11 de marzo de 2020 de: <http://www.acmor.org.mx/sites/default/files/10asd/asd%20%20barras.pdf>

ISO 27000. (2013). ISO 27001: El estándar de seguridad de la información. Recuperado el 7 de abril de 2019, de <http://www.iso27000.es/sgsi.html>

Marciales, D. (17 de mayo de 2011). Workflow. Recuperado el 24 de abril de 2019, de Blog Sistemas de información <http://sistemasinfoger.blogspot.com/>

Microsoft Technet. (12 de abril de 2013). Informacion General de SQL Server. Recuperado el 7 de enero de 2020, de [https://technet.microsoft.com/es-es/library/ms166352\(v=sql.90\).aspx](https://technet.microsoft.com/es-es/library/ms166352(v=sql.90).aspx)

Microsoft, D. N. (2015). "Arquitectura orientada a servicios (SOA)". Recuperado el 6 de enero de 2020, de <https://msdn.microsoft.com/en-us/library/>

Mollericona, E. (2010). CakePHP. Recuperado el 6 de enero de 2020, de <http://www.edsonmm.com/desarrolla-tus-aplicaciones-web-con-el-framework-cakephp/>

Rodríguez Vargas, M., & Serna Martínez, L. (2019). SEFT - Software Empresarial para Flujos de Trabajo. Recuperado el 24 de enero de 2020, <http://repositorio.utp.edu.co/dspace/>

Slideshare.net. (octubre de 1999). Workflow y Uml. Recuperado el 24 de abril de 2015, de Slideshare.net: <http://es.slideshare.net/guestc13577/workflow-1167198>

Twomey M. 2017, Construyendo Aplicaciones web con flask, recuperado de: <https://buildmedia.readthedocs.org/media/pdf/building-web-applications-with-flask/latest/building-web-applications-with-flask.pdf>, Fecha de consulta: 5 de abril de 2020

Van Rossum G.(2009), Python, Recuperado de: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>, Fecha de consulta 30 de marzo de 2020

Vega A. 2015, Responsive Web Design: Interfaces web adaptables al dispositivo empleado HTML 5 y CSS 3, definición de responsive web design, recuperado 1 de marzo de 2019 de: <https://ebuah.uah.es/dspace/bitstre>

## ANEXOS



## **ANEXO “A”: ENTREVISTA**

### **¿Quien es usted y que cargo desempeña?**

Res. Soy la licenciada Verónica Mollo Condori, encargada de sistemas de la dirección de posgrado

### **¿En su tiempo trabajando en posgrado que deficiencias existen en este departamento?**

Res. Pues aquí tenemos varios problemas en los diferentes procesos que realizamos, te puedo mencionar demasiados pero ahora las deficiencias que más tenemos son con el seguimiento de pagos, algunas veces vienen los cursantes de maestrías o diplomados y quieren saber si tienen algún tipo de pago pendiente, el sistema que tiene la EMI es bastante cuadrado y la responsable de los pagos acá, tiene problemas con ello todo el tiempo. Otra deficiencia es la promoción de algunos diplomados o maestría, aquí los programas se aperturan en base a un punto de equilibrio, este punto de equilibrio no es nada más que tener la cantidad de cursantes suficientes para poder solventar los diferentes gastos que implica aperturar un programa y estos programas al no llegar a demasiadas personas a veces se tienen que cerrarse y hay que reembolsar el dinero a las personas que ya hicieron depósitos bancarios para inscribirse al programa que cerramos. Otro problema con el que batallamos todas las gestiones es con la recepción de documentación de cursantes y docentes, se almacena la información de los docentes en carpetas y se las ordena en los estantes que están en las oficinas, el problema es que cada gestión está aumentando los documentos de docentes y el estante ya no tiene espacio, algunos documentos se pierden porque a veces se sacan los documentos para revisar algún dato y al momento de regresar el documento a su lugar se pierden, porque algún empleado se distrajo o recibió una llamada para una reunión o la guardan en un lugar para devolverlo luego y se olvidan de regresar el documento en fin son varias las cosas que suceden para que se pierdan los documentos o se mezclen y cuando hay inspección de acreditación o del director o se requieren los documentos y están perdidos tenemos muchos problemas internos porque recibimos llamadas de atención o

memorándums y solo nos queda contactar con el docente o cursantes para que nos puedan volver a entregar la documentación pero muchas veces ellos no tienen tiempo o incluso causamos la molestia en ellos

porque consideran que sus documentos no son tratados con la seriedad necesaria, o a veces esa documentación extraviada aparece de nuevo y llegamos a tener documentación duplicada.

### **¿Quien recepciona la documentación?**

Res. La documentación solo debería recibirlo el técnico de trámites que es el Lic. Alejandro Huaylla pero a veces, cuando los docentes o estudiantes vienen a entregar la documentación académica pertinente ya sea para completar su proceso de inscripción a algún programa o contratación de algún docente, no se encuentra el Lic. Huaylla entonces a veces están la secretaría, el coordinador de posgrado, o la encargada de marketing y ellos recepcionan esta documentación, a veces la documentación que reciben esta incompleta pero ellos no se percatan de esto, por lo tanto al momento de hacer el proceso de registrar en planillas de excel los documentos entregados, existen ciertos documentos que faltan, entonces debemos contactarnos con la persona que entregó los documentos y decirle que nos entregue la documentación ausente, pero en algunas ocasiones estas personas ya sean docentes o cursantes aseguran que entregaron toda la documentación y que fuimos nosotros quienes hicimos extraviar esa información por lo que se hace muy complicado que nos entreguen esa documentación faltante, o incluso genera una disconformidad por parte de ellos donde se crea un sentimiento de que su información no es tratada con la seriedad del caso.

### **¿Existe algún tipo de alteración de información?**

Res. Por lo menos en estos dos años en los que trabajo en posgrado no hemos recibido denuncias ni nada semejante a plagio de información.

### **¿Existe o existió robo de los documentos?**

Res. Hasta ahora no recibimos ninguna denuncia de ello pero esto puede suceder ya que si observas la documentación de algunos docentes se encuentra en el estante que está al lado de la puerta de ingreso a las oficinas, quien sabe que cuando las oficinas están llenas atendiendo a muchas personas, alguien puede aprovechar en ese momento y sustraer alguna carpeta, quizás nadie se percataría en el momento, ya con esa información quien sabe se podría alterar esos documentos.

## **ANEXO “B”**

## ESTUDIANTES INSCRITOS A MAESTRIAS MAE

## Planillas de inscritos a maestrias

Fuente: Dirección de Posgrado CBBA 2019

ANEXO “C”

## LISTA DE DOCENTES DE DIPLOMADOS DPPC

## **Lista de docentes de diplomados DPPC**

*Fuente: Dirección de Posgrado CBBA 2019*

## ANEXO "D"

### LISTA DE REQUISITOS DE DOCUMENTOS PARA INSCRIPCION DE CURSANTES

#### **Lista de documentos para inscripción de cursantes**

	REGISTRO: FILES DE INGRESO CURSANTES	Código: GAP-REG-FIC-22
		Aprobación: 25/02/2019
		Versión: 1.0
		Página 1 de 1

REQUISITOS DE INSCRIPCIÓN Y/O TRAMITE DE TITULACIÓN			
NOMBRE DE CURSANTE:			
PROGRAMA:			
Nº	DOCUMENTOS	X	OBSERVACIÓN
1	Formulario de Admisión		
2	Contrato compromiso de pago		
3	Certificado de Nacimiento Original (F. 516 p/c)		
4	Fotocopia simple de Cedula de Identidad		
5	Fotocopia de factura de Luz		
6	Croquis de Domicilio		
7	Curriculum Vitae documentado		
8	4 Fotografías 4x4 fondo azul para Diplomado		
	6 Fotografías 4x4 fondo azul para Maestría		
	6 Fotografías 4x4 fondo azul para Doctorado		
9	Copia Legalizada del Titulo en Provisión Nacional (F. 5)		
10	Copia Legalizada del Titulo de Maestría (Para Doctorado)		
11	Reporte de Notas		
12	Fotocopia de factura de depósito en favor de EMI		
14	Formulario T4, T11 y T17B (Llenado y firmado)		

ENTREGADO

RECIBIDO

*Fuente: Dirección de Posgrado CBBA 2020*