



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

**LABORATORIO WEB PARA ESTIMAR LA
DEMANDA DE CURSOS EN LA ESCUELA
DE INGENIERÍA EN EL PRIMER CICLO
FORMATIVO**

ERWIN ANDRÉS AGÜERO MEZA

Memoria para optar al grado de
Ingeniero Civil Industrial, con Diploma en Ingeniería en Com-
putación

Profesor Supervisor:
JAIME NAVÓN COHEN

Santiago de Chile, Agosto 2017



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

**LABORATORIO WEB PARA ESTIMAR LA
DEMANDA DE CURSOS EN LA ESCUELA
DE INGENIERÍA EN EL PRIMER CICLO
FORMATIVO**

ERWIN ANDRÉS AGÜERO MEZA

Miembros del Comité:

JAIME NAVÓN COHEN

YADRAN ETOROVIC SOLANO

DANIEL OLIVARES QUERO

Memoria para optar al grado de
Ingeniero Civil Industrial, con Diploma en Ingeniería en Com-
putación

Santiago de Chile, Agosto 2017

*A todos aquellos que contribuyeron
en mi paso por la universidad :)*

AGRADECIMIENTOS

Write in a sober style your acknowledgements to those persons that contributed to the development and preparation of your thesis.

ÍNDICE DE CONTENIDOS

AGRADECIMIENTOS	iv
ÍNDICE DE FIGURAS	vii
ÍNDICE DE TABLAS	ix
ABSTRACT	x
RESUMEN	xi
1. Capítulo 01	1
1.1. Motivación	1
1.2. Contexto	2
1.2.1. Estructura Interna de la Pontificia Universidad Católica de Chile . . .	2
1.2.2. Cursos	6
1.2.3. Sistema de Créditos Académicos Transferibles	6
1.2.4. Estructura del Plan de Estudios de Ingeniería UC	7
1.3. Desafíos de información identificados	11
2. Capítulo 02 : Marco Teórico	12
2.1. Introducción	12
2.2. Ingeniería de Software	12
2.2.1. Modelos y Ciclos de Vida del Desarrollo de Software	12
2.2.2. Metodologías de Desarrollo de Software	19
2.2.3. Levantamiento de Requerimientos	26
2.2.4. Metodología y técnicas usadas en este trabajo	35
2.3. Descubrimiento de Conocimiento (Knowledge Discovery)	36
2.3.1. Categorías, metodologías y modelos	36
2.3.2. Metodologías y modelos usadas en este trabajo	42
3. Capítulo 03 : Diseño de la Solución	44

3.1.	Introducción	44
3.2.	Objetivos	44
3.3.	Grandes subsistemas y metodología de trabajo	45
3.3.1.	Subsistemas	45
3.3.2.	Metodología de trabajo	45
3.4.	Requisitos	46
3.4.1.	Funcionales	46
3.4.2.	Relatos de usuarios agrupados por Épicas	48
3.4.3.	Requisitos No Funcionales	72
3.5.	Arquitectura	73
3.5.1.	Diagrama de Contexto	73
3.5.2.	Diagrama de Componentes	74
3.5.3.	Descripción de los Datos	75
4.	Conclusiones	79
REFERENCES		80
ANEXO		83
A.	Glosario	84
B.	An Interesting Short Story	86

ÍNDICE DE FIGURAS

1.1	Modelo T en el primer ciclo formativo	8
1.2	Malla curricular primer ciclo formativo	10
2.1	Fases del Modelo Cascada	13
2.2	Ejemplo de Ciclo de Vida del Modelo Incremental	14
2.3	Sucesión de Incrementos en Timebox de 4 (o 3) pasos	16
2.4	Ejemplo del Ciclo de Vida del Desarrollo Ágil	18
2.5	Uso de Metodologías Ágiles	19
2.6	Elementos presentes en Scrum	20
2.7	Scrum Framework	21
2.8	Flujo de Trabajo en <i>Extreme Programming</i>	26
2.9	Niveles de los Objetivos	30
2.10	Versión de Caso de Uso de una columna	31
2.11	Representación Gráfica de un Caso de Uso	32
2.12	Estructura de un relato de Usuario	34
2.13	Etapas del proceso tradicional de KDD	37
2.14	Niveles de Categorización de Actividades en CRISP-DM	39
2.15	Fases en CRISP-DM	40
2.16	Fases en ASM-DM	42
3.1	Subsistemas y sus objetivos	46

3.2	Diagrama de Contexto	73
3.3	Diagrama de Componentes	74
3.4	Modelo de Base de Datos	75

ÍNDICE DE TABLAS

1.1	Facultades , Institutos, Escuelas y Departamentos en la UC	4
1.1	Facultades , Institutos, Escuelas y Departamentos en la UC	5
1.2	Sistema de conversión de Créditos UC y Chileno	7
1.3	Distribución de créditos SCT-Chile y UC en el primer ciclo formativo	9
2.1	Modelos a lo largo de la historia de <i>KDP</i>	38
3.1	Criterios de fechas para cambios de periodos académicos	78

ABSTRACT

The aim of this project is create a platform which allows to estimate and predict the number of students will take a course in The Engineering School in the next academic period. This need borns for the implementation of the new acamedic plan created in 2013, this program is divided into 2 formative phases: Bachelor of Engineering, during the first 4 years, and follows for an UC Professional Title or another academic degree. The flexibility that allows this structure has as consecuence that one specific course can be founded in the acamedic path of very diverse students, it brings a really high variability in the number of students whom take the course.

In order to manage the courses demand was created a web plataform using the framework Ruby On Rails and R ,the language and environment for statistical computing and graphics. These technologies allow to face in a modular way the challenges to create predictives models and manage the information of them through the web platform. In the requirements gathering process was used the methodology User Stories and for the design of predictive models was choosen Cross Industry Standard Process for Data Mining (CRISP-DM). In general, the web platform manage all the information used by the predictive models and allow load new data, update and/or delete the current information, at the same time, it allows to operate the predictive models for each course and visualize the results obtained.

To conclude, the results of the predictions are presented based on the data provided by the *Dirección de Pregrado*, as well as a list of solutions to the problems that exits for the lack of data given that there is still no generation of students who have completed the first cycle.

RESUMEN

El objetivo del presente trabajo es crear una plataforma que permita estimar y predecir la cantidad de alumnos que tomarán un curso en la Escuela de Ingeniería en el siguiente periodo académico. Esto nace gracias al nuevo plan de estudios implementado por la Escuela de Ingeniería en el 2013, éste se encuentra dividido en 2 ciclos formativos: Licenciatura en Ciencias de la Ingeniería, correspondiente a los primeros 4 años, y articulación con un Título Profesional UC u otros grados académicos. La flexibilidad de esquema tiene como consecuencia que un curso puede estar en el camino académico de alumnos muy diversos, lo que se traduce en alta variabilidad en el número de alumnos que toman el curso.

Para gestionar la demanda de cursos dado los ciclos formativos en la Escuela de Ingeniería se diseñó una plataforma web utilizando el framework Ruby On Rails en conjunto con el entorno estadístico R. Este stack de tecnologías permite abordar de forma modular los desafíos a nivel de información para la creación de modelos predictivos y gestión de información a través de la plataforma web. Se enfrentó el proceso de levantamiento de requerimientos con la metodología de Relatos de Usuario y para el diseño de modelos predictivos se escogió la metodología Cross Industry Standard Process for Data Mining (CRISP-DM). A modo general, la plataforma web permite gestionar la información presente (cargar nuevos datos , actualizar y/o eliminación la información ya existente), gestionar los modelos predictivos para cada curso y visualizar los resultados obtenidos.

Para concluir, se presentan los resultados de las predicciones en base a los datos proporcionados por Dirección de Pregrado, además de un listado de soluciones a los problemas que se presentan por la falta de datos dado que aún no existe una generación de alumnos que haya finalizado el primer ciclo.

1. CAPÍTULO 01

1.1. Motivación

La Escuela de Ingeniería de la Pontificia Universidad Católica desde año 2013 decidió implementar un nuevo plan de estudio dinámico y flexible basado en nuevas áreas de ingeniería e interdisciplina como foco de desarrollo (*Sitio Web Ingeniería - PUC2*, n.d.). Este nuevo currículum académico tiene dos ciclos de formación : Licenciatura en Ciencias de la Ingeniería y Articulación. La duración para el primer ciclo de formación es de 4 años y el segundo va desde 2 a 4 años dependiendo de la modalidad escogida por el estudiante.

La flexibilidad de esta modalidad de estudios se basa en el hecho de que alumno tiene múltiples opciones de formación en ambos ciclos. Dado que el primer ciclo se basa en una estructura T, la cual define un bloque de amplitud y otro de profundidad, el alumno debe seleccionar el conjunto de cursos para definir su especialización, denominado Major, y el conjunto de cursos denominados Minor con el objetivo de completar aún más su especialización o completar sus estudios en una área diferente a la enseñada en su Major. Al haber finalizado este primer ciclo de estudios, el alumno obtiene el grado de licenciado.

La formación del estudiante continúa con el segundo ciclo de formación, el cual posee los siguientes caminos:

- (i) Articulación con un Título Profesional de Ingeniero UC.
- (ii) Articulación con otros títulos profesionales UC. Hoy existen convenios para articular con los títulos de Médico Cirujano, Arquitecto y Diseñador UC.
- (iii) Continuar con un grado académico superior de postgrado (magíster y doctorado). Este puede ser UC o no UC, y requiere de postulación independiente.
- (iv) Emprendimiento o empleo temprano.

Uno de los principales desafíos que genera esta estructura dinámica y flexible es la gestión de recursos necesarios para la programación de los cursos dictados semestre a semestre, dentro de los recursos más valiosos podemos mencionar la gestión de la salas

para el uso de cátedras, laboratorios y ayudantías de los cursos así como también la planta de profesores necesarios para las diferentes secciones de cada curso.

Todo lo anterior genera la necesidad de conocer cuántos alumnos tomarán un determinado curso en el siguiente periodo académico. El presente trabajo busca dar la respuesta a esta problemática a través de la implementación de una plataforma web que permita la recolección de la nueva información generada por el nuevo currículum, la gestión de modelos predictivos y, finalmente, la entrega de resultados para poder tomar decisiones de cómo estructurar algunos de los recursos anteriormente mencionados, en particular, la planta de profesores.

1.2. Contexto

1.2.1. Estructura Interna de la Pontificia Universidad Católica de Chile

Para entender en el contexto el cual se desenvuelve el nuevo plan de estudio de Ingeniería UC, el cual tiene dentro de sus objetivos principales el cultivo de disciplinas interdisciplinarias, se hace necesario entender la estructura académica sobre la cual se rige la Pontificia Universidad Católica de Chile, en adelante UC.

Los siguientes puntos obtenidos del Reglamento sobre la Estructura Académica de la Universidad Católica de Chile (*Reglamento de Estructura Académica*, n.d.) permiten entender las entidades que conforman a la universidad y cuáles de ellas se encuentran autorizadas para proveer grados académicos y títulos profesionales chilenos:

- (i) De acuerdo al punto I, artículo 2, se extrae lo siguiente:
 - (a) La Pontificia Universidad Católica de Chile se estructura sobre la base de **unidades académicas** que son la organización fundamental para la cual la Universidad realiza sus actividades propias.

- (b) Las unidades académicas a través de las cuales la Universidad realiza sus labores de docencia investigación y extensión son las **Facultades**, los **Institutos**, las **Escuelas** y los **Departamentos**.
- (c) Las unidades académicas anteriormente mencionadas, a excepción de los Departamentos, podrán otorgar **grados académicos** y **títulos profesionales**, de conformidad con la reglamentación vigente.
- (ii) De acuerdo al punto I, artículo 4, se enuncia:
- (a) Las Facultades son las unidades académicas principales, organizadas en torno a una o más áreas del saber, con el propósito de coordinar las actividades de docencia, investigación y extensión, y asegurar una adecuada representación ante el Honorable Consejo Superior de la Universidad.
 - (b) Las Facultades pueden estar formadas por uno o más Institutos y Escuelas y dos o más Departamentos.
 - (c) Algunas unidades académicas podrán pertenecer a más de una Facultad, con el objeto de cumplir mejor sus fines y tener una estructura más adecuada y coherente con el trabajo interdisciplinario que realizan.
 - (d) Los Institutos y las Escuelas son unidades académicas de la Universidad dependientes de una Facultad.

(iii) De acuerdo al punto I, artículo 6, se entiende:

- (a) Los Departamentos son unidades académicas de la Universidad, integradas por académicos que desarrollan actividades en torno a una misma disciplina o disciplinas afines del saber. Los Departamentos pueden depender de los Institutos o Escuelas o directamente de la Facultad a la que pertenecen.

Frente a los puntos anteriormente expuestos, podemos establecer el siguiente nivel jerárquico para las unidades académicas relacionadas con la Escuela de Ingeniería:

- (i) La universidad está compuesta por Facultades.
- (ii) Las Facultades están compuestas por Institutos y/o Escuelas.

- (iii) Los Departamentos pueden depender de los Institutos o Escuelas o directamente de la Facultad a la que pertenecen.

El esquema anterior permite representar las unidades académicas existentes y la relación entre ellas. A continuación la tabla 3.1 se muestran las Facultades y las unidades académicas a su cargo.

Tabla 1.1. Facultades , Institutos, Escuelas y Departamentos en la UC

Facultad	Unidad Académica	Unidad Académica	Unidad Académica
Facultad de	-	-	-
Agronomía e			
Ingeniería Forestal			
Facultad de	Escuela de	Escuela de Diseño	Instituto de Estudios
Arquitectura,	Arquitectura		Urbanos
Diseño y Estudios			
Urbanos			
Facultad de Artes	Escuela de Arte	Escuela de Teatro	Instituto de Música
Facultad de Ciencias	-	-	-
Biológicas			
Facultad de Ciencias	Escuela de	Instituto de	-
Económicas y	Administración	Economía	
Administrativas			
Facultad de Ciencias	Escuela de	Instituto de	Escuela de Trabajo
Sociales	Psicología	Sociología	Social
Facultad de	-	-	-
Comunicaciones			
Facultad de Letras	-	-	-

Tabla 1.1. Facultades , Institutos, Escuelas y Departamentos en la UC

Facultad	Unidad Académica	Unidad Académica	Unidad Académica
Facultad de Derecho	-	-	-
Facultad de Educación	-	-	-
Facultad de Filosofía	Instituto de Estética	Instituto de Filosofía	-
Facultad de Física	Instituto de Astrofísica	Instituto de Física	-
Facultad de Historia, Geografía y Ciencia Política	Instituto de Ciencias Políticas	Instituto de Geografía	Instituto de Historia
Facultad de Ingeniería	Escuela de Construcción Civil	Escuela de Ingeniería	-
Facultad de Matemática	-	-	-
Facultad de Medicina	Escuela de Enfermería	Escuela de Medicina	-
Facultad de Química	-	-	-
Facultad de Teología	-	-	-
Campus Villarica	-	-	-

1.2.2. Cursos

Un curso dentro de la universidad consiste en el instrumento académico a través del cual una unidad académica entrega un conjunto de habilidades y conocimientos relacionados a un tema o más temas en particular. Cada curso puede formar parte de diferentes programas de cursos cumpliendo un rol diferente en cada uno. Los roles que puede tener un curso son mínimo, optativo de profundización o electivo según el programa del alumno.

Por otro lado, cada curso exige un nivel de esfuerzo a realizar por el alumno, el cual se encuentra medido por sus créditos, los cuales representan la expresión cuantitativa del trabajo académico efectuado por el alumno, necesaria para alcanzar los objetivos y logros de aprendizaje del curso o actividad curricular (*Créditos UC*, n.d.). Este trabajo incluye clases teóricas o de cátedra, actividades prácticas , de laboratorio o taller, actividades clínicas o de terreno, estudio personal y evaluaciones. Un crédito UC equivale a una hora de trabajo por semana. (*Créditos UC*, n.d.)

1.2.3. Sistema de Créditos Académicos Transferibles

El plan de estudios de Ingeniería UC posee la particular de ser compatible con mallas curriculares a nivel internacional de manera de impulsar la movilidad de hacia postgrados locales e internacionales (*Sitio Web Ingeniería - PUC2*, n.d.).

Para alcanzar este objetivo, dentro de los programas de cursos ya mencionados anteriormente ,Major y Minor, el trabajo académico se expresa en una unidad académica denominada crédito SCT-Chile, la cual se rige bajo El Sistema de Créditos Transferibles, denominado SCT-Chile (*Sistema de Créditos Transferible Chile*, n.d.), el cual posee el mismo objetivo de créditos UC, poder cuantificar de forma racional el trabajo académico que un alumno debe dedicar a sus estudios en un año o semestre.

A continuación se muestra la tabla de equivalencia entre créditos SCT-Chile y créditos UC:

Tabla 1.2. Sistema de conversión de Créditos UC y Chileno

Sistema de créditos académicos transferibles SCT-Chile	Sistema de Créditos UC
30 créditos SCT-Chile al semestre	50 Créditos UC al semestre
Un año de estudios a tiempo completo equivale a 60 créditos SCT-Chile	Un año de estudios completo equivale a 100 créditos, lo cual es equivalente a 1800 horas de trabajo académico del estudiante

Fuente: <http://admissionyregistros.uc.cl/alumnos/cursos/creditos-de-un-curso>

De la tabla anterior 1.2, podemos extraer 1 crédito SCT-Chile equivale a 1.67 créditos UC.

1.2.4. Estructura del Plan de Estudios de Ingeniería UC

A continuación se entregan los detalles sobre la estructura que posee cada ciclo de formación, Licenciatura en Ciencias de la Ingeniería y Continuidad con un Título Profesional UC y/o articulación con otros grados académicos, así como empleo temprano y el emprendimiento, en el plan de estudios mencionado en los párrafos anteriores.

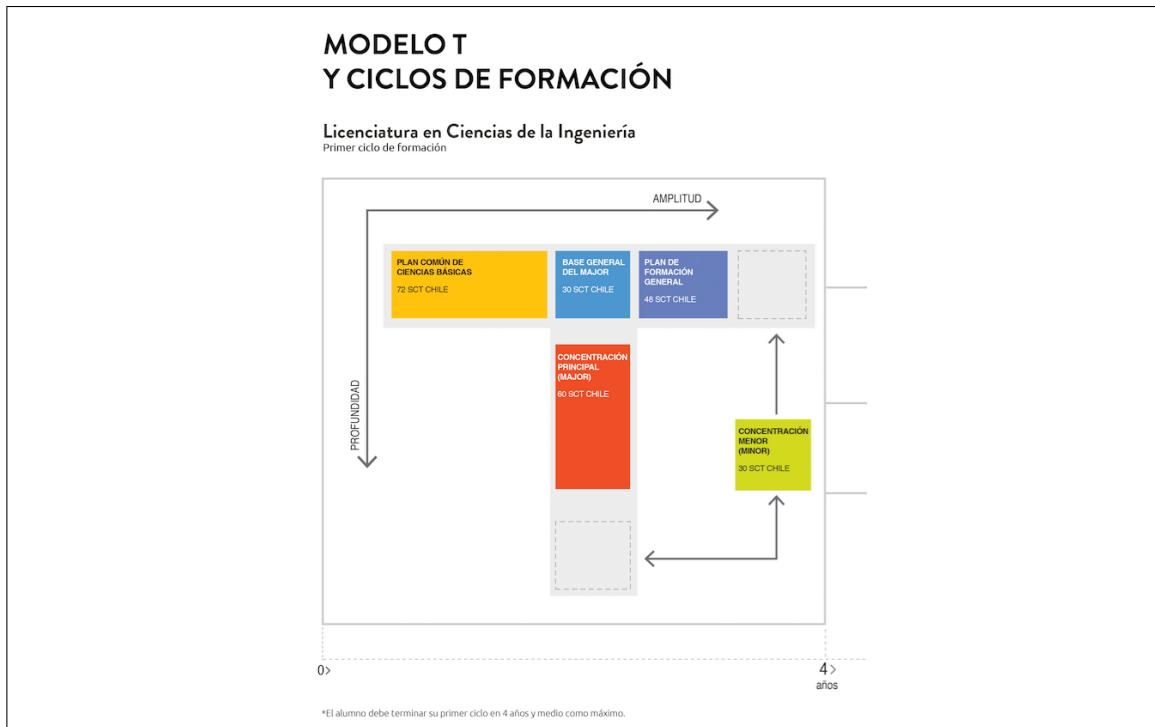
1.2.4.1. Licenciatura en Ciencias de la Ingeniería

Este ciclo formativo posee una duración de 4 años y posee los siguientes componentes (*Guía Alumno Ingeniería UC 2017*, n.d.):

- (i) **Plan Común de Ciencias Básicas:** El objetivo de esta área es consolidar las bases en física, biología, química y matemática.
- (ii) **Base General del Major:** Cimienta las bases para desarrollar las ciencias de la ingeniería.

- (iii) **Plan de Formación General:** Cursos en disciplinas diferentes.
- (iv) **Concentración Principal (Major):** Programa que entrega profundidad en un área disciplinaria o interdisciplinaria.
- (v) **Concentración Menor (Minor):** Programa que permite ampliar o profundizar en un área disciplinaria o interdisciplinaria dependiendo del área escogida en el major.

La estructuración de los 5 elementos mencionados se basa en un modelo T definiendo una amplitud gracias al Plan Común de Ciencias Básicas, Base General del Major y Plan de Formación General, y otra de profundidad entregada por la Concentración Principal (Major). Este modelo T puede verse reflejado en la figura 1.1:



Fuente: (Guía Alumno Ingeniería UC 2017, n.d.)

Figura 1.1. Modelo T en el primer ciclo formativo

De la figura 1.1 cabe destacar que el Minor complementa el aspecto profundidad o amplitud dependiendo del área del Major escogido.

Dado el objetivo de compatibilizar con mallas curriculares internacionales, los ciclos formativos miden su exigencia académica en créditos SCT-Chile, los cuales puede ser transformados a créditos UC gracias a la equivalencia entregada por la universidad. La distribución de carga académica por componente tanto en créditos SCT-Chile como créditos UC se distribuye en la siguiente tabla:

Tabla 1.3. Distribución de créditos SCT-Chile y UC en el primer ciclo formativo

Componente	Créditos SCT-Chile	Créditos-UC
Plan Común de Ciencias Básicas	72	120
Base General del Major	48	80
Plan de Formación General	48	80
Concentración Principal (Major)	60	180
Concentración Menor (Minor)	30	50

Esta estructura define la malla curricular del primer ciclo formativo, la cual puede ser visualizada en la ilustración 1.2.

La diversidad en la formación que puede escoger un alumno se encuentra en la conjunto Major-Minor que escoge, actualmente la Escuela de Ingeniería ofrece un total de 21 Majors, de los cuales 14 son disciplinarios y 7 interdisciplinarios, al mismo tiempo, entrega una oferta de 57 Minors entre los cuales, 32 son profundidad y 25 de amplitud. En el Anexo I se detalla los Majors y Minors disponibles.

Finalmente, con el objetivo de orientar a los alumnos hacia una elección de Major/Minor basada en sus intereses, se les solicita que hagan declaraciones de Major al término del segundo semestre y tercero para terminar con elección de Major y Minor al finalizar su cuarto semestre.

Licenciatura en Ciencias de la Ingeniería							
Primer ciclo de formación							
SEMESTRE 1	SEMESTRE 2	SEMESTRE 3	SEMESTRE 4	SEMESTRE 5	SEMESTRE 6	SEMESTRE 7	SEMESTRE 8
MAT1610 CÁLCULO I	MAT1620 CÁLCULO II	MAT1630 CÁLCULO III	EYP1113 PROBABILIDADES Y ESTADÍSTICA	MAJOR	MAJOR	MAJOR	CAPSTONE MAJOR
QIM100A QUÍMICA GENERAL II	FIS1513/ICE1513 ESTÁTICA Y DINÁMICA Y FIS0151 LABORATORIO DE ESTÁTICA Y DINÁMICA	FIS1523/IQM1003 TERMODINÁMICA Y FIS0152 LABORATORIO DE TERMODINÁMICA	FIS1533 ELECTRICIDAD Y MAGNETISMO Y FIS0153 LABORATORIO DE ELECTRICIDAD Y MAGNETISMO	MAJOR	MAJOR	MAJOR	MAJOR
MAT1203 ÁLGEBRA LINEAL	ICS1513 INTRODUCCIÓN A LA ECONOMÍA	MAT1640 Ecuaciones Diferenciales	OPTATIVO BIOLÓGICO (E) o OTRO SIMILAR DEL ÁREA BIOLÓGICA (D)	ING2030 INVESTIGACIÓN, INNOVACIÓN Y EMPRENDIMIENTO	MINOR	MAJOR	MINOR
ING1004 DESAFIOS DE LA INGENIERIA	IIC1103 INTRODUCCIÓN A LA PROGRAMACIÓN	OPTATIVO DE EXPLORACIÓN DE MAJORES (B)	OPTATIVO DE FUNDAMENTOS DE CIENCIAS O INGENIERIA (E)	MINOR	MINOR	MINOR	MINOR
LET0003 DESARROLLO DE HAB. COMUNICATIVAS PARA INGENIEROS (A) o FIL168 ÉTICA PARA INGENIEROS (A)	FIL168 ÉTICA PARA INGENIEROS (A) o LET0003 DESARROLLO DE HAB. COMUNICATIVAS PARA INGENIEROS (A)	OFG	OFG	OFG	OFG	OFG	OFG
PLAN COMÚN DE CIENCIAS BÁSICAS		72 SCT CHILE	REQUISITOS ADICIONALES: -ING1110 Taller de Hábitos y Estrategias de Estudio -VRA100C Examen de Castellano -VRA3010 English Test (Sufficiency ALTE 3) -ING1001 Práctica 1 (F) -Examen de competencias fundamentales				
BASE GENERAL DEL MAJOR		30 SCT CHILE	LICENCIATURA EN CIENCIAS DE LA INGENIERÍA 240 SCT CHILE				
FORMACIÓN GENERAL		48 SCT CHILE					
MAJOR		60 SCT CHILE					
MINOR		30 SCT CHILE					

Fuente: (*Guía Alumno Ingeniería UC 2017*, n.d.)

Figura 1.2. Malla curricular primer ciclo formativo

1.2.4.2. Continuidad - Articulación Posterior a la Licenciatura

El segundo ciclo formativo se inicia una vez obtenida la Licenciatura en Ciencias de la Ingeniería, el cual entrega los siguientes caminos a seguir:

- (i) Continuidad a un título profesional de Ingeniería Civil UC.
- (ii) Continuidad a otros títulos profesionales UC. Hoy existen convenios para continuar estudios a los títulos de Médico Cirujano, Arquitecto y Diseñador UC.
- (iii) Articulación al título profesional Ingeniería UC y al Magíster en Ciencias de la Ingeniería o el Doctorado en Ciencias de la Ingeniería UC.
- (iv) Realizar un grado académico superior de postgrado (magíster o doctorado) UC u otra.
- (v) Salida al Mercado Laboral: Emprendimiento o Empleo Temprano.

1.3. Desafíos de información identificados

El plan de estudios descrito recientemente fue implementado el año 2013, esto trae consigo una serie de cambios tanto en la comunidad de estudiantes por los nuevos conceptos establecidos como a nivel administrativo para la planificación de cursos para los diferentes ciclos y profesores para los cursos impartidos en cada uno. Esta situación se vuelve particularmente crítica en particular por varios cambios realizados en la planificación inicial de cada Major y/o Minor, así como también en las Articulaciones establecidas entre la Escuela de Ingeniería y otras unidades académicas como Diseño y Medicina. Tanto los cambios como la propia estructura flexible levantan un desafío a nivel de información en cómo realizar un seguimiento adecuado del plan de estudio que realiza cada alumno, para poder disponer de la mejor manera los recursos disponibles por la Escuela.

Las necesidades de información que se detectan para realizar un seguimiento adecuado del plan de estudios de cada alumnos pueden ser descritas en las siguientes categorías:

- (i) Registro de cursos tomados por los alumnos en cada periodo académico.
- (ii) Registro de las declaraciones de Major realizadas por alumnos por periodo académico.
- (iii) Gestión de demanda de cursos para cursos, Major y Minor por parte de los alumnos de las diferentes generaciones del nuevo plan.

2. CAPÍTULO 02 : MARCO TEÓRICO

2.1. Introducción

El presente trabajo se basa en conocimiento de Ingeniería de Software (*Software Engineering*) y el proceso denominado Descubrimiento del Conocimiento (*Knowledge Discovery*). En el presente capítulo se exponen las definiciones de ambos campos, así como también los conceptos de cada área del conocimiento en Ciencias de la Computación.

2.2. Ingeniería de Software

De acuerdo a la *Association for Computer Machinery* (*Association for Computing Machinery*, n.d.), de ahora en adelante A.C.M., Ingeniería de Software o *Software Engineering*, en adelante S.E., se preocupa de la construcción y mantención de software que posee las siguientes características:

- (i) Comportamiento confiable y eficiente.
- (ii) Existe el modo de poder seguir desarrollando sobre ellos y al mismo tiempo mantenerlos.
- (iii) Satisfacen todos los requerimientos definidos por las necesidades de los clientes.

Por su parte, *Institute of Electrical and Electronics Engineers*, de ahora en adelante I.E.E.E. (P. & R., 2004) define S.E. como la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software, es decir, la aplicación de ingeniería al software.

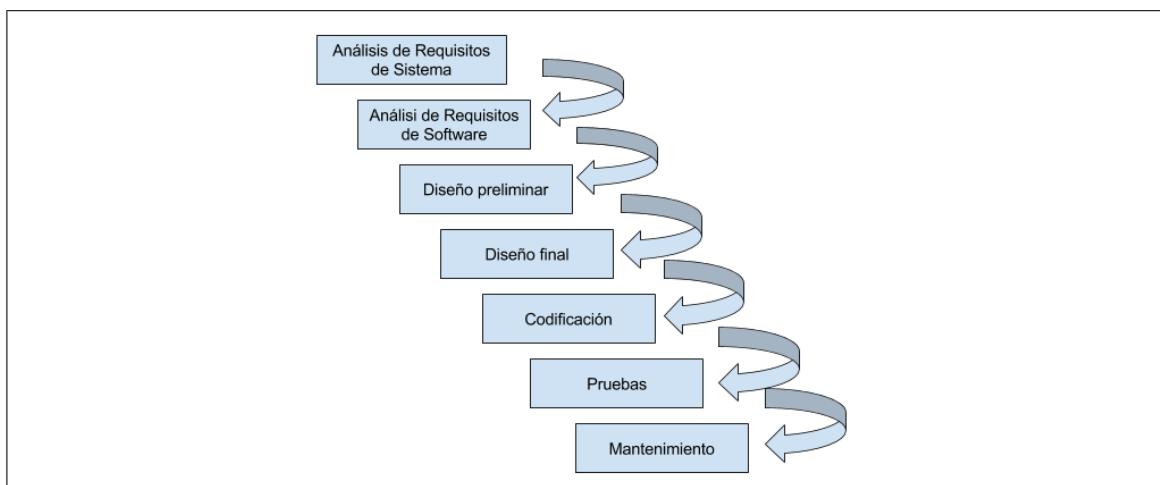
2.2.1. Modelos y Ciclos de Vida del Desarrollo de Software

A lo largo de la historia de S.E. se ha creado una diversidad de modelos o ciclos de vida para la creación y mantención de software. Cada modelo define la forma o estrategia en que las distintas actividades del proceso deben ser llevadas a cabo para el desarrollo

y mantenimiento del producto de software. A continuación se analizan los enfoques más descritos en la literatura actual.

2.2.1.1. Modelo Cascada

Este enfoque define una secuencia lineal de pasos para la creación de un software. Las etapas suelen incluir análisis de requisitos del sistema, análisis de requisitos de software, diseño preliminar, diseño, codificación, pruebas y mantenimiento. La siguiente ilustración muestra las etapas mencionadas anteriormente.



Fuente: Fuente Propia
Figura 2.1. Fases del Modelo Cascada

El modelo de cascada se caracteriza por sus fases secuenciales, las cuales deben terminar de modo completo antes de iniciar una nueva. Adicionalmente, posee un fuerte enfoque en el levantamiento de requerimientos antes de iniciar cualquier actividad de código o de implementación del software, por lo cual se caracteriza por una intensa comunicación con los clientes o stakeholders en las fases iniciales, para luego dar protagonismo a las fases de codificación, pruebas y mantenimiento.

Ventajas

- (i) Simple, permite una clara visualización de avance del proyecto.

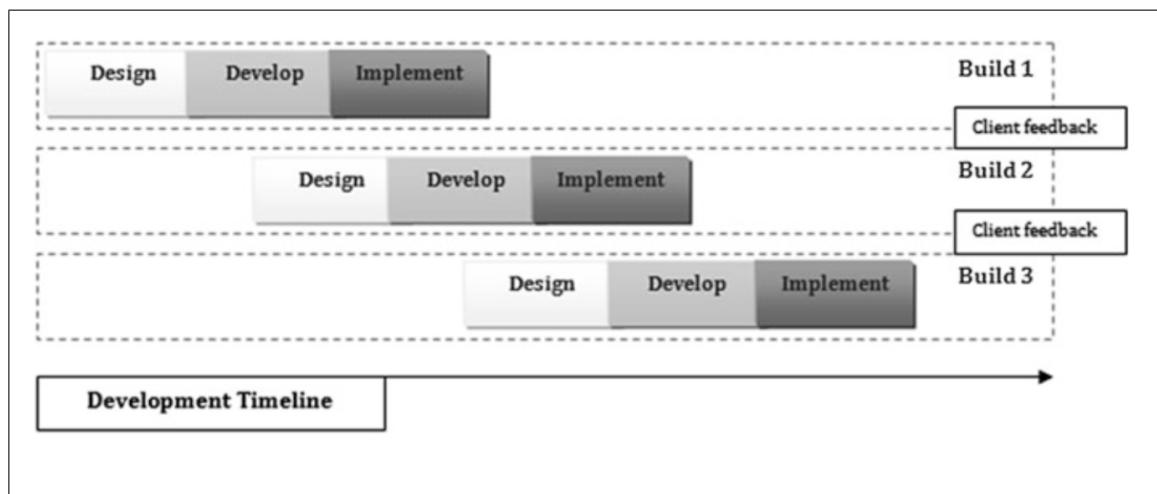
- (ii) Se genera buena documentación que permite una clara planificación de los eventos dentro del proyecto.
- (iii) Etapa de codificación sólo se enfrenta después de tener un diseño robusto.

Desventajas

- (i) No es flexible frente a los cambios y deseos del cliente.
- (ii) Los errores sólo pueden ser detectados en etapas tardías de implementación.
- (iii) Cambios en los requerimientos agregan altísimos costos adicionales.

2.2.1.2. Modelo Incremental

Este modelo consiste en una evolución del anterior mejorando la flexibilidad y al mismo tiempo disminuyendo el tiempo de interacción con los stakeholders a lo largo de todo el proyecto. Con estos objetivos en mente, en vez de crear un proceso con etapas aisladas, se crea un proyecto que puede ser diseñado, desarrollado e implementado en etapas sucesivas, incorporando al mismo tiempo la retroalimentación del cliente en etapas intermedias (Massey & Satao, 2012). El siguiente esquema ejemplifica la filosofía de este nuevo enfoque:



Fuente: (Isaias P., 2015, p. 25)
Figura 2.2. Ejemplo de Ciclo de Vida del Modelo Incremental

Ventajas

- (i) Incorpora flexibilidad al permitir incorporar necesidades que no fueron previstas o consideradas en la etapa inicial del proyecto.
- (ii) Permite el desarrollo progresivo del proyecto hasta que está completo validando con los stakeholders durante todo el proyecto y no cuando éste haya finalizado.

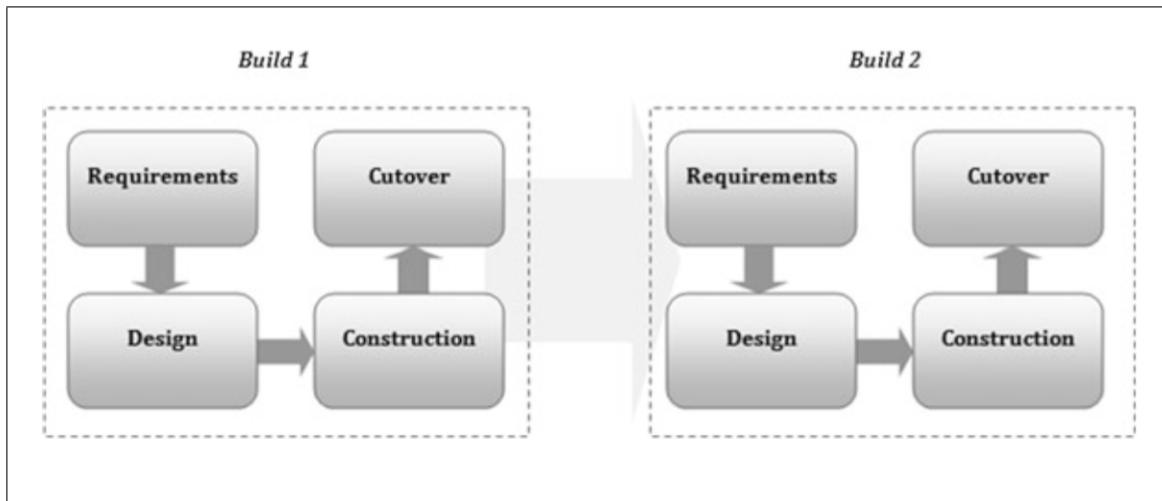
Desventajas

- (i) La flexibilidad permitida por el proyecto puede hacer el proyecto más costoso debido a los constantes cambios que pueden realizarse.
- (ii) Al incorporar nuevos elementos, debido a las etapas incrementales, puede tener problemas de incompatibilidad con versiones anteriores del software.

2.2.1.3. Rapid Application Development (RAD)

Esta filosofía de desarrollo se basa en la idea de que los métodos desarrollados hasta los inicios de los 90 simplemente eran muy rígidos, en consecuencia no se podían aplicar de forma eficiente cuando las fechas de entrega cobran importancia. La ideología se basa en la entrega rápida de software mientras se mantiene una alta calidad en la implementación desarrollada. Esta metodología fue desarrollada y formalizada por James Martin en el mismo horizonte de tiempo.

Este proceso de creación posee dos modalidades, la primera consiste en 4 fases secuenciales de tiempo : planificación de requerimientos, diseño desde el punto de vista del usuario, construcción y transición a la siguiente fase. En la segunda modalidad se fusionan las fases de planificación de requerimientos y diseño en una sola de análisis. Todas las fases de desarrollo en ambas modalidades se desarrolla en un espacio de tiempo determinado denominado timebox, donde se priorizan las actividades en función de las necesidades del negocio sin importar el desafío técnico al cual se ven enfrentadas (Gottesdiener, 1995). A continuación se presenta un posible esquema de trabajo bajo esta metodología:



Fuente: (Isaias P., 2015, p. 30)

Figura 2.3. Sucesión de Incrementos en Timebox de 4 (o 3) pasos

Ventajas

- (i) Al enfocarse en prioridades de negocio sin importar dificultades técnicas permite entregar una implementación que satisface las necesidades más críticas de los *stakeholders*.
- (ii) Al tener un tiempo específico de desarrollo se mantiene un paso a producción marcado por hitos.
- (iii) Debido a que los espacios de tiempo de desarrollo entre bloques no son necesariamente del mismo horizonte temporal es posible adaptarlos a las nuevas necesidades que vayan surgiendo en el transcurso del proyecto.
- (iv) Posee una comunicación constante para el desarrollo de cada fase.

Desventajas

- (i) Sólo puede ser usada cuando las personas que lo desarrollan tienen un alto nivel técnico y una gran capacidad de trabajo en equipo, debido a que los tiempos de desarrollo necesitan ser lo más preciso posible por cada bloque.

- (ii) Es posible que se creen expectativas falsas por parte de las personas que gestionan el proyecto al ser trabajado bajo espacios específicos de tiempo, lo que podría generar discusiones internas con el equipo de desarrollo.

2.2.1.4. Enfoque Ágil

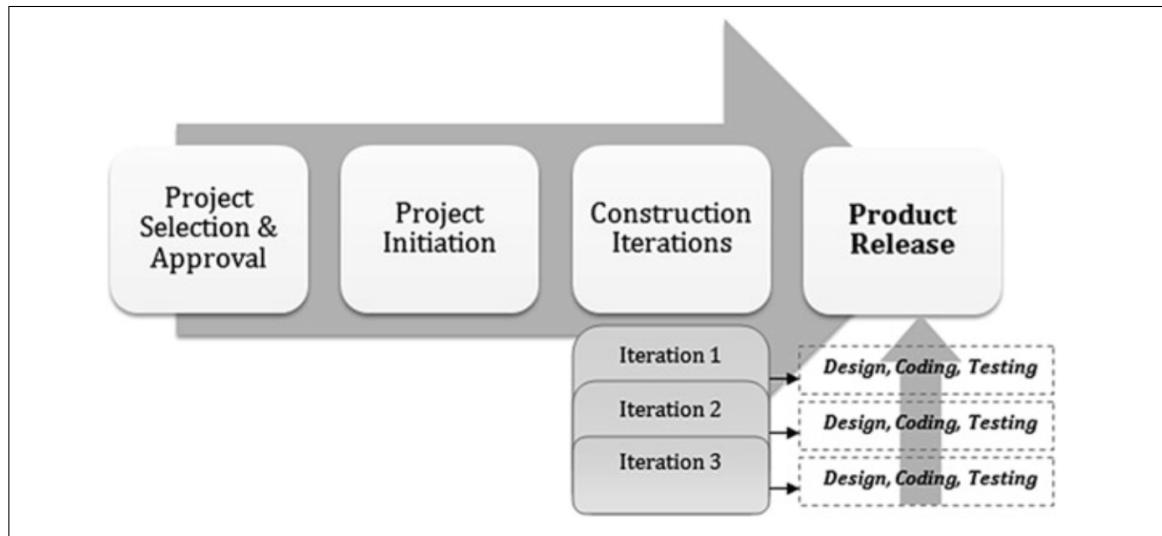
Esta modalidad de desarrollo nace específicamente en el año 2001 con la publicación del Manifiesto Ágil, en donde se da a conocer un *framework* de cómo construir un software en un ambiente de constantes cambios y al mismo tiempo mantener un alto nivel de calidad de implementación y satisfacción al cliente. Existen 12 principios declarados en este manifiesto, los cuales pueden ser resumidos en las siguientes declaraciones (Isaias P., 2015):

- (i) La satisfacción del cliente es la principal prioridad.
- (ii) Cambios en los requerimientos son bienvenidos, no son más un obstáculo.
- (iii) El *software* es entregado regularmente en constantes *release*.
- (iv) Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- (v) Individuos motivados son una pieza vital para el éxito del proyecto.
- (vi) Conversaciones cara a cara es esencial para una exitosa colaboración.
- (vii) El *software* funcionando es la medida principal de progreso.
- (viii) El desarrollo sostenible debe ser alentado.
- (ix) Enfocado en bases técnicas y diseño de calidad.
- (x) Simplicidad debe ser favorecida.
- (xi) La mejor forma de gestionar un proyecto es con equipos auto organizados.
- (xii) Debe haber constantes discusiones para el mejoramiento del equipo.

En términos generales es posible establecer 4 etapas para el desarrollo bajo este enfoque (Isaias P., 2015):

- 1. Selección y aprobación del proyecto:** Los stakeholders (administradores, desarrolladores, clientes, entre otros) definen de modo conjunto el alcance, propósito y requerimientos del producto o implementación.
- 2. Iniciación del proyecto:** El equipo de trabajo es construido en base a los requerimientos anteriormente establecidos. Adicionalmente, se establece la infraestructura de trabajo y se procede a la instalación de las herramientas tecnológicas a ocupar. En general, se establecen espacios de tiempo de trabajo para el equipo y reuniones de acuerdo a las necesidades del proyecto.
- 3. Construcción basada en iteraciones:** Las iteraciones consisten en dos etapas, planificación y construcción. La idea es que al final de cada iteración se obtenga una pieza de software terminada.
- 4. Entrega de producto maduro (*product release*):** Esta fase final está caracterizada nuevamente por dos etapas, pruebas y mejoramiento así como también de la documentación necesaria para el entendimiento del proyecto.

El esquema 2.4 muestra el proceso genérico bajo cualquier proceso con enfoque ágil:



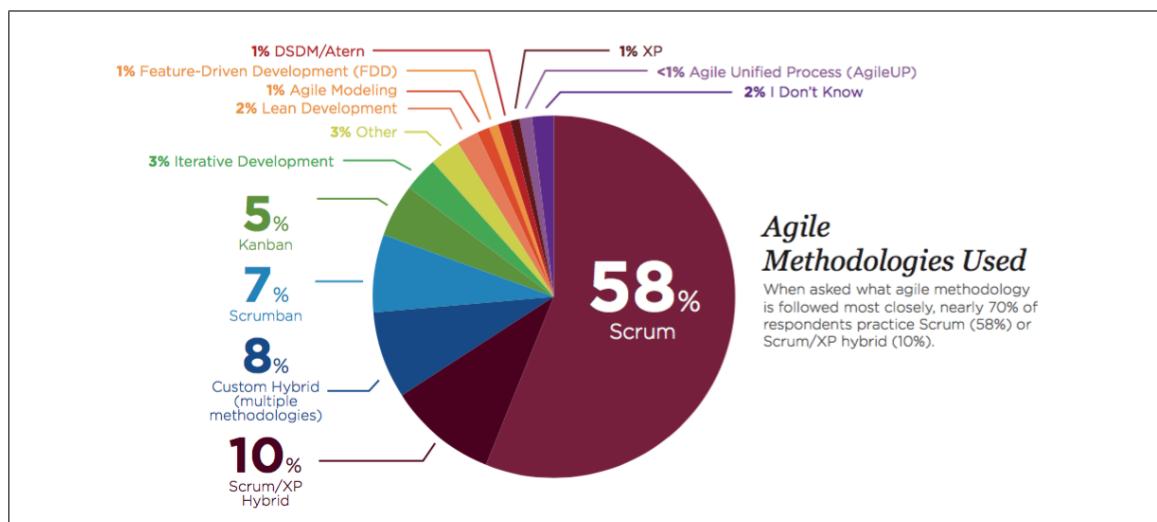
Fuente: (Isaias P., 2015, p. 32)

Figura 2.4. Ejemplo del Ciclo de Vida del Desarrollo Ágil

2.2.2. Metodologías de Desarrollo de Software

Un modelo o ciclo de vida puede ser considerado como una estructura de referencia, una estrategia general para enfrentar el desarrollo, pero no prescribe el modo concreto de llevarlo a cabo. La forma específica de cómo llevar a cabo el proyecto la define una metodología orientada por un determinado modelo o enfoque.

Dentro de los modelos mencionados anteriormente el más relevante hoy en día es enfoque ágil debido a los cambios que ha sufrido la industria tanto a nivel tecnológico como social. Desde su nacimiento en el año 2001 por la declaración del Manifiesto Ágil han surgidos diversas metodologías, las cuales enfrentan diferentes desafíos impuestos por los supuestos de esta modalidad de desarrollo, tales como incorporación de cambios en cualquier momento del proyecto. A continuación se describen dos de las metodologías más populares que permiten implementar el enfoque ágil, éstas son Scrum y Extreme programming (XP). La figura 2.5 muestra la dominancia de Scrum entre las metodologías ágiles:



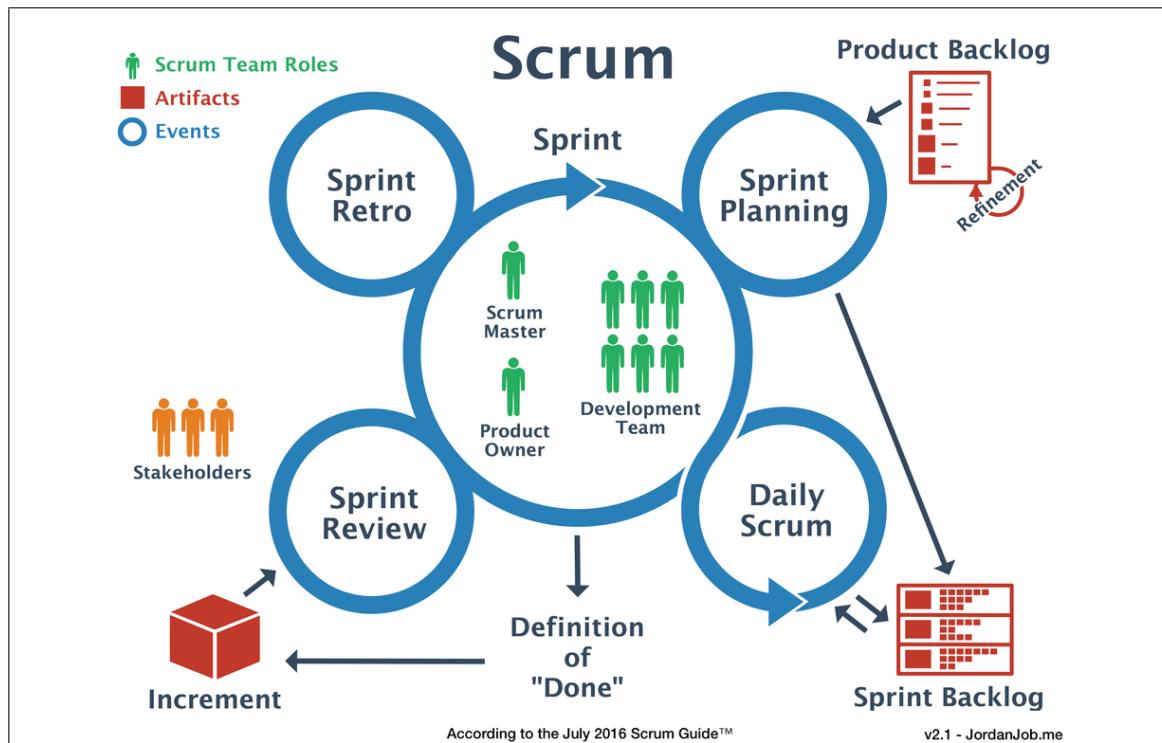
Fuente: (One, 2016, p. 9)
Figura 2.5. Uso de Metodologías Ágiles

Como se aprecia sólo Scrum cubre más del 50% de la industria. Si bien XP representa sólo un 1%, es la base de Scrum/XP Hybrid que es usado por 10% de la industria y es muy

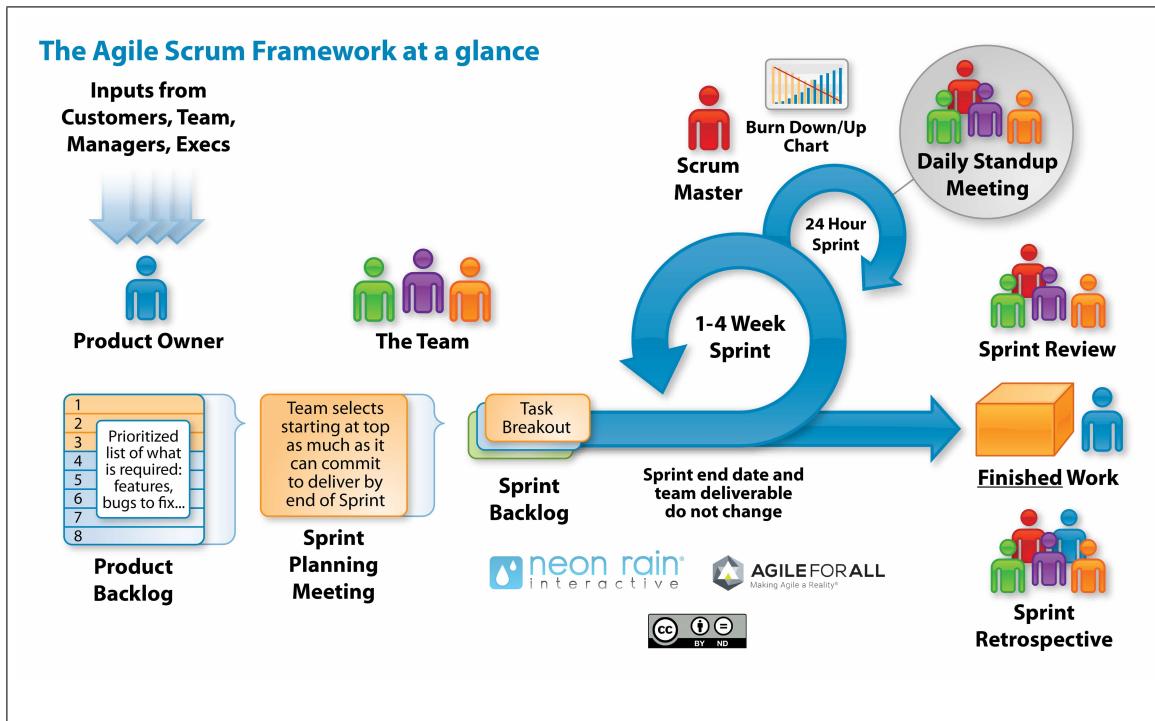
usada en grupos pequeños de desarrollo. Por lo tanto, en el desarrollo de este trabajo se consideraron tanto Scrum como XP.

2.2.2.1. Scrum

Esta metodología se basa principalmente en cómo un equipo debe funcionar y las relaciones que los miembros de éste establecen. Para lograr este objetivo se definen roles, eventos, artefactos y reglas. Cada componente dentro de este marco de referencia sirve para un propósito específico que son esenciales para éxito de su aplicación (Schwaber K., 2013). A continuación se ilustra la forma de trabajo en esta metodología en dos aspectos, primero la figura 2.6 se identifican los elementos presentes en la metodología y a continuación en la figura 2.7 se muestra cómo estos interactúan en el transcurso de un proyecto.



Fuente: (Scrum Diagram, n.d.)
Figura 2.6. Elementos presentes en Scrum



Fuente: (*Scrum Framework*, n.d.)

Figura 2.7. Scrum Framework

Los roles identificados en este metodología son:

- 1. Stakeholder:** Corresponde a todas aquellas personas que se involucran en el proyecto, desde los usuarios finales hasta quienes financian el desarrollo del proyecto. Se identifica a todos aquellos que poseen algún interés una vez que el proyecto esté finalizado.
- 2. Product Owner:** Es el encargo de entender el contexto en el cual se desenvuelve el proyecto y en base a eso llevar a cabo el levantamiento de requerimientos del proyecto.
- 3. Scrum Master:** Es el responsable de verificar que la metodología Scrum sea aplicada de forma correcta, así como también el encargado de proporcionar a cada miembro del equipo los elementos que necesite para llevar a cabo su tarea. Este rol debe estar al servicio tanto

del *Product Owner* como para los miembros del equipo de desarrollo.

- 4. Development Team:** Son los responsables de la creación del proyecto en base a los requerimientos entregados por el *Product Owner*.

Por su parte, la definición de los artefactos en esta metodología son:

- 1. Product Backlog:** Es una lista ordenada de todas las necesidades del producto/proyecto y es la única fuente de requerimientos existente. El responsable de su escritura es el *Product Owner*, quien debe procurar por su contenido, disponibilidad y orden. Se debe indicar que nunca está completo y permanece en frecuente cambio frente a los diferentes requerimientos que el proyecto pueda ir necesitando. A modo general este artefacto posee las funcionalidades, requerimientos, mejoras y reparaciones para las próximas versiones del producto. Los elementos presentes poseen nombre o título, atributos que permiten describirlo, orden, estimación y valor para el proyecto (Schwaber K., 2013).
- 2. Sprint Backlog:** Es el conjunto de ítems seleccionados del Product Backlog que conforman el siguiente entregable del producto/proyecto que se está desarrollando. Se puede interpretar como la proyección de lo que será el software en su siguiente fase, donde todos los elementos presentes en este conjunto se considerarán terminados.
- 3. Increment:** Es la suma de todos los elementos desarrollados en un Sprint Backlog. Este debe considerarse terminado y listo para ser usado en la próxima versión del software.

Finalmente, se definen eventos dentro de Scrum con el objetivo de mantener una comunicación regular y eliminar la necesidad de encuentros. Cada evento posee una duración

limitada y que puede variar según el proyecto así como también del equipo quien implemente la metodología, sin embargo, existen un periodo de tiempo recomendado en cada caso.

Los eventos presentes en Scrum son (Schwaber K., 2013):

1. Sprint: Se considera como el corazón de Scrum debido a que es el periodo de tiempo base en el cual se entrega un incremento (incremento) del software. La duración recomendada para un sprint es un mes o semanas y es ideal que mantenga la misma duración durante todo el proyecto. Dentro de este periodo de tiempo ocurre el sprint planning, daily scrum, sprint review y, finalmente, sprint retrospective.

2. Sprint Planning: Periodo de tiempo en el cual se planifica el trabajo y orden del mismo a realizarse en durante el sprint. El tiempo máximo para un sprint planning es de 8 horas cuando el tiempo del sprint es de un mes. El scrum master es el responsable de la gestión y la ejecución de esta fase. Por otro lado, se entiende que este periodo tiene por objetivo responder ¿qué elementos formarán parte del próximo incremento? y, al mismo tiempo, ¿qué trabajo se realizará para cumplir el requerimiento descrito en el Product Backlog?

3. Daily Scrum: Consiste en una reunión de 15 minutos de duración donde el equipo de desarrollo sincroniza y crea el plan de desarrollo para las próximas 24 horas. Este trabajo se hace en base a la última reunión realizada. Adicionalmente, los miembros del equipo explican qué hicieron en el último daily scrum, qué harán en el actual y de forma conjunta identifican e informan los problemas que han ocurrido y los que podrían ocurrir para el desarrollo del siguiente daily scrum.

- 4. Sprint Review:** Es una reunión informal con máxima duración de 4 horas donde los miembros del equipo de desarrollo identifican los logros obtenidos al final del sprint. Es una instancia en donde es posible modificar el Product Backlog. En esta reunión deben participar stakeholders invitados por el Product Owner y todos los miembros de la metodología. Los principales objetivos de este intervalo de tiempo son:
- (i) Qué se hizo y qué no hizo.
 - (ii) Se explican en detalle los problemas ocurridos y cómo fueron resueltos.
 - (iii) Se discute de modo general qué elementos son importantes en base al avance logrado para el siguiente sprint planning.
 - (iv) Se discute el estado del Product Backlog y se modifica en base al incremento logrado, a los cambios provenientes de los stakeholders así como también de los eventos externos del proyecto.

5. Sprint Retrospective:

Es un encuentro donde el equipo de desarrollo analiza su modo de trabajo y crea un plan de mejoras para ser atacadas en el próximo sprint. Se inspeccionan todos los ámbitos dentro de un equipo los miembros involucrados en el proyecto, relaciones entre ellos, procesos y herramientas utilizadas. Esta reunión se recomienda una duración máxima de 3 horas y se realiza al finalizar el sprint.

2.2.2.2. Extreme Programming

Esta metodología basa su funcionamiento en 2 principios principales: simplicidad y eficiencia. Con el objetivo de lograr estos principios se definen los siguientes 4 valores o elementos, los cuales poseen sus propias características:

(i) Comunicación (Communication)

Enfocada en lograr la cooperación y buena relación entre los miembros del equipo, para lograr esto se propicia la participación constante del cliente, programación a pares y poseer dentro del equipo de desarrollo una estandarización de código.

(ii) Diseñar de modo simple (Simplicity)

Orienta a que los elementos que se crean sean simples y basado en el pensamiento asociativo usando el mundo real como referencia, es decir, el uso de metáforas. Esto incluye las mejoras que se realizan a través de *refactoring* (Anexo A - 1).

(iii) Retroalimentación constante (Feedback)

Consiste en definir los modos para asegurarse que el software es aceptado tanto por el equipo de desarrollo como los clientes. Para ello necesita de testing continuo, integración continua (recomendado al menos una vez al día) y ,finalmente, basar las entregas en pequeñas funcionalidades.

(iv) Aceptación de nuevos cambios (Courage)

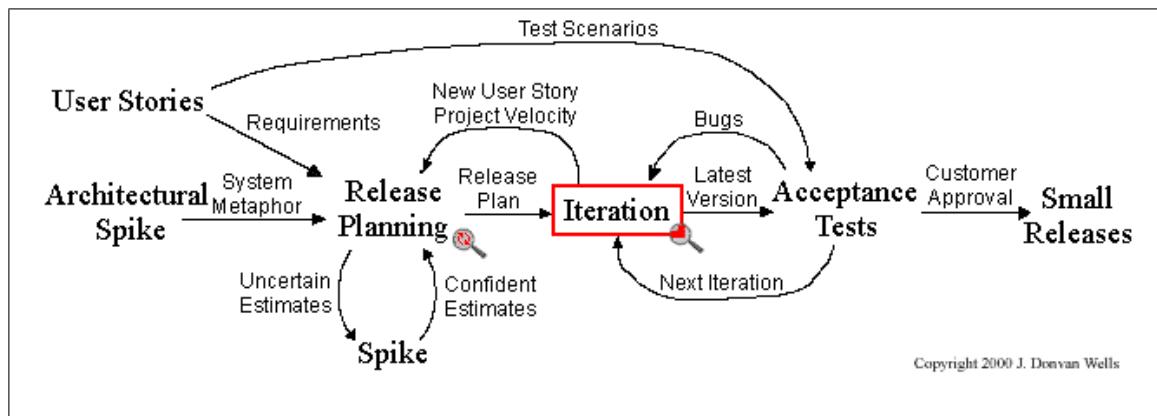
Consta de todas las acciones usadas para la organización del proyecto y del equipo, para ello motiva realizar el levantamiento de requerimientos en base a relatos de usuario para el proyecto y a nivel de equipo estimula el concepto de propiedad de código colectiva.

Los tiempos de desarrollo en XP se basan en iteraciones de 3 semanas en las cuales se recomienda seguir las siguientes 12 prácticas de software:

- (i) Planificación del proceso : En esta instancia es donde se recomienda usar relatos de usuarios para la descripción de las funcionalidades del software.
- (ii) Entregables pequeños.
- (iii) Entregables pequeños.
- (iv) *Test Driven Development* (Anexo A - 2).
- (v) *Refactoring* (Anexo A - 1).

- (vi) Mantener el diseño simple.
- (vii) Programación a Parea (Anexo A - 3).
- (viii) Propiedad colectiva del código.
- (ix) Uso de estándares de programación.
- (x) Integración continua (Anexo A - 4).
- (xi) Mantener al cliente cercano.
- (xii) Mantener un buen clima laboral.
- (xiii) Utilización de metáforas para el diseño : Consiste en crear un lenguaje común para el equipo de desarrollo y clientes con el objetivo de que todos puedan comprender los conceptos y definiciones usadas en el proyecto del mismo modo.

La metodología es poco prescriptiva (en comparación a Scrum) pero existen prácticas sugeridas para organizar el flujo de trabajo. Por ejemplo, el sitio ExtremProgramming.Org sugiere el esquema presentado en la figura 2.8.



Fuente: (*Extreme Programming Project*, n.d.)
Figura 2.8. Flujo de Trabajo en *Extreme Programming*

2.2.3. Levantamiento de Requerimientos

De acuerdo al Glosario Estándar de Ingeniería de Software de la IEEE (IEEE, 1990), el concepto de requisito puede entenderse de acuerdo a alguna de las siguientes definiciones:

- (i) Condición o capacidad necesaria por un usuario para solucionar un problema o alcanzar un objetivo.
- (ii) Condición o capacidad que debe ser encontrada o poseer un sistema o algún componente de éste que satisfaga un contrato, un estándar, especificación u otra formalidad impuesta por documentación.

Al mismo tiempo, de acuerdo a la guía PMBOK (Institute, 2013), identifica 3 categorías:

(i) Requerimientos del Negocio

Describen las necesidades de alto nivel de la organización como un todo, tales como, oportunidades o problemas del negocio, y los motivos por los cuales un proyecto ha sido iniciado.

(ii) Requerimientos de los Stakeholders

Describen las necesidades de uno o varios *stakeholders*.

(iii) Requerimientos de la Solución

Describen las funcionalidades, funciones o características de un producto, servicio o resultado que resuelve los requerimientos del negocio y de los stakeholder.

Dentro de los requerimientos de solución es posible encontrar dos tipos:

(a) **Requerimientos funcionales:** Describen el comportamiento del producto o desarrollo. Por ejemplo: procesos, información, base de datos e interacciones del usuario y producto.

(b) **Requerimientos no funcionales:** Entrega los criterios medibles a través de los cuales un sistema puede comprobar su funcionamiento. Las principales áreas en las cuales se definen estos criterios son rendimiento, nivel de seguridad, escalabilidad, entre otros.

Por lo descrito anteriormente, las metodologías permiten la captura de requisitos de diferentes maneras, las más usuales para los requerimientos del negocio y *stakeholders* corresponden a *focus groups* (Anexo A - 5), reuniones y encuestas. De modo transversal a las metodologías, este procedimiento de entendimiento donde se desarrolla el producto o

software finaliza con un objetivo principal y una serie de objetivos secundarios que deben cumplirse al término de éste. Esta dinámica en las metodologías mencionadas anteriormente se realiza en una etapa específica o designa un responsable para ello, por ejemplo, en Scrum el responsable de este procedimiento durante todo el proyecto es el *Product Owner*. Adicional a esto, al momento de especificar los requerimientos de solución, debido a que los requisitos funcionales describen comportamiento, el cual dentro de lo posible debe ser interpretado del mismo por todo los miembros del equipo, presentan la necesidad de realizarse bajo una estructura o esquema de descripción específico. Esta situación no sucede con los requerimientos no funcionales, debido a que ellos presentan métricas y sus criterios para ser evaluadas , por ejemplo, el tiempo de carga al sitio inicial del *software* no debe ser mayor a 1 segundo.

Dentro de las técnicas de descripción o **levantamiento de requisitos funcionales**, se encuentran Casos de Usos (*Use Cases*) y Relatos de Usuarios (*User Stories*).

2.2.3.1. Casos de Uso

De Acuerdo a Cockburn (Cockburn, 2000), un caso de uso se define como la descripción del comportamiento del sistema bajo varias condiciones que responden a la necesidad de uno o varios *stakeholders*. Adicionalmente, se espera que un caso de uso sea fácil de leer, para una completa identificación de las necesidades se propone las siguientes partes para una descripción completa:

(i) **Nombre (Name)**

Correspondiente al nombre de la acción que se desea realizar.

(ii) **Objetivo en Contexto (Goal in Context)**

Menciona las motivaciones del actor principal para la realización de sus acciones, así como también la descripción del entorno en el cual las realiza.

(iii) **Alcance del diseño (Design Scope)**

Corresponde a la definición de que se desarrolla y no desarrolla dentro del

proyecto. Con el objetivo de describir el alcance dentro de un contexto Cockburn define los siguientes niveles o tipos de alcance:

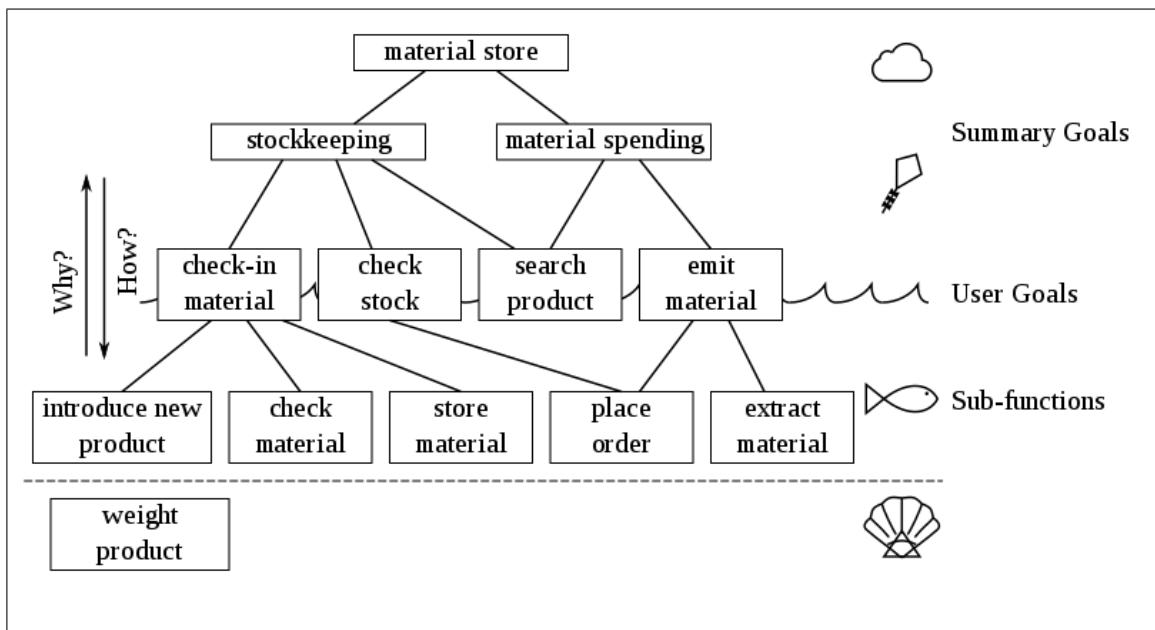
- (a) **Corporativo (Corporative)**: Indica que la discusión para definir el alcance envuelve al comportamiento de toda la organización. Por ejemplo, cuando se debe entregar un desarrollo a un agente externo al contexto del proyecto.
- (b) **Sistema (System)**: Corresponde sólo a una parte del producto o software que se está construyendo.
- (c) **Sub sistema (Sub system)**: Si bien Cockburn indica que a un alcance de sistema debería bastar para la descripción de un software. En caso de que se requiera dividir una pieza de software en varias partes recomienda ocupar la categoría de sub sistema para ella.

(iv) Niveles de los objetivos

Corresponde al nivel de especificidad que se declara en un objetivo. A continuación se enuncian los niveles existentes:

- (a) **Very High Summary** Indica la razón principal de las acciones del usuario principal.
- (b) **Summary** Describe un global más global que puede abarcar diferentes visiones provenientes de diferentes actores principales.
- (c) **User-goal level** Corresponde a una descripción amplia de lo que se desea lograr. Esta descripción o deseo se hace a nivel del actor principal del caso de uso, por lo tanto, indica un nivel de agrupación de objetivos secundarios identificados como *subfunction*.
- (d) **Subfunction** Corresponde al conjunto de objetivos obligatorios que deben cumplirse poder alcanzar otros objetivos de nivel superior. A este nivel un objetivo secundario ya es posible identificarlo como una acción clara del usuario.
- (e) **Too Low** Corresponde a un nivel de objetivos más abajo que subfunction y agrupan todas las tareas más básicas que deben realizarse lograr el objetivo planteado.

El esquema 2.9 permite visualizar de forma gráfica un ejemplo de los niveles mencionados.



Fuente: (Cockburnstyle Use Cases, n.d.)

Figura 2.9. Niveles de los Objetivos

- (f) **Actor Principal (Principal Actor):** Indica al usuario que realiza o ejecuta el conjunto de acciones indicadas por el caso de usuario. Si bien Cockburn menciona en esta versión de caso de uso sólo un actor dentro del caso de uso, es posible incluir otros como actores secundarios.
- (g) **Stakeholders e intereses (Stakeholders and Interests):** Se atribuye al conjunto de stakeholder y los objetivos que cada uno busca.
- (h) **Precondiciones (Preconditions):** Corresponden al conjunto de elementos que se deben cumplir antes de iniciar el flujo de etapas del caso de uso.
- (i) **Estado de éxito (Success End Condition):** Muestra el estado en el cual el sistema se encuentra de la una ejecución exitosa del caso de uso.
- (j) **Estado de error (Failed End Protection):** Describe el estado del sistema en caso de no alcanzar el objetivo.
- (k) **Trigger:** Corresponde a la acción que desencadena el caso de uso.

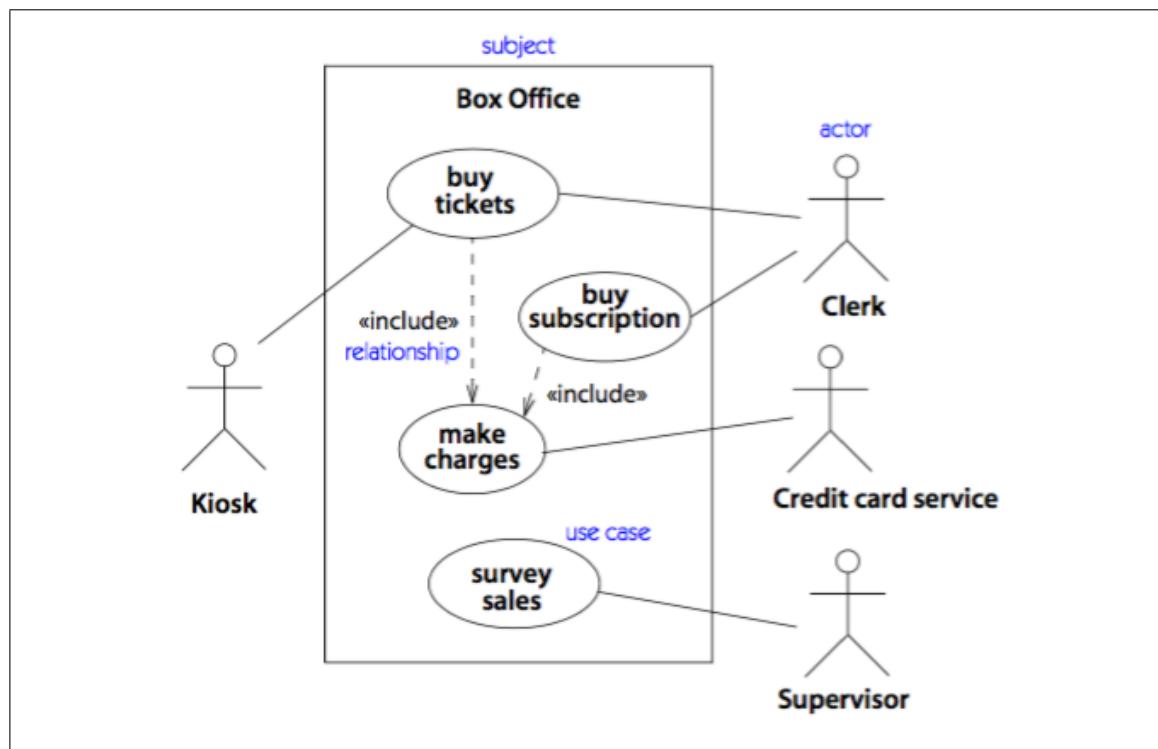
- (l) **Descripción (Description):** Conjunto de pasos a seguir para alcanzar el objetivo del caso de uso.
- (m) **Extensiones (Extensions):** Corresponde a un flujo adicional que puede ser el caso de uso al momento de darse un condición especial sobre el sistema o los actores involucrados.
- (n) **Variaciones (Variants):** Indica la ejecución de una misma acción de un modo distinto dentro del sistema , por ejemplo, al momento de pagar puede realizarse a través de dinero en efectivo, transferencia bancaria, entre otros.

La lista anteriormente descrita puede verse resumida en el recuadro 2.10 conocido por Cockburn (Cockburn, 2000) como la versión completamente vestida (fully dressed).

USE CASE #	<the name is the goal as a short active verb phrase>	
Context of use	<a longer statement of the context of use if needed>	
Scope	<what system is being considered black box under design>	
Level	<one of : Summary, Primary Task, Subfunction>	
Primary actor	<a role name for the primary actor, or description>	
Stakeholder & Interests	Stakeholder	Interest
	<stakeholder name>	<put here the interest of the stakeholder>
	<stakeholder name>	<put here the interest of the stakeholder>
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Protection	<the proper state of the world if goal abandoned>	
Trigger	<the action upon the system that starts the use case>	
DESCRIPTION	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>
	2	<...>
	3	
EXTENSIONS	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub-use case>
VARIATIONS		Branching Action
	1	<list of variation s>

Fuente: (Cockburn, 2000, p. 66)
 Figura 2.10. Versión de Caso de Uso de una columna

Adicionalmente a lo descrito anteriormente, con el objetivo de tener una mejor visualización de la interacción entre casos de usos y actores principales como secundarios el *Unified Modeling Language (U.M.L.)* (Rumbaugh J., 2004) entrega los estándares y elementos para realizar la diagramación de un caso de uso. Este tipo de diagrama se denomina use case view o vista de un caso de uso. Los elementos presentes en esta visualización se aprecian en la representación gráfica de un caso de uso (figura 2.11).



Fuente: (Rumbaugh J., 2004, p. 78)

Figura 2.11. Representación Gráfica de un Caso de Uso

De este modo, para lograr la descripción de un caso de uso, se necesita tanto su diagrama descrito en UML como su detalle, el cual puede estar basado en la estructura propuesta por Cockburn o una estructura similar, pero más simple.

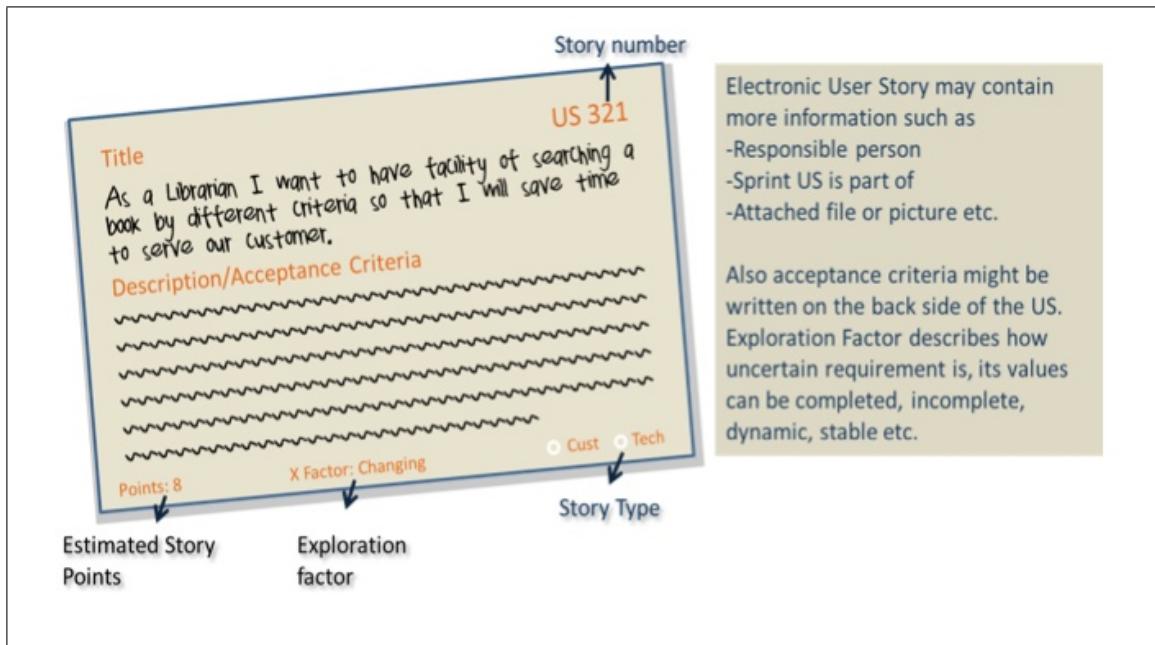
2.2.3.2. Relatos de Usuarios

Según Mike Cohn (M., 2004), un relato de usuario describe una funcionalidad que es valorada por un usuario o otro sistema que se alimenta de ella. Las partes de un relato consisten en la identificador, descripción de la funcionalidad, los criterios de aceptación, prioridad para el negocio y el puntaje de dificultad identificado por el equipo desarrollador. Adicional a lo anterior, cada relato de usuario debe cumplir las siguientes propiedades:

- (i) **Independiente:** Un relato de usuario es independiente de los demás.
- (ii) **Negociable:** No son contratos , por lo que puede ser discutida y descrita por todos los *stakeholders* y los miembros del equipo desarrollador.
- (iii) **Valorada por los compradores y por los usuarios:** Indica que las características del software descrito debe agregar valor tanto para las personas que lo ocupan como aquellas que “pagan” por él.
- (iv) **Estimable:** Es importante poder asignarle un tamaño al relato de usuario para identificar la cantidad de tiempo que tomará. Se indica que un relato que no es posible estimarlo es demasiado grande y es recomendable subdividirlo en varios.
- (v) **Pequeña:** Indica que la descripción de un relato de un usuario no puede ser tan general porque en caso de ser así no podrá ser estimada de forma adecuada. En caso de suceder esto, se recomienda agruparlas bajo un epic (Anexo A - 6) y que la funcionalidad sea dividida en relatos de usuarios más pequeños y con mayor cantidad de detalle.
- (vi) **Comprobable:** Indica que debe existir una forma concreta y no general de que el relato de usuario se desarrolló de forma exitosa.

Con el objetivo de que el relato de usuario pueda cumplir con las propiedades descritas anteriormente, Cohen recomienda utilizar el formato de tarjeta para su descripción. La ilustración 2.12 ilustra esta propuesta.

- (i) **Identificador:** Compuesto por un número o nomenclatura con el formato US-XX, donde XX corresponde al número del relato.



Fuente: (*User Story Format*, n.d.)

Figura 2.12. Estructura de un relato de Usuario

- (ii) **Título:** Describe la funcionalidad que se debe desarrollar. El título debe poseer el siguiente formato

As a <type of user>, I want <some goal> so that <some reason>.

Cohen (M., 2004) indica que la parte <some reason> en muchos casos es opcional.

- (iii) **Puntaje:** Correspondiente al grado de dificultad del relato. Se espera que un relato de 2X tome el doble de tiempo que uno de X puntos.

- (iv) **Prioridad:** Indica el orden en el cual deben ser desarrolladas los relatos de usuarios.

- (v) **Criterios de Aceptación:** Corresponde a las condiciones que debe cumplir el relato de usuario. No se especifica un formato para escribir un criterio de aceptación, pero se recomienda que indique de forma precisa lo que se espera del software, por ejemplo, el sistema debe solicitar el nombre, apellido y correo electrónico para poder identificar a un usuario.

Otro punto importante que destaca Cohn es el uso de *epics* (Anexo A - 6), él indica que no existe un criterio universal para clasificar una historia de usuario como grande, pero sí deben buscar cumplir del mejor modo las propiedades anteriormente descritas. Frente a la necesidad de agrupar, dos o más relatos de usuarios bajo un mismo concepto es cuando se hace el uso de un epic, el cual posee simplemente un nombre relativo a la funcionalidad que describe, una descripción para dar más contexto a quien lee la documentación del proyecto y un código identificador para realizar mejor el agrupamiento de los relatos de usuarios descritos anteriormente (M., 2004).

2.2.4. Metodología y técnicas usadas en este trabajo

Dentro de las metodologías y técnicas para levantamiento de requisitos descritas anteriormente para este proyecto se escogió la metodología XP y levantamiento de requisitos funcionales vía relatos de usuario. Se consideró que para la implementación de Scrum se hacía necesario incorporar un Product Owner, y un Scrum Master además de los miembros de equipo de desarrollo. XP es menos prescriptivo y permite organizar el trabajo en función de la experiencia del equipo de desarrollo, siempre y cuando, se respeten los principios que ésta estipula. Otro factor que se tuvo en cuenta es que dada la considerable complejidad del problema, lo que buscábamos es más bien producir un primer prototipo o lo que se conoce como un *Minimal Viable Product*, en adelante MVP (Anexo A - 7). XP gracias a su principio de simplicidad (Simplicity) permite abordar de mejor manera este escenario.

Por otro parte, a nivel de levantamiento de requisitos si bien tanto Casos de Usos como Relatos pueden ser aplicados con metodologías ágiles, se escogió relatos de usuario debido a su naturaleza menos estructurada que privilegia la comprensión de los requisitos en base a conversaciones con los stakeholder. Adicionalmente a lo anterior, las propiedades de los relatos de usuarios (independiente, negociable, valorado por los usuarios o compradores, estimable, pequeña y comprobable) permite un mejor registro de los esfuerzos invertidos en el desarrollo del prototipo.

Finalmente, el hecho de que el formato de los relatos de usuarios marque explícitamente el objetivo que se buscar resolver permite una mejor visualización general de las soluciones implementadas y al mismo tiempo permite llevar un control efectivo del avance del proyecto de modo más sencillo en base a cada uno de los artefactos requeridos por éste.

2.3. Descubrimiento de Conocimiento (Knowledge Discovery)

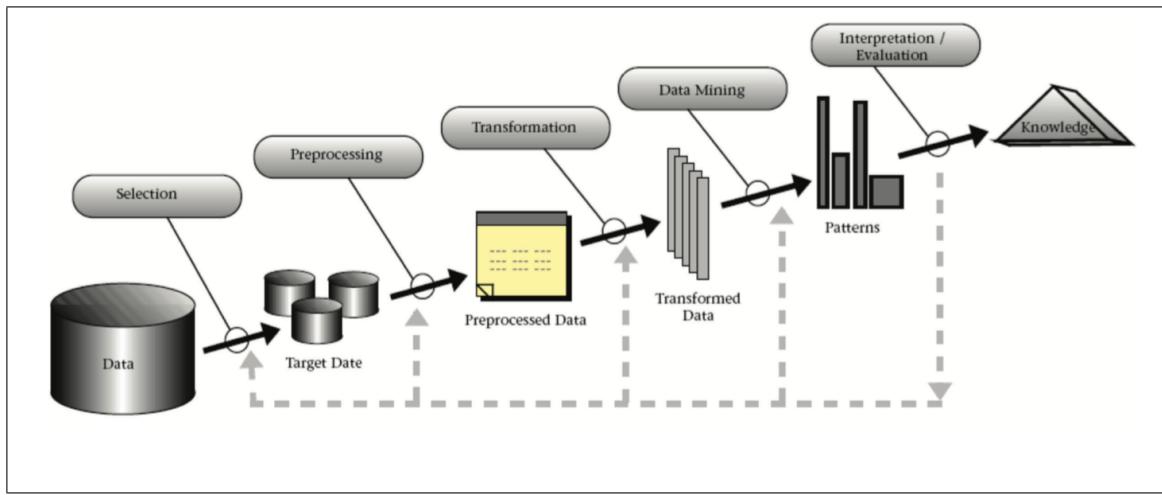
Fayyad (Mouhib Alnoukari, 2012) en 1996 define término Knowledge Discovery como el conjunto de conocimientos relativos al proceso de extracción, almacenamiento y accesibilidad de datos; el uso de algoritmos ,en forma eficiente, para el análisis de los mismos; y, finalmente, la interpretación y visualización de resultados en base a estos, este resultado es conocido como *knowledge*(conocimiento).

2.3.1. Categorías, metodologías y modelos

Desde la creación de este término hasta hoy en día han surgido varias metodologías para llevar a cabo el proceso anteriormente descrito, el cual se denomina *Knowledge Discovery Process (KDP)*. Se logran lograr identificar un total de 4 (Mouhib Alnoukari, 2012):

(i) *Traditional KDP Approach*

Agrupa aquellas metodologías que ocupan un proceso similar al propuesto por Fayyad et al., los cuales poseen las etapas de entendimiento del negocio, entendimiento de los datos, procesamientos de los datos, modelamiento o minería de datos, evaluación de modelos y ,finalmente, paso a producción o visualización. La característica principal de estas metodologías es que consisten en etapas secuenciales y sin mucha flexibilidad para la aceptación de cambios. La figura 2.13 permite la visualización de la secuencia de etapas.



Fuente: (Mouhib Alnoukari, 2012, p. 75)
 Figura 2.13. Etapas del proceso tradicional de KDD

(ii) **Ontology-based KDP Approach**

Se basa en la mezcla de ingeniería ontológica (Anexo A - 8) y un enfoque tradicional para KDP.

(iii) **Web-based KDP Approach**

Incorpora las mismas etapas que la primera categoría, pero reconoce de forma especial a la información proveniente de *web logs* y ,por lo mismo, define etapas precisas para el tratamiento de la misma.

(iv) **Agile-based KDP Approach**

Esta categoría agrupa aquellos modelos que incorporan metodologías ágiles a un enfoque tradicional para KDP.

2.3.1.1. Modelos

Los mismos autores mencionados anteriormente, Mouhib y Asim, reconocen al mismo al mismo tiempos 9 modelos considerados líderes a lo largo de la historia de KDP, estos modelos son:

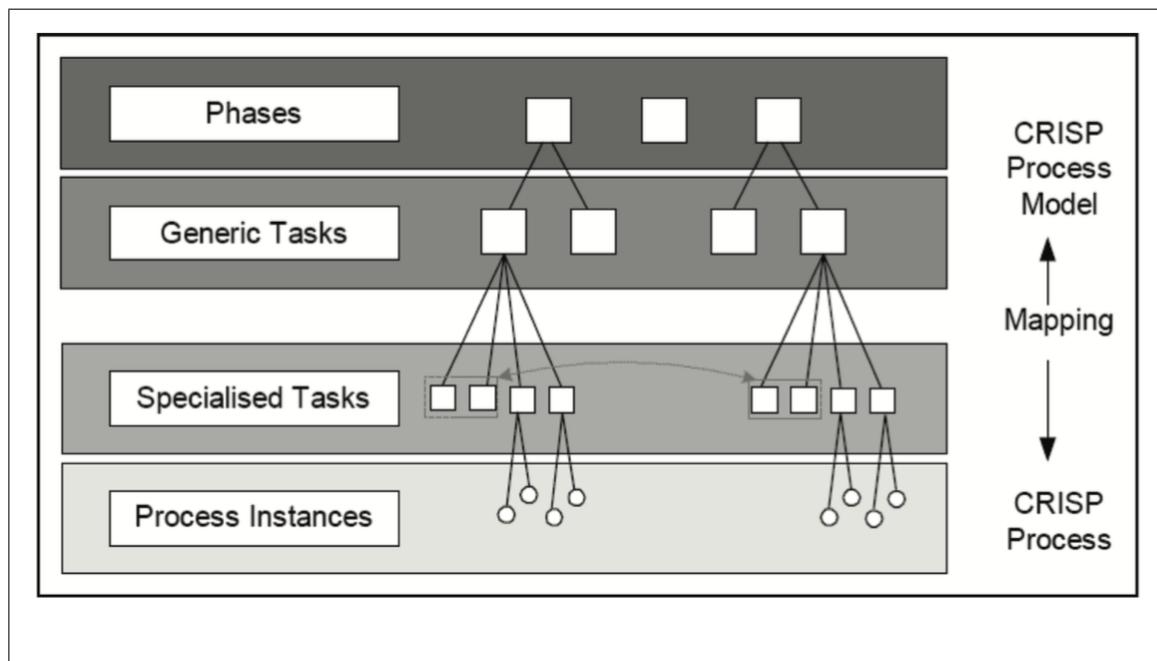
Tabla 2.1. Modelos a lo largo de la historia de *KDP*

Nombre	Autor	Año de Creación
Knowledge Discovery in Databases (KDD) Process	Fayyad et al.	1996
Information Flow in a Data Mining Life Cycle	Ganesh et al.	1996
SEMMA	SAS Institute	1997
Refined KDD paradigm	Collier et al.	1998
Knowledge Discovery Life Cycle (KDLC) Model	Lee and Kerschberg	1998
CRoss-Industry-Standard Process for Data Mining (CRISP-DM)	CRISP-DM	2000
Generic Data Mining Life Cycle by (DMLC)	Hofmann	2003
Ontology Driven Knowledge Discovery Process (ODKD)	Gottgtroy	2007
Adaptive Software Development-Data Mining (ASD-DM) Process Model	Alnoukari et al.	2008

Dentro de todos los modelos mencionados anteriormente se consideran relevantes para el presente trabajo CRISP-DM y ASD-DM, los cuales se describen las siguientes secciones.

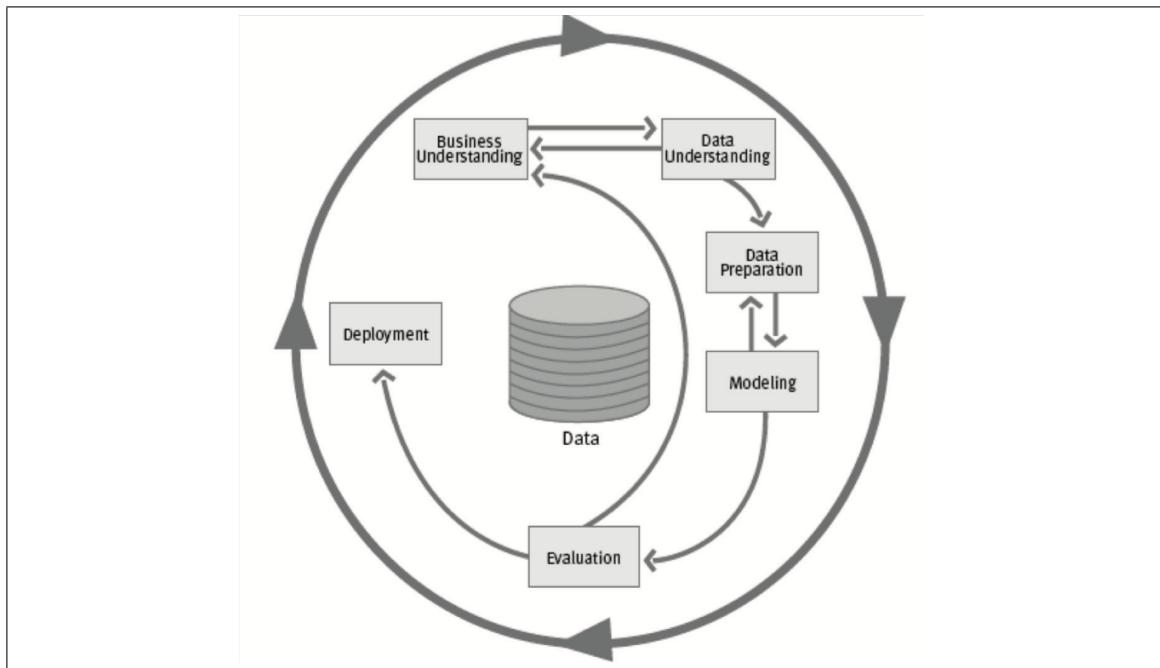
2.3.1.2. CRISP-DM

Este fue uno de los primeros modelos que intentó abarcar la problemática impuesta por el área de *Knowledge Discovery*. Para lograr lo anterior este modelo define 4 niveles de abstracción, en los cuales se realizan actividades, las cuales a su vez pertenecen a una de las 6 fases específicas por las cuales pasa todo proceso. La ilustración 2.14 muestra los 4 niveles creados en este modelo.



Fuente: (Mouhib Alnoukari, 2012, p. 82)
Figura 2.14. Niveles de Categorización de Actividades en CRISP-DM

Dentro de cada nivel se deben definir actividades, las cuales a su vez se encuentran categorizadas y divididas al mismo tiempo en sub tareas para poder llevarse a cabo. La intención de estos niveles jerárquicos es poder tener una clara agrupación de qué tareas específicas se realizan en cada proceso utilizado para la generación de conocimiento. Por otro lado, no existe un listado de actividades o tareas específicas que se deban realizar en cada uno de los niveles, sólo se espera que las tareas en las fases que se ilustran en el diagrama 2.15 contengan sus tareas o actividades categorizadas en los niveles previamente mencionados.



Fuente: (Mouhib Alnoukari, 2012, p. 83)

Figura 2.15. Fases en CRISP-DM

Por la definición de estas fases se considera que este modelo posee un enfoque tradicional para KDP. Es posible establecer las siguientes definiciones para cada una de las fases ilustradas anteriormente (Mouhib Alnoukari, 2012):

- (i) ***Business Understanding:*** Consiste en el entendimiento de los requerimientos generales del contexto en el cual se desenvuelve el proyecto así. En base a lo anterior, se crea una lista de objetivos que puedan ser respondidos por minería de datos y, finalmente, elemento se crea una planificación general del proyecto.
- (ii) ***Data Understanding:*** Esta etapa concierne específicamente a la recolección de los datos iniciales, descripción de ellos a través de metadata, exploración y verificación de la calidad de los mismos.
- (iii) ***Data Preparation:*** Corresponden a todas las actividades necesarias para la creación del conjunto de datos final que será utilizada en la siguiente fase de modelamiento y análisis. Se encuentra dividido en: selección, limpieza, construcción, integración y formateo de datos.

- (iv) ***Modeling:*** Orientado a la selección y aplicación apropiada de técnicas de modelamiento y minería de datos. Se encuentra seccionado en las fases de selección de técnicas de modelamiento, generación de pruebas, creación de modelos y, finalmente, evaluación de los modelos generados.
- (v) ***Evaluation:*** Relacionado a la evaluar si el modelo escogido cumple con los objetivos impuestos en la primera fase.
- (vi) ***Deployment:*** Etapa relacionada a la instalación a todo lo necesario para que el conocimiento generado puede ser ocupado por los usuarios finales. Esta etapa se caracteriza por presentar las siguientes fases: plan de instalación, plan de monitoreo y mantenimiento, generación de un reporte final donde se explica lo realizado y revisión de los pasos a seguir, en caso de ser requerido, de lo contrario, la finalización del proyecto.

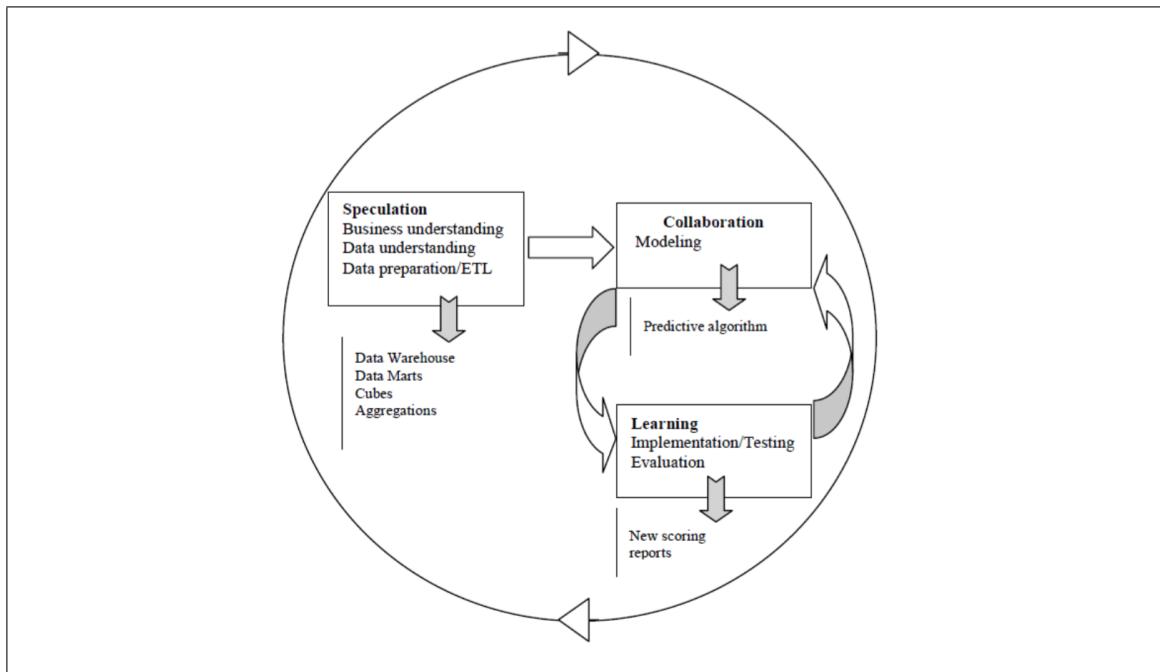
2.3.1.3. Adaptative Software Development - Data Minning (ASD-DM)

Este modelo se caracteriza por la incorporación de elementos presentes en metodologías ágiles para los proyectos de minería de datos. La idea basal sobre esta filosofía es que un enfoque adaptativo (flexible y/o incremental) posee un mejor rendimiento al momento de trabajar con requerimientos desconocidos o cambiantes, efecto que no es posible resolver de forma eficiente en los enfoques tradicionales de KDP.

Para lograr su objetivo, este modelo (Mouhib Alnoukari, 2012) define 6 etapas organizadas en 3 fases que se muestran en el diagrama 2.16.

Las etapas presentadas anteriormente pueden definirse del siguiente modo:

- (i) ***Etapa de Especulación (Speculation):*** Incluye todas actividades que se necesiten realizar para el entendimiento del negocio y la preparación de los datos. Esta etapa termina cuando se han creado las fuentes de información en forma de *data warehouse*, *data marts*, cubos o agrupaciones de los datos.



Fuente: (Mouhib Alnoukari, 2012, p. 90)

Figura 2.16. Fases en ASM-DM

- (ii) **Etapa de Colaboración (Collaboration):** Propicia toda actividad en donde se involucre a los stakeholders del proyecto para así poder escoger el mejor modelo para realizar la predicción posible.
- (iii) **Etapa de Aprendizaje (Learning):** Orientada a definir las tareas de pruebas y evaluación de los resultados obtenidos por el o los modelos escogidos en la fase anterior. En caso de ser los resultados aceptados por los *stakeholders*, se procede a la entrega de un reporte final con los resultados explicados, de contrario, se vuelve a la fase anterior.

2.3.2. Metodologías y modelos usadas en este trabajo

Si bien la metodología CRISP-DM presenta muchas de las cualidades que se necesitan para el desarrollo de este proyecto, no era la metodología más adecuada debido a que incluye una clara paso a producción una vez que la etapa de evaluación haya llegado a su fin. El objetivo de este proyecto es poder presentar una primera solución al cálculo de

la demanda de cursos del nuevo currículum de estudios, el cual aún no posee datos suficientes para poder realizar una generalización de datos para los próximos años. Frente a esta situación, se hace necesario que el proceso de aprendizaje se haga de un modo más continuo y que la nueva información que se vaya consiguiendo en el tiempo permita mejorar los niveles de predicción obtenidos. Teniendo en cuenta estos objetivos, se consideró que la metodología *Adaptive Software Development-Data Mining* era la apropiada gracias a la interacción continua entre sus fases collaboration y learning, logrando así una incorporación de nuevos datos a medida que pase el tiempo e ir mejorando así los modelos predictivos creados.

3. CAPÍTULO 03 : DISEÑO DE LA SOLUCIÓN

3.1. Introducción

El presente capítulo presenta los objetivos buscados y el planteamiento de una solución basada en el uso de técnicas de Ingeniería de Software y Descubrimiento del Conocimiento. Primeramente se presentan los requisitos funcionales expresados como relatos de usuario, luego requisitos no funcionales y finalmente la arquitectura. El capítulo termina con una descripción de los datos a utilizar.

3.2. Objetivos

Frente a la necesidad presentada de poder gestionar de forma adecuada la planificación de cursos y profesores dentro de la Escuela de Ingeniería en el contexto del nuevo currículum, se identifica como objetivo principal calcular la cantidad de alumnos para un curso específico en el siguiente periodo académico, es decir, calcular la demanda para un curso en el siguiente semestre académico. Este objetivo plantea una serie de objetivos secundarios. A continuación se enuncian estos objetivos secundarios para un mejor entendimiento del contexto en el cual se desenvuelve la problemática:

- (i) **Objetivo Principal:** Predecir en forma aproximada la demanda de alumnos para un curso específico en el siguiente semestre.
- (ii) **Objetivos Secundarios**
 - (a) Tener una representación de la estructura de unidades académicas de la universidad en cada uno de sus niveles: escuelas e institutos.
 - (b) Apoyar en la gestión de los cursos del primer ciclo formativo (Licenciatura), distribuidos en sus componentes básicas: plan de ciencias básicas, major y minor.
 - (c) Apoyar en la gestión de los majors y minors existentes así como la relación entre ellos.

- (d) Apoyar en la gestión de cursos en las unidades académicas.
- (e) Manejar la información histórica correspondiente al avance curricular por alumno en cada una de las generaciones.
- (f) Evaluar diferentes estrategias de entrenamiento y predicción en los análisis predictivos de modo de tener modelos actualizados.
- (g) Visualizar las estimaciones históricas realizadas en base a la información que se posee.

3.3. Grandes subsistemas y metodología de trabajo

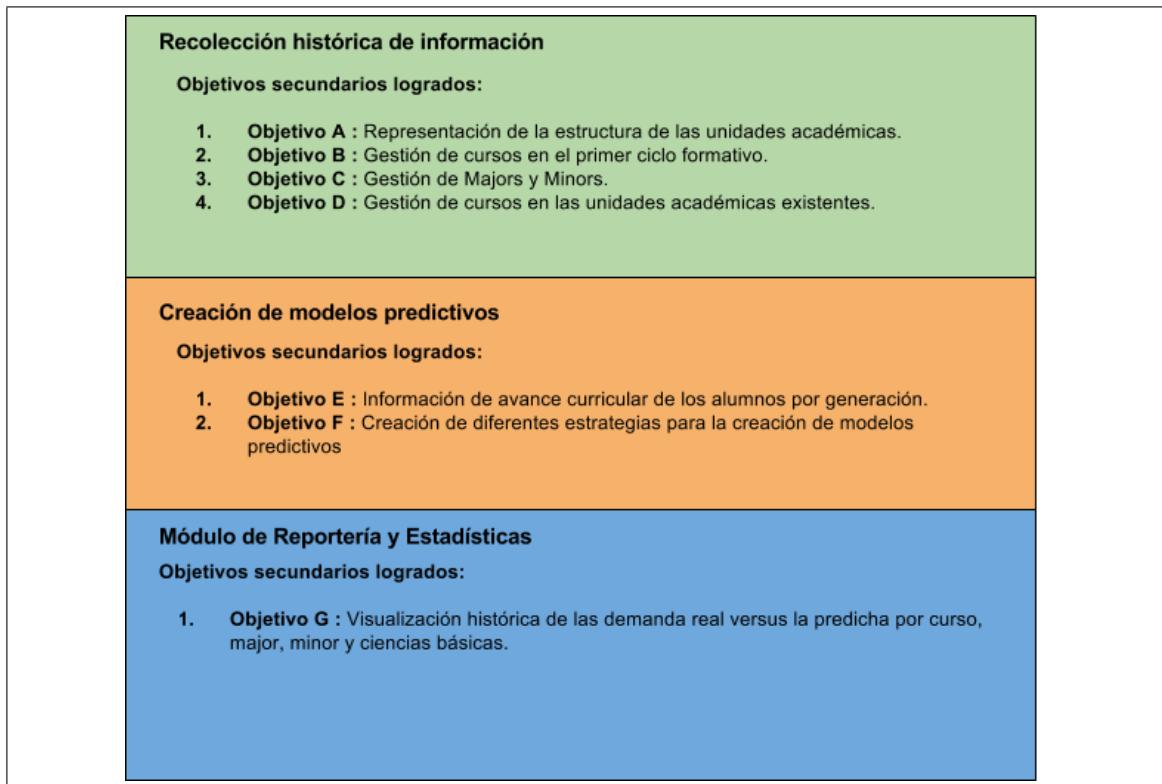
3.3.1. Subsistemas

La solución responde a los objetivos planteados mediante una arquitectura de 3 grandes subsistemas. Cada uno de ellos está asociado al cumplimiento de un conjunto determinado de objetivos. Así por ejemplo, la plataforma web permite el almacenamiento de la información histórica de los alumnos, el subsistema de modelos predictivos ataca la información de avance curricular y la creación de estrategias y el subsistema de reportería permite comparar los resultados para un curso en particular. El cuadro 3.1 muestra los objetivos asociados a cada uno de estos subsistemas.

3.3.2. Metodología de trabajo

La primera etapa incluye, por un lado, un levantamiento de los requerimientos funcionales, y por otro el análisis y preparación de los datos y utilizamos para ello métodos asociados a *Extreme Programming (XP)* y *Adaptive Software Development-Data Mining (ASD-DM)* respectivamente. Los requerimientos funcionales quedan sintetizados en forma de relatos de usuario asociados a un conjunto de épicas.

Luego de la etapa inicial descrita anteriormente, se da inicio al proceso de desarrollo de software propiamente tal en forma iterativa e incremental, es decir, la creación sucesiva de pequeños entregables acumulativos usando un proceso iterativo de conversaciones con



Fuente: Fuente Propia

Figura 3.1. Subsistemas y sus objetivos

los *stakeholders*. En paralelo con ello se lleva a cabo el proceso cíclico de modelación y evaluación de los algoritmos predictivos en *ASDM-DM*.

3.4. Requisitos

3.4.1. Funcionales

3.4.1.1. Tipos (Roles) de Usuarios

- (i) **Administrador:** Manejo de usuarios, modelos predictivos, ejecuciones de entrenamiento y predicción, informes.
- (ii) **Estadístico:** Rol encargado de la mejora de los modelos estadístico, para ello puede solicitar entrenar modelos con diferentes estrategias y carga de modelos DSF en el sistema para un curso en particular. También posee la opción de

realizar diferentes predicciones en un curso en particular o de todos los cursos cargados en el sistema.

- (iii) **Analista:** Puede correr nuevas predicciones y ver los resultados relativos a esas predicciones. Al mismo tiempo, puede ver la información acerca de los majors, minors, estudiantes y la historia de los mismos.

3.4.1.2. Épicas

Las épicas corresponden a grupos de relatos de usuario que en forma conjunta permiten constituir una funcionalidad mayor. La siguiente lista muestra las épicas identificadas en este proyecto:

- (i) **Administrador de Majors:** Permite la gestión de los majors dentro de la aplicación, es decir, la creación, eliminación , edición y la visualización dentro de la solución a implementarse. Adicionalmente, provee la posibilidad de administrar los cursos pertenecientes a cada major en particular.
- (ii) **Administrador de Minors:** Permite la gestión de los minors dentro de la aplicación, es decir, la creación, eliminación , edición y la visualización dentro de la solución a implementarse. Adicionalmente, provee la posibilidad de administrar los cursos pertenecientes a cada major en particular.
- (iii) **Carga de alumnos:** Corresponde a todas las funcionales que permiten la carga de información demográfica y de la universidad del alumno, tales como declaraciones de majors, minors, edad, generación, entre otras.
- (iv) **Carga de cursos:** Otorga la posibilidad de subir la información de los cursos y requisitos a través de un módulo de carga vía excel.
- (v) **Carga de información histórica de un alumno:** Es el módulo encargado de validar y permite subir el avance curricular de los alumnos al sistema vía un archivo excel. Además, entrega indica cuando alguna información no entrega el formato esperado por el sistema.

- (vi) **Validar información de los cursos:** Conjunto de funcionalidades que permiten establecer el formato adecuado para la carga de los cursos de la universidad por unidad académica. Además, debe indicar el campo o los campos que no cumplen con el formato exigido.
- (vii) **Validar información de alumno:** Conjunto de funcionalidades que permiten establecer el formato adecuado para la carga de los alumnos. Además, debe indicar el campo o los campos que no cumplen con el formato exigido.
- (viii) **Validar información histórica de un alumno:** Conjunto de funcionalidades que permiten establecer el formato adecuado para el avance curricular de los alumnos. Además, debe indicar el campo o los campos que no cumplen con el formato exigido.
- (ix) **Laboratorio Predictivo:** Es la infraestructura y conjunto de funcionalidad que permite ejecutar modelos en el lenguaje de programación R, cuyos resultados son almacenados en la base de datos que utiliza el sistema.
- (x) **Laboratorio de Entrenamiento:** Permite definir estrategias de entrenamiento sobre los datos históricos de los alumnos con el objetivo de poder mejorar la precisión de los modelos. Además, permite la gestión y almacenamiento de modelos en el formato RDS y *scriptps* (Anexo A - 9) del lenguaje R para la ejecución de modelos.
- (xi) **Otro:** Agrupa a todos aquellos relatos de usuarios que no encuentran en un epic anterior. Abarcan diferentes objetivos del software y ,por lo mismo, los relatos de usuarios no necesariamente presentan una correlación entre ellos.

3.4.2. Relatos de usuarios agrupados por Épicas

3.4.2.1. Administrador de Majors

- (i) Relato 01

(a) **Descripción:** Como administrador deseo desde el menú superior poder acceder a la opción Majors, así poder visualizar los majors creados en el sistema.

(b) **Criterios de aceptación**

(i) Dado que ingresé a la sección Majors, el sistema me muestra una tabla paginada de los majors existentes en el sistema ordenados por fecha de creación. Los campos de la tabla corresponden a: nombre major y departamento al cual pertenecen.

(ii) Dado que estoy en la tabla paginada de los majors existentes, desde uno en particular es posible acceder al detalle de un major, eliminarlo. Al realizar la eliminación se elimina el major y las asociaciones respectivas con los cursos, sin eliminar estos últimos.

(iii) Desde la tabla paginada, en la parte superior derecha existe un nuevo botón indicando **Nuevo Major**. El sistema despliega un formulario solicitando nombre y departamentos encargados.

(ii) Relato 02

(a) **Descripción:** Como un Administrador deseo poder visualizar el detalle de un major, el cual incluye un listado de cursos asociados al major indicando qué cursos son obligatorios y opcionales, así puedo visualizar el listado completo de cursos.

(b) **Criterios de aceptación**

(i) Dado que ingresé al detalle de un major, la tabla paginada me muestra la sigla del curso, nombre, nombre del departamento cual lo dicta e indicando si el curso es obligatorio u opcional.

(iii) Relato 03

(a) **Descripción:** Como un Administrador deseo poder incorporar un nuevo curso a un major disponible en el sistema, así puedo administrar la estructura de los majors.

(b) **Criterios de aceptación**

- (i) Dado que estoy visualizando la tabla de cursos en el major, el sistema en la parte superior derecha de la tabla se podrá visualizar un botón que indica **Agregar Curso**.
 - (ii) Dado incorporé un curso ya existente en la estructura, el sistema debe rechazar la incorporación de este.
- (iv) Relato 04
- (a) **Descripción:** Como un Estadístico deseo desde el menú superior poder acceder a la opción Majors, así poder visualizar los majors creados en el sistema.
 - (b) **Criterios de aceptación**
 - (i) Dado que ingresé a la sección Majors, el sistema me muestra una tabla paginada de los majors existentes en el sistema ordenados por fecha de creación. Los campos de la tabla corresponde a: nombre major y departamento al cual pertenecen.
 - (ii) Dado que estoy la tabla paginada de los majors existentes, desde uno en particular es posible acceder al detalle de un major, eliminarlo. Al realizar la eliminación se elimina el major y las asociaciones respectivas con los cursos, sin eliminar estos últimos.
- (v) Relato 05
- (a) **Descripción:** Como un Estadístico deseo poder visualizar el detalle de un major, el cual incluye un listado de cursos asociados al major indicando qué cursos son obligatorios y opcionales, así puedo visualizar el listado completo de cursos.
 - (b) **Criterios de aceptación**
 - (i) Dado que ingresé al detalle de un major, la tabla paginada me muestra la sigla del curso, nombre, nombre del departamento cual lo dicta e indicando si el curso es obligatorio u opcional.
- (vi) Relato 06

(a) **Descripción:** Como un Estadístico deseo poder incorporar un nuevo curso a un major disponible en el sistema, así puedo administrar la estructura de los majors.

(b) **Criterios de aceptación**

(i) Dado que estoy visualizando la tabla de cursos en el major, el sistema en la parte superior derecha de la tabla se podrá visualizar un botón que indica **Agregar Curso**.

(ii) Dado incorporé un curso ya existente en la estructura, el sistema debe rechazar la incorporación de este.

3.4.2.2. Administrador de Minors

(i) Relato 01

(a) **Descripción:** Como un Administrador deseo desde el menú superior poder acceder a la opción Minors, así puedo reorganizar nuevos elementos dentro del minor.

(b) **Criterios de aceptación**

(i) Dado que accedí a la opción Minors, se muestra una tabla pagina de los minors en el sistema, indicando nombre, fecha de creación y departamentos/facultades participantes en esté.

(ii) Relato 02

(a) **Descripción:** Como un Administrador deseo poder incorporar un nuevo curso a un minor disponible en el sistema, así puedo identificar el nivel de importancia del minor dentro de los estudiantes.

(b) **Criterios de aceptación**

(i) Dado que estoy visualizando la lista de minors, en la parte superior derecha existe un botón que indica **Nuevo Minor**, al presionarlo se abre otra ventana con los datos en la tabla.

- (ii) Dado que estoy en el formulario de incorporación de cursos al minor, el curso debe existir previamente en el sistema. No es posible crearlo desde esta ventana.
- (iii) Dado que estoy en un formulario de asociación, el sistema debe consultar si el curso dentro del minor es optativo u obligatorio.

(iii) Relato 03

(a) **Descripción:** Como un Administrador deseo poder visualizar la estructura de cursos y alumnos han inscrito el minor, así puedo visualizar la cantidad total de minors en el sistema.

(b) **Criterios de aceptación**

- (i) Dado que estoy en la lista de minor disponibles en el sistema, en una columna existe una opción denominada **Ver detalle**, al hacer click en ella, el sistema me lleva a otra ventana donde se detalle una lista de los cursos pertenecientes al minor, alumnos que hay inscritos en él y un botón que indica **Incorporar curso**.
- (ii) Dado que estoy en la lista de cursos disponibles en el minor, al lado de cada curso existen las opciones de **desvincular** y cambiar estado de obligatorio a opcional y viceversa.

(iv) Relato 04

(a) **Descripción:** Como un Estadístico deseo desde el menú superior poder acceder a la opción Minors, así puedo reorganizar nuevos elementos dentro del minor.

(b) **Criterios de aceptación**

- (i) Dado que accedí a la opción Minors, se muestra una tabla página de los minors en el sistema, indicando nombre, fecha de creación y departamentos/facultades participantes en este.

(v) Relato 05

(a) **Descripción:** Como un Estadístico deseo poder incorporar un nuevo curso a un minor disponible en el sistema, así puedo identificar el nivel de importancia del minor dentro de los estudiantes.

(b) **Criterios de aceptación**

(i) Dado que estoy visualizando la lista de minors, en la parte superior derecha existe un botón que indica **Nuevo Minor**, al presionarlo se abre otra ventana con los datos en la tabla.

(ii) Dado que estoy en el formulario de incorporación de cursos al minor, el curso debe existir previamente en el sistema. No es posible crearlo desde esta ventana.

(iii) Dado que estoy en un formulario de asociación, el sistema debe consultar si el curso dentro del minor es optativo u obligatorio.

(vi) Relato 06

(a) **Descripción:** Como un Estadístico deseo poder visualizar la estructura de cursos y alumnos han inscrito el minor, así puedo mantener actualizado la información de los alumnos en el sistema.

(b) **Criterios de aceptación**

(i) Dado que estoy en la lista de minor disponibles en el sistema, en una columna existe una opción denominada **Ver detalle**, al hacer click en ella, el sistema me lleva a otra ventana donde se detallan una lista de los cursos pertenecientes al minor, alumnos que hay inscritos en él y un botón que indica "**Incorporar curso**".

(ii) Dado que estoy en la lista de cursos disponibles en el minor, al lado de cada curso existen las opciones de **desvincular** y cambiar estado de obligatorio a opcional y viceversa.

3.4.2.3. Carga de alumnos

(i) Relato 01

(a) **Descripción:** Como un Administrador deseo poder cargar un archivo excel con nuevos alumnos, así puedo ingresar nueva información histórica de alumnos

(b) **Criterios de aceptación**

(i) Dado que ingresé a la sección alumnos, el sistema me muestra un formulario con la opción de subir un archivo excel.

(ii) Dado que visualicé el formulario para subir archivos, debajo de él existe una plantilla excel indicando los elementos y formato que se requieren. Los campos solicitados son rut, nombre, apellido paterno, apellido materno, generación, declaración de major a través de su código, declaración de minors a través de su código, major escogido a través de su código y minor a través de su código.

(ii) Relato 02

(a) **Descripción:** Como un Estadístico deseo poder cargar un archivo excel con nuevos alumnos, así puedo ingresar nueva información histórica de alumnos.

(b) **Criterios de aceptación**

(i) Dado que ingresé a la sección alumnos, el sistema me muestra un formulario con la opción de subir un archivo excel.

(ii) Dado que visualicé el formulario para subir archivos, debajo de él existe una plantilla excel indicando los elementos y formato que se requieren. Los campos solicitados son rut, nombre, apellido paterno, apellido materno, generación, declaración de major a través de su código, declaración de minors a través de su código, major escogido a través de su código y minor a través de su código.

(iii) Relato 03

(a) **Descripción:** Como un Analista deseo poder cargar un archivo excel con nuevos alumnos, así puedo ingresar nueva información histórica de alumnos.

(b) Criterios de aceptación

- (i) Dado que ingresé a la sección alumnos, el sistema me muestra un formulario con la opción de subir un archivo excel.
- (ii) Dado que visualicé el formulario para subir archivos, debajo de él existe una plantilla excel indicando los elementos y formato que se requieren. Los campos solicitados son rut, nombre, apellido paterno, apellido materno, generación, declaración de major a través de su código, declaración de minors a través de su código, major escogido a través de su código y minor a través de su código.

(iv) Relato 04

- (a) Descripción:** Como un Administrador deseo poder revisar las cargas de alumnos que se han realizado, así puedo llevar un control de los alumnos que se han ingresado al sistema.

(b) Criterios de aceptación

- (i) Dado que estoy en la página inicial, desde el menú Alumnos es posible obtener una opción para acceder a la cargas realizadas.
- (ii) Dado que hice click en la opción de cargas realizadas, el sistema me muestra una tabla paginada con el responsable, cantidad de alumnos y fecha de realización de la carga.
- (iii) Dado que ingresé al detalle de una carga, es posible tener la opción de eliminar la información de los alumnos subidos. Se borra toda la información relacionada, historial de cursos, predicciones realizadas.

(v) Relato 05

- (a) Descripción:** Como un Estadístico deseo poder revisar las cargas de alumnos que se han realizado, así puedo llevar un control de los alumnos que se han ingresado al sistema.

(b) Criterios de aceptación

- (i) Dado que estoy en la página inicial, desde el menú Alumnos es posible obtener una opción para acceder a la cargas realizadas.

- (ii) Dado que hice click en la opción de cargas realizadas, el sistema me muestra una tabla paginada con el responsable, cantidad de alumnos y fecha de realización de la carga.

3.4.2.4. Carga de cursos

(i) Relato 01

- (a) **Descripción:** Como un Administrador deseo poder cargar cursos a través de un archivo excel, así puedo ingresarlos al momento de que se creen en la UC.

(b) **Criterios de Aceptación**

- (i) Dado que me encuentro en la sección cursos, arriba de la tabla existe un formulario para ingresar un nuevo archivo excel.
- (ii) Dado que me encuentro en la sección cursos, debajo del formulario para subir nuevos cursos puedo descargar un archivo excel de plantilla para cumplir el formato requerido.
- (iii) Dado que ingresé un formulario completo, con datos correctos e hice click en el botón Cargar Cursos, en caso de que cumplan el formato, el sistema debe entregar un mensaje de éxito.

(ii) Relato 02

- (a) **Descripción:** Como un Estadístico deseo poder cargar cursos a través de un archivo excel, así puedo ingresarlos al momento de que se creen en la UC.

(b) **Criterios de Aceptación**

- (i) Dado que me encuentro en la sección cursos, arriba de la tabla existe un formulario para ingresar un nuevo archivo excel.
- (ii) Dado que me encuentro en la sección cursos, debajo del formulario para subir nuevos cursos puedo descargar un archivo excel de plantilla para cumplir el formato requerido.

(iii) Dado que ingresé un formulario completo, con datos correctos e hice click en el botón Cargar Cursos, en caso de que cumplan el formato, el sistema debe entregar un mensaje de éxito.

(iii) Relato 03

(a) **Descripción:** Como un Analista deseo poder cargar cursos a través de un archivo excel, así puedo ingresarlos al momento de que se creen en la UC.

(b) **Criterios de Aceptación**

(i) Dado que me encuentro en la sección cursos, arriba de la tabla existe un formulario para ingresar un nuevo archivo excel.

(ii) Dado que me encuentro en la sección cursos, debajo del formulario para subir nuevos cursos puedo descargar un archivo excel de plantilla para cumplir el formato requerido.

(iii) Dado que ingresé un formulario completo, con datos correctos e hice click en el botón Cargar Cursos, en caso de que cumplan el formato, el sistema debe entregar un mensaje de éxito.

(iv) Relato 04

(a) **Descripción:** Como un Administrador deseo poder revisar la cargas que he realizado, así puedo llevar un seguimiento de qué cursos y cuándo se han cargado.

(b) **Criterios de Aceptación**

(i) Desde el menú superior Cursos, el sistema me muestra la opción de acceder a historial de cargas.

(ii) Dado que accedí al historial de descargar, el sistema me indica una tabla paginada indicando fecha de la carga de cursos, cantidad de cursos cargados y la opción para ver el detalle.

(v) Relato 05

(a) **Descripción:** Como un Administrador deseo poder visualizar el detalle de una carga histórica, así puedo revisar de modo específico qué cursos se cargaron y quién fue el usuario que lo realizó.

(b) Criterios de Aceptación

(i) Dado que presioné un click sobre la opción visualizar detalle desde la tabla página de historial de carga, el sistema me indica de modo general el usuario que realizó la carga, la cantidad de cursos cargados, la fecha de carga y una tabla paginada indicando los cursos subidos.

3.4.2.5. Carga de información histórica de un alumno

(i) Relato 01

(a) **Descripción:** Como un Administrador deseo poder información histórica de rendimiento de uno o varios a alumnos a través de un archivo excel, así puedo estar seguro de que el sistema usa la misma nomenclatura para la información histórica de alumnos.

(b) Criterios de aceptación

(i) Dado que me encuentro en la lista de alumnos disponibles en el sistema, desde un lugar entre el menú superior y la tabla de los alumnos, existe un formulario donde se me da la opción de subir el archivo excel.

(ii) Dado que estoy visualizando el formulario para el archivo excel, existe un link de descarga con la plantilla que puedo seguir. En ella se me indican los campos y formatos que debo seguir.

(ii) Relato 02

(a) **Descripción:** Como un Estadístico poder cargar la información histórica de rendimiento de uno o varios a alumnos a través de un archivo excel, así puedo estar seguro de que el sistema usa la misma nomenclatura para la información histórica de alumnos.

(b) Criterios de aceptación

(i) Dado que me encuentro en la lista de alumnos disponibles en el sistema, desde un lugar entre el menú superior y la tabla de los alumnos,

exista un formulario donde se me da la opción de subir el archivo excel.

(ii) Dado que estoy visualizando el formulario para el archivo excel, existe un link de descarga con la plantilla que puedo seguir. En ella se me indican los campos y formatos que debo seguir.

(iii) Relato 03

(a) **Descripción:** Como un Analista deseo poder información histórica de rendimiento de uno o varios alumnos a través de un archivo excel, así puedo estar seguro de que el sistema usa la misma nomenclatura para la información histórica de alumnos.

(b) **Criterios de aceptación**

(i) Dado que me encuentro en la lista de alumnos disponibles en el sistema, desde un lugar entre el menú superior y la tabla de los alumnos, exista un formulario donde se me da la opción de subir el archivo excel.

(ii) Dado que estoy visualizando el formulario para el archivo excel, existe un link de descarga con la plantilla que puedo seguir. En ella se me indican los campos y formatos que debo seguir.

3.4.2.6. Validar información de los cursos

(i) Relato 01

(a) **Descripción:** Como un Administrador deseo recibir un email cuando los cursos hayan sido validados y cargados en el sistema, así puedo asegurarme que la información cargada tenga la misma nomenclatura

(b) **Criterios de aceptación**

(i) Dado que ingresé un excel al formulario con datos incorrectos, el sistema me debe indicar la línea y columna que no cumple con el formato requerido.

- (ii) Dado que ingresé un sigla ya existente, el sistema debe indicarme que no es posible crear este curso indicando fila y sigla existente.
 - (iii) Dado que el ingresé una sigla, el nombre no puede estar vacío.
- (ii) Relato 02
- (a) **Descripción:** Como un Estadístico deseo recibir un email cuando los cursos hayan sido validados y cargados en el sistema, así puedo asegurarme que la información cargada tenga la misma nomenclatura
 - (b) **Criterios de aceptación**
 - (i) Dado que ingresé un excel al formulario con datos incorrectos, el sistema me debe indicar la línea y columna que no cumple con el formato requerido.
 - (ii) Dado que ingresé un sigla ya existente, el sistema debe indicarme que no es posible crear este curso indicando fila y sigla existente.
 - (iii) Dado que el ingresé una sigla, el nombre no puede estar vacío.
- (iii) Relato 03
- (a) **Descripción:** Como un Analista deseo recibir un email cuando los cursos hayan sido validados y cargados en el sistema, así puedo asegurarme que la información cargada tenga la misma nomenclatura.
 - (b) **Criterios de aceptación**
 - (i) Dado que ingresé un excel al formulario con datos incorrectos, el sistema me debe indicar la línea y columna que no cumple con el formato requerido.
 - (ii) Dado que ingresé un sigla ya existente, el sistema debe indicarme que no es posible crear este curso indicando fila y sigla existente.
 - (iii) Dado que el ingresé una sigla, el nombre no puede estar vacío.

3.4.2.7. Validar información de alumno

- (i) Relato 01

(a) **Descripción:** Como un Administrador deseo poder recibir el resultado de la carga de nuevos alumnos vía email y en la pantalla del usuario, así verificar que se use la misma nomenclatura para todos los participantes del sistema.

(b) **Criterios de Aceptación:**

(i) Dado que ingresé un archivo excel con información correcta, el sistema me indica un mensaje de éxito.

(ii) Dado que ingresé un archivo excel con información incorrecta en formato, el sistema me indica la fila y columna e indicando que existe un error de formato. Además, indica cómo el campo debe ser completado.

(iii) Dado que ingresé un archivo excel con información ya existente, el sistema me indica la fila y columna e indica cuál es la información duplicada.

(ii) Relato 02

(a) **Descripción:** Como un Estadístico deseo poder recibir el resultado de la carga de nuevos alumnos vía email y en la pantalla del usuario, así verificar que se use la misma nomenclatura para todos los participantes del sistema.

(b) **Criterios de Aceptación:**

(i) Dado que ingresé un archivo excel con información correcta, el sistema me indica un mensaje de éxito.

(ii) Dado que ingresé un archivo excel con información incorrecta en formato, el sistema me indica la fila y columna e indicando que existe un error de formato. Además, indica cómo el campo debe ser completado.

(iii) Dado que ingresé un archivo excel con información ya existente, el sistema me indica la fila y columna e indica cuál es la información duplicada.

(iii) Relato 03

(a) **Descripción:** Como un Analista deseo poder recibir el resultado de la carga de nuevos alumnos vía email y en la pantalla del usuario, así verificar que se use la misma nomenclatura para todos los participantes del sistema.

(b) **Criterios de Aceptación:**

(i) Dado que ingresé un archivo excel con información correcta, el sistema me indica un mensaje de éxito.

(ii) Dado que ingresé un archivo excel con información incorrecta en formato, el sistema me indica la fila y columna e indicando que existe un error de formato. Además, indica cómo el campo debe ser completado.

(iii) Dado que ingresé un archivo excel con información ya existente, el sistema me indica la fila y columna e indica cuál es la información duplicada.

3.4.2.8. Validar información histórica de un alumno

(i) Relato 01

(a) **Descripción:** Como un Administrador deseo poder saber que la información que incorporé en el excel se haya subido de forma exitosa, así puedo saber si se mantiene actualizada.

(b) **Criterios de Aceptación**

(i) Dado que subí un excel con información correcta, el sistema me debe indicar los elementos fueron subidos de forma exitosa.

(ii) Dado que subí un excel con información incorrecta, el archivo me debe indicar fila y columna del error, además indicar el tipo de error que existe. Pueden ser por formato o información ya existente.

(ii) Relato 02

(a) **Descripción:** Como un Estadístico deseo poder saber que la información que incorporé en el excel se haya subido de forma exitosa, así puedo saber si se mantiene actualizada.

(b) **Criterios de Aceptación**

- (i) Dado que subí un excel con información correcta, el sistema me debe indicar los elementos fueron subidos de forma exitosa.
- (ii) Dado que subí un excel con información incorrecta, el archivo me debe indicar fila y columna del error, además indicar el tipo de error que existe. Pueden ser por formato o información ya existente.

(iii) Relato 03

- (a) **Descripción:** Como un Analista deseo poder saber que la información que incorporé en el excel se haya subido de forma exitosa, así puedo saber si se mantiene actualizada.

(b) **Criterios de Aceptación**

- (i) Dado que subí un excel con información correcta, el sistema me debe indicar los elementos fueron subidos de forma existosa.
- (ii) Dado que subí un excel con información incorrecta, el archivo me debe indicar fila y columna del error, además indicar el tipo de error que existe. Pueden ser por formato o información ya existente.

3.4.2.9. Laboratorio Predictivo

(i) Relato 01

- (a) **Descripción:** Como un Administrador poder visualizar todos las predicciones realizadas en el sistema según tipo de predicción : curso, major,minor , departamento o conjunto de cursos., así puedo realizar análisis de modo granularidad por los tipo de predicciones realizadas.

(b) **Criterios de Aceptación**

- (i) Dado que inicié sesión, puedo visualizar un menú superior denominado Laboratorios y como submenú **Laboratorio predictivo**.
- (ii) Dado que hice click en la opción de **Laboratorio predictivo**, el sistema me dirige la página de las predicciones que se muestran en

una tabla paginada indicando nombre de la predicción, fecha de realización, tipo y botón de acciones. Las opciones son ver detalle y eliminar.

- (iii) Dado que hice click en eliminar, el sistema debe consultar si estoy seguro de querer realizarla. En caso de aceptar, se borra las predicciones realizadas y no los cursos y asociaciones entre alumnos.

(ii) Relato 02

- (a) **Descripción:** Como un Administrador crear una nueva predicción indicando su nombre, tipo de predicción (curso, major, minor, departamento o conjunto de cursos), así puedo conocer un orden de magnitud de la cantidad de alumnos que tendrá el curso.

(b) **Criterios de Aceptación**

- (i) Dado que seleccioné tipo de evaluación curso, puedo seleccionar un curso en específico a través de un select.
- (ii) Dado que seleccioné tipo de evaluación major o minor, el sistema carga una lista de forma automática de los cursos de los cuales se hará una predicción.
- (iii) Dado que seleccioné tipo de evaluación departamento, el sistema carga de forma automática la lista de cursos que dicta el departamento.

(iii) Relato 03

- (a) **Descripción:** Como un Administrador poder visualizar el resultado de la predicción por cursos al hacer click en **detalle** desde la tabla de predicciones realizadas, así puedo realizar visualizar si puedo realizar una predicción de forma correcta por tipo de predicción (curso, major, minor, departamento o conjunto de cursos).

(b) **Criterios de Aceptación**

- (i) Dado que realicé la predicción, los detalles esperados son : cantidad de alumnos por curso. En caso de existir más de un curso, se espera tener una tabla paginada ordenada por sigla de cada uno.

3.4.2.10. Laboratorio de Entrenamiento

(i) Relato 01

(a) **Descripción:** Como un Administrador poder visualizar todos los cursos que tienen asignado un modelo predictivo , así poder visualizar si el curso está listo para poder realizar un predicción de la cantidad de alumnos a tomar el próximo semestre.

(b) Criterios de Aceptación

(i) Dado que inicié sesión, en la barra superior puedo acceder al menú **Laboratorio** y como submenu **Laboratorio de entrenamiento**.

(ii) Dado que hice click en la opción **Laboratorio de entrenamiento**, puedo visualizar una lista de cursos en una tabla con el siguiente detalle : Sigla del curso, nombre del curso, estado (indica si posee o no un modelo predictivo cargado) y acciones, indicando ver detalle, cargar modelo, carga script de ejecución, carga script de entrenamiento y eliminar. (Los script cargados deben ser en código R).

(iii) Dado que seleccioné la opción eliminar, el sistema me indica un mensaje indicando si estoy seguro de hacerlo. En caso afirmativo, el sistema borra el script de ejecución del modelo junto al modelo cargado.

(ii) Relato 02

(a) **Descripción:** Como un Administrador poder visualizar el detalle de un curso en el laboratorio predictivo, así puedo estar seguro de que mi modelo se está entrenando con una buena calidad.

(b) Criterios de Aceptación

(i) Dado que hice click en la parte **detalle** de la lista de cursos en el laboratorio predictivo, el sistema me lleva a una página indicando sigla, nombre y departamento que lo dicta (en caso de ser un curso en conjunto, se muestra la lista de los departamentos), checklist de scripts cargados o creados para el o los cursos indicando si el script de predicción, entrenamiento y modelo se encuentran creados.

(ii) Dado que me encuentro en el detalle un curso en el laboratorio de entrenamiento, puedo visualizar una tabla con los entrenamientos realizados en el curso. Al momento de realizar un entrenamiento, se crea el modelo como un archivo RDS (en caso de uno existente, crea una nueva versión del archivo). La opción para crear un nuevo entrenamiento se encuentra en la parte superior de todos los realizados.

(iii) Relato 03

(a) **Descripción:** Como un Administrador poder realizar un nuevo entrenamiento a un curso en particular definiendo el porcentaje para el set de entrenamiento, validación y regularización (evitar overfitting). , así puedo evaluar qué modelo predice mejor.

(b) **Criterios de Aceptación**

(i) Dado que deseo crea un nuevo entrenamiento, debo definir los porcentajes para cada tipo de set de datos (entrenamiento, validación y regularización) y la semilla para organizar de forma random los elementos del set de entrenamiento.

(ii) Dado que se ha creado una predicción, se crea una nueva versión de los archivos de ejecución, entrenamiento y modelo.

(iv) Relato 04

(a) **Descripción:** Como un Administrador poder crear comparación de estrategias seleccionando diferentes porcentajes para los de entrenamiento y validación.

(b) **Criterios de Aceptación**

(i) Dado que apreté nueva predicción para un curso determinado, debo seleccionar qué tipo de entrenamiento será: simple o comparativo. En el primer caso se solicita sólo una vez los porcentajes para los sets de datos, en el segundo, dos veces para realizar la comparación.

- (ii) Dado que realicé una predicción, su detalle entrega la matriz de confusión de los datos, así como también las siguientes medidas de error rate, accuracy, recall y sensitivity.

3.4.2.11. Otro

(i) Relato 01

(a) **Descripción:** Como un Administrador poder ingresar al sistema usando mi email uc o de ingeniería, puedo acceder a todos los modelos del sistema.

(b) **Criterios de Aceptación**

- (i) Dado que tengo una cuenta en el sistema , debería ser capaz de acceder usando mi email y password.
- (ii) Dado que ingresé un email no válido para el sistema, debe indicarse que el email no existe.
- (iii) Dado que ingresé un texto en el campo email que no cumplía con el formato, se debe mostrar un mensaje de error.

(ii) Relato 02

(a) **Descripción:** Como un Analista poder ingresar al sistema usando mi email uc o de ingeniería, puedo acceder a todos los modelos del sistema.

(b) **Criterios de Aceptación**

- (i) Dado que recibí un email de creación de cuenta, debería ser capaz de acceder usando mi email y password.
- (ii) Dado que ingresé un email no válido para el sistema, debe indicarse que el email no existe.
- (iii) Dado que ingresé un texto en el campo email que no cumplía con el formato, se debe mostrar un mensaje de error.

(iii) Relato 03

(a) **Descripción:** Como un Estadístico poder ingresar al sistema usando mi email uc o de ingeniería, puedo acceder a todos los modelos del sistema.

(b) **Criterios de Aceptación**

- (i) Dado que recibí un email de creación de cuenta, debería ser capaz de acceder usando mi email y password.
 - (ii) Dado que ingresé un email no válido para el sistema, debe indicarse que el email no existe.
 - (iii) Dado que ingresé un texto en el campo email que no cumplía con el formato, se debe mostrar un mensaje de error.
- (iv) Relato 04
- (a) **Descripción:** Como un Administrador deseo poder recuperar mi contraseña recibiendo un email al email asociado a la cuenta creada.
 - (b) **Criterios de Aceptación**
 - (i) Dado que tengo mi cuenta creada, debería ser capaz de enviarme un email de recuperación de contraseña a través de un link.
 - (ii) Dado que recibí el email para cambiar mi contraseña e hice click en el link antes de las 24 hrs de su expiración, debería poder escoger una nueva contraseña con validación para ingresar nuevamente.
- (v) Relato 05
- (a) **Descripción:** Como un Estadístico deseo poder recuperar mi contraseña recibiendo un email al email asociado a la cuenta creada.
 - (b) **Criterios de Aceptación**
 - (i) Dado que tengo mi cuenta creada, debería ser capaz de enviarme un email de recuperación de contraseña a través de un link.
 - (ii) Dado que recibí el email para cambiar mi contraseña e hice click en el link antes de las 24 hrs de su expiración, debería poder escoger una nueva contraseña con validación para ingresar nuevamente.
- (vi) Relato 06
- (a) **Descripción:** Como un Estadístico deseo poder recuperar mi contraseña recibiendo un email al email asociado a la cuenta creada.
 - (b) **Criterios de Aceptación**

- (i) Dado que tengo mi cuenta creada, debería ser capaz de enviarme un email de recuperación de contraseña a través de un link.
- (ii) Dado que recibí el email para cambiar mi contraseña e hice click en el link antes de las 24 hrs de su expiración, debería poder escoger una nueva contraseña con validación para ingresar nuevamente.

(vii) Relato 07

(a) **Descripción:** Como un Administrador deseo poder manejar a los usuarios dentro del sistema que no sea administradores, así puedo corregir problemas relacionados a su información o interacción en el sistema.

(b) **Criterios de Aceptación**

- (i) Dado que inicie sesión como administrador, mi primera vista debe ser una lista página de los usuarios disponibles indicando nombre, apellido, email y rol.
- (ii) Dado que tengo privilegios de administrador, puedo resetear la contraseña de un usuario ya sea ingresando una nueva con verificación o enviándole un email para que él lo haga.
- (iii) Dado que tengo privilegios de administrador, puede deshabilitar un usuario sin borrar su cuenta (soft delete).

(viii) Relato 08

(a) **Descripción:** Como un Administrador deseo poder visitar la sección cursos, así puedo gestionar los cursos existentes.

(b) **Criterios de Aceptación**

- (i) Dado que ingrese a la sección desde el menú superior, debo poder visualizar todos los cursos ingresados al sistema en una lista página indicando nombre del curso , sigla, descripción y programa.
- (ii) Dado que estoy en sección cursos, en la lista paginada cada curso tiene la opción de visitar detalles, cambiar su estado (habilitado/deshabilitado). La opción eliminar queda habilitada sólo para el administrador.

(ix) Relato 09

(a) **Descripción:** Como un Analista deseo poder visitar la sección cursos, así puedo gestionar los cursos existentes.

(b) **Criterios de Aceptación**

- (i) Dado que ingrese a la sección desde el menú superior, debo poder visualizar todos los cursos ingresados al sistema en una lista paginada indicando nombre del curso , sigla, descripción y programa.
- (ii) Dado que estoy en sección cursos, en la lista paginada cada curso tiene la opción de visitar detalles, cambiar su estado (habilitado/deshabilitado). La opción eliminar queda habilitada sólo para el administrador.

(x) Relato 10

(a) **Descripción:** Como un Estadístico deseo poder visitar la sección cursos, así puedo gestionar los cursos existentes.

(b) **Criterios de Aceptación**

- (i) Dado que ingrese a la sección desde el menú superior, debo poder visualizar todos los cursos ingresados al sistema en una lista página indicando nombre del curso , sigla, descripción y programa.
- (ii) Dado que estoy en sección cursos, en la lista paginada cada curso tiene la opción de visitar detalles, cambiar su estado (habilitado/deshabilitado). La opción eliminar queda habilitada sólo para el administrador.

(xi) Relato 11

(a) **Descripción:** Como un Administrador deseo poder revisar alumnos en el sistema, así puedo realizar un seguimiento si la información de alumnos se encuentra actualizada en el sistema.

(b) **Criterios de Aceptación**

- (i) Dado que ingresé a la opción alumnos, el sistema me muestra un resumen indicando la cantidad de alumnos por generación y una tabla paginada de los alumnos ordenados por fecha de creación.
- (ii) Dado que me encuentro en el menú superior, me indica la opción **Alumnos**.

(xii) Relato 12

(a) **Descripción:** Como un Administrador deseo poder visualizar la información histórica de un alumnos en particular, así puedo mantener actualizado la información de los alumnos en el sistema.

(b) **Criterios de Aceptación**

(i) Dado que estoy en la sección **Alumnos**, desde la tabla de alumnos paginada existe la opción ver detalle. En el detalle se indica nombre, apellido, declaración de major, declaración de minor, lista de cursos tomados junto a su respectivos resultados.

(xiii) Relato 13

(a) **Descripción:** Como un Estadístico deseo poder visualizar la información histórica de un alumno en particular, así puedo mantener actualizado la información de los alumnos en el sistema.

(b) **Criterios de Aceptación**

(i) Dado que estoy en la sección **Alumnos**, desde la tabla de alumnos paginada existe la opción ver detalle. En el detalle se indica nombre, apellido, declaración de major, declaración de minor, lista de cursos tomados junto a su respectivos resultados.

(xiv) Relato 14

(a) **Descripción:** Como un Analista deseo poder visualizar la información histórica de un alumnos en particular, así puedo llevar un control de todos los cursos que se han realizado una predicción.

(b) **Criterios de Aceptación**

(i) Dado que estoy en la sección **Alumnos**, desde la tabla de alumnos paginada existe la opción ver detalle. En el detalle se indica nombre, apellido, declaración de major, declaración de minor, lista de cursos tomados junto a su respectivos resultados.

3.4.3. Requisitos No Funcionales

3.4.3.1. Rendimiento

- (i) El sistema debe indicar en no más de 1 segundo que ha iniciado ha procesar la solicitud de predicción de los modelos predictivos.
- (ii) El sistema debe demorarse a lo más 12 hrs en procesar su predicción más compleja, en caso de no hacerlo, debe informar al usuario los motivos.

3.4.3.2. Escalamiento

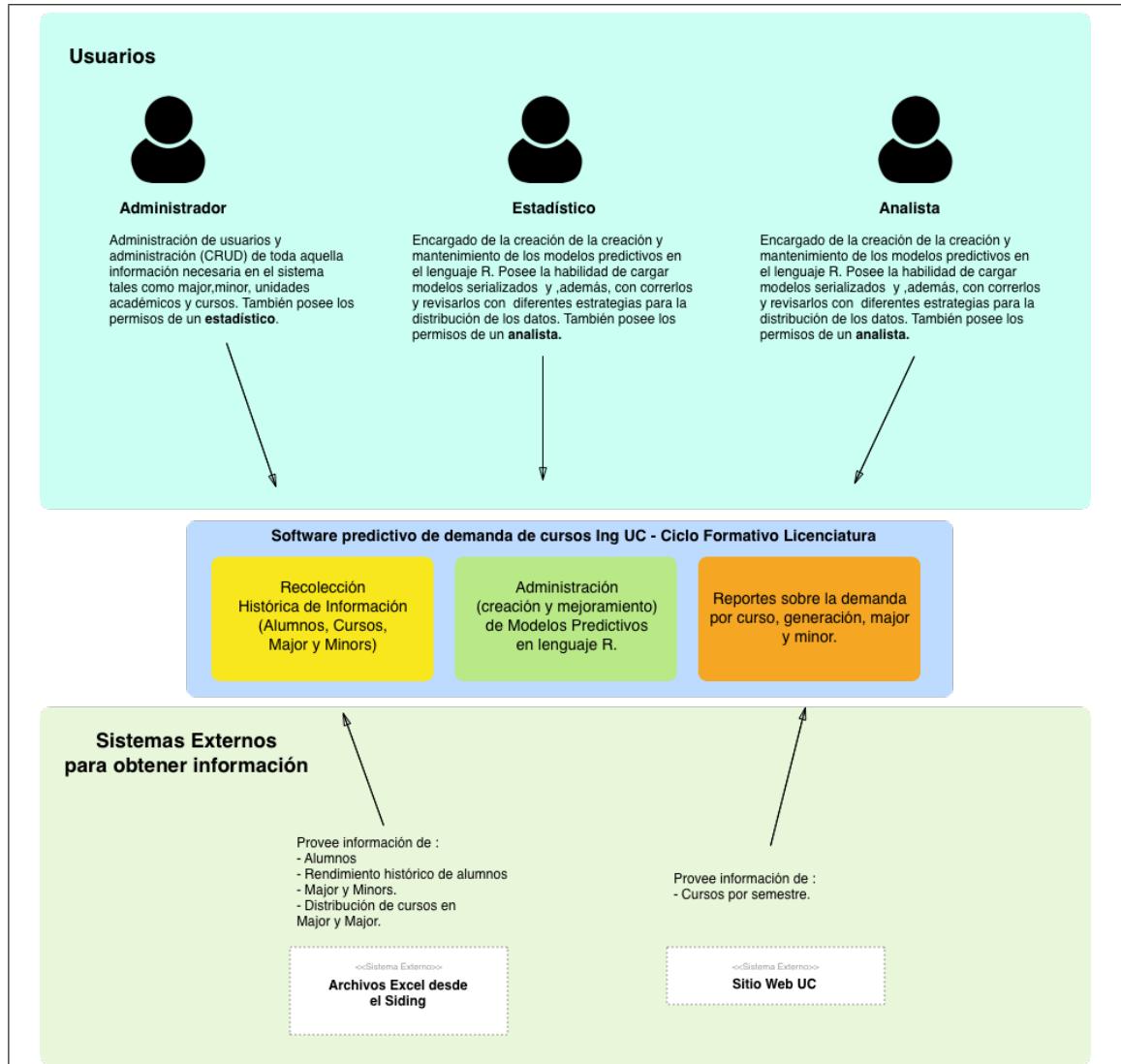
- (i) El sistema debe soportar la solicitud de una predicción ya sea para un curso o conjunto de ellos en cualquier instante.
- (ii) Para lograr mejoras en los procesamiento de las predicciones, el sistema ejecuta cada predicción de forma independiente en servers configurados en el sistema.

3.4.3.3. Usabilidad

- (i) El sistema usa el framework Bootstrap para lograr un estándar básico de diseño.
- (ii) Se realiza una prueba de usabilidad con los usuarios finales utilizando el manual del sistema, el cual contempla desde la carga de los datos hasta la visualización del resultado de un modelo predictivo.

3.5. Arquitectura

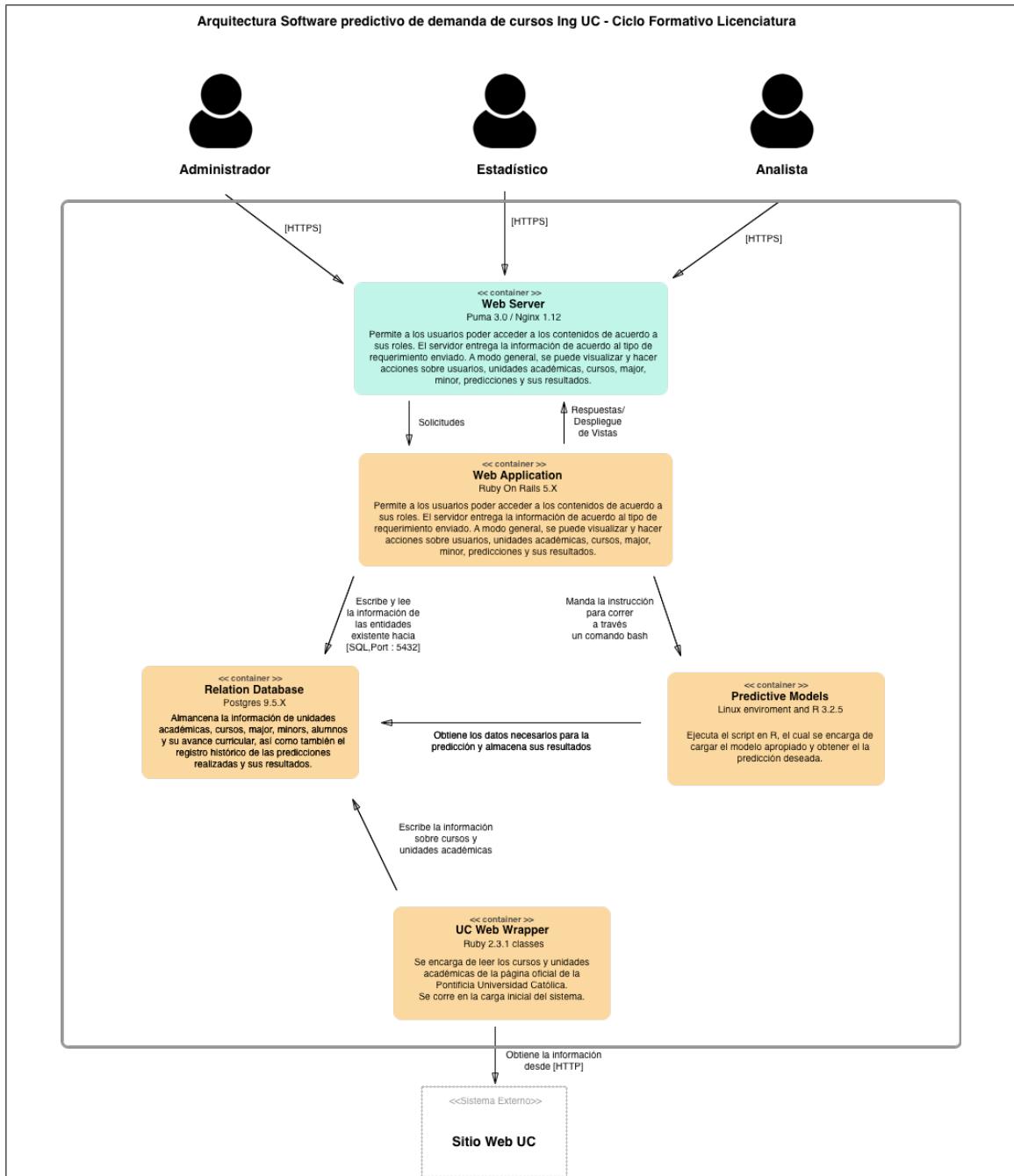
3.5.1. Diagrama de Contexto



Fuente: Fuente Propia

Figura 3.2. Diagrama de Contexto

3.5.2. Diagrama de Componentes



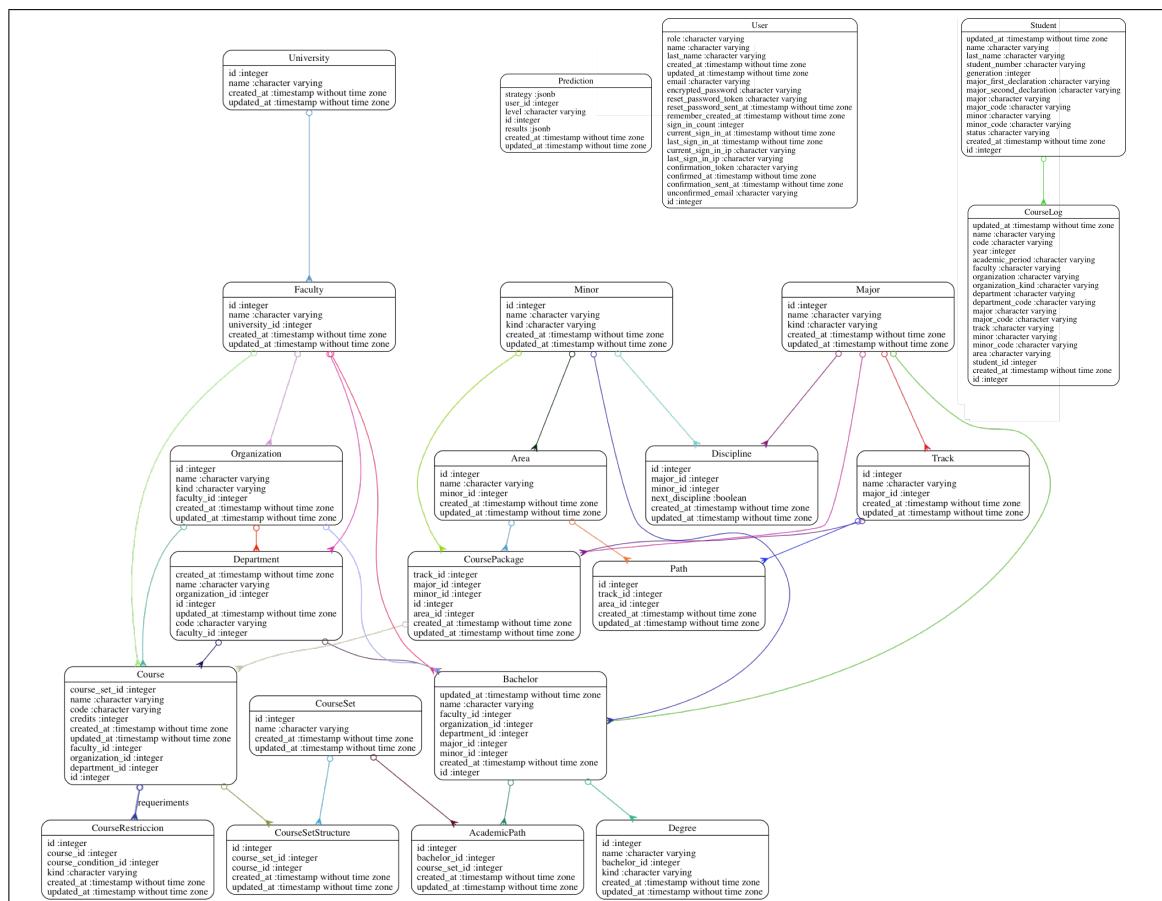
Fuente: Fuente Propia
Figura 3.3. Diagrama de Componentes

3.5.3. Descripción de los Datos

En las siguientes secciones se describen el conjunto datos y su estructura para realizar las predicciones de demanda de los cursos. En la primera parte se describe el modelo de base de datos utilizado y ,a continuación, el procedimiento de de cómo debe ir alimentándose el software para crear mantener la consistencia de la información.

3.5.3.1. Modelo de datos

El modelo de datos de la solución a implementar es el siguiente:



Fuente: Fuente Propia
Figura 3.4. Modelo de Base de Datos

Este modelo puede ser dividido de modo conceptual en las siguientes partes:

- (i) **Universidad:** Las tablas *University*, *Faculty*, *Organization*, *Department*, *Course* y *CourseRestrictions* permiten representar las relaciones entre la PUC, sus unidades académicas, los cursos que dictan y las requisitos que poseen.
- (ii) **Plan de Estudio:** El plan de estudio corresponde al conjunto de cursos que debe realizar un alumno para obtener un grado académico o título profesional UC. Las tablas que permiten almacenar esta información son: *Bachelor*, *Degree*, *AcademicPath*, *CourseSet*, *CoureSetStructure*, *Major*, *Track*, *Minor* y *Area*. Además de las tablas intermedias descritas en el modelo de datos.
- (iii) **Predicciones de demanda:** Son las tablas que permiten llevar el registro de las predicciones solicitadas y sus resultados respectivamente. Las tablas que permiten esto son: *Student*, *CourseLog* y *Predictions*. Cabe destacar que esta última tabla posee un enfoque SQL y NoSQL debido a que posee dos columnas *strategy* y *results* en formato JSONB, el permite almacenar datos en formato JSON en representación binaria.

El enfoque de diseño que se utilizó para el desarrollo de la base de datos fue modelo de Entidad Relación, para así permitir una mejor administración de unidades académicas, cursos y los alumnos dentro de la universidad en la plataforma. Con el fin de mantener las estadísticas y las predicciones de forma aislada , se crearon las tablas *Student*, *CourseLog* y *Predictions*.

3.5.3.2. Carga Inicial

Dado que las proyecciones de demanda se basan en la información histórica del avance curricular de los alumnos se hace necesario poder cargar de forma inicial los elementos mencionados a continuación con el objetivo de poder realizar predicciones al momento de poner en marcha el sistema. Los elementos que cargan son los siguientes:

- (i) **Universidad:** Nombre de universidad, facultades, centros de estudios (organization), escuelas (organization), departamentos, los cursos y sus requisitos o correquisitos de cursos.
- (ii) **Plan de Estudio:** Se hace la carga inicial de los major y sus tracks, minor y sus áreas, la relación entre ambos y los cursos pertenecientes a cada uno.
- (iii) **Avance Curricular:** Se hace carga del avance curricular de las generaciones del 2013 hacia adelante con la información actualizada hasta noviembre del 2016 entregada por Dirección de Pregrado.

3.5.3.3. Descripción de los Datos

Las cargas periódicas están enfocadas en mantener actualizada la información necesaria para las predicciones de demanda de cursos semestre a semestre, por lo tanto, se hace importante que el sistema cuenta con un sistema que permite realizar la carga de la siguiente información:

- (i) **Información de Cursos:** Corresponde a la carga masiva de cursos vigentes en la universidad, cabe destacar que el sistema no considera un curso en un periodo académico determinado, lo que busca es mantener actualizado sus cursos para determinar cuales son vigentes al momento de realizar una predicción.
- (ii) **Avance curricular de los alumnos:** Esta información es esencial para realizar predicciones más acertadas con el avance del tiempo.

Ambas informaciones descritas anteriormente deben ser cargadas al finalizar un periodo académico, ya sea semestre o temporada académica de verano. El sistema posee un sistema de alerta que permite recordar a los usuarios administradores vía email y notificaciones en la plataforma web de que es necesario realizar este proceso. Debido a que cada año un semestre (primero o segundo) o temperada académica de verano inician y terminan en fechas diferentes, se crea una forma genérica para ir calculando un inicio y término ficticio, el cual es ajustable por un usuario administrador. Los criterios de cambio de periodo académico son los siguientes:

Tabla 3.1. Criterios de fechas para cambios de periodos académicos

Periodo Académico	Fecha de Inicio Genérica	Fecha de Término Genérica
Temporada Académica de Verano (TAV)	02 de Enero	31 de Enero
Primer Semestre	01 de Marzo	15 de Julio
Segundo Semestre	01 de Agosto	15 de Diciembre

Las fechas descritas anteriormente permiten al sistema cuán actualizada está su información con respecto al tiempo presente, se podrán efectuar predicciones siempre y cuando esta información esté actualizada al periodo académico anterior de cuando se desea realizar.

4. CONCLUSIONS

Nothing to say. Be happy.

REFERENCES

- Agile User Stories,Epics and Themes.* (n.d.). Retrieved from <https://www.scrumalliance.org/community/spotlight/mike-cohn/march-2014/agile-user-stories-epics-and-themes>
- Association for Computing Machinery.* (n.d.). Retrieved from http://computingcareers.acm.org/?page_id=12
- Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Professional. Online.
- Cockburnstyle Use Cases.* (n.d.). Retrieved from https://commons.wikimedia.org/wiki/File:Cockburnstyle_use_cases.svg
- Créditos UC.* (n.d.). Retrieved from <http://admisionyregistros.uc.cl/alumnos/cursos/creditos-de-un-curso>
- Definición de Focus Group.* (n.d.). Retrieved from <https://www.definicionabc.com/comunicacion/focus-group.php>
- Extreme Programming Project.* (n.d.). Retrieved from <http://www.extremeprogramming.org/map/project.html>
- Gottesdiener. (1995). *BUSCAR NOMBRE - Arreglar*. WHO KNOWS. Hardcover.
- Guía Alumno Ingeniería UC 2017.* (n.d.). Retrieved from https://issuu.com/ingenieriauc/docs/guia_del_alumno_2017
- IEEE. (1990). *High Level Models and Methodologies for Information Systems*. Author. Online.
- Ingeniería Ontológica.* (n.d.). Retrieved from https://en.wikipedia.org/wiki/Ontology_engineering
- Institute, T. P. M. (2013). *A Guide to the Project Management of Body Knowledge (PMBOK GUIDE)*. IEEE. Online.
- Isaias P., I. T. (2015). *High Level Models and Methodologies for Information Systems*. Springer. Hardcover.

- M., C. (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional. Online.
- Massey, & Satao. (2012). *BUSCAR NOMBRE - Arreglar*. IEEE Computer Society, Los Alamitos, CA, USA. Hardcover.
- Minimal Viable Product*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Minimum_viable_product
- Mouhib Alnoukari, A. E. S. (2012). *Chapter 4 Knowledge Discover Process Models: From Traditional to Agile Modeling*. Business Science Reference (an imprint of IGI Global). Online.
- One, V. (2016). 10th annual state of agile report.
- P., B., & R., D. (2004). *Guide to the Software Engineering, Body of Knowledge - Arreglar*. IEEE Computer Society, Los Alamitos, CA, USA. Hardcover.
- Reglamento de Estructura Académica*. (n.d.). Retrieved from http://secretariageneral.uc.cl/images/Reglamento_sobre_Estructura_Acad%C3%A9mica_v_23_09_16.pdf
- Rumbaugh J., B. G., Jacobson I. (2004). *The Unified Modeling Language Reference Manual (2nd Edition)*. Addison-Wesley Professional. Online.
- Schwaber K., S. J. (2013). *The Scrum Guide*. Scrum.Org and ScrumInc. Hardcover.
- Scrum Diagram*. (n.d.). Retrieved from <https://jordanjob.me/2015/12/07/scrum-diagram/>
- Scrum Framework*. (n.d.). Retrieved from <https://www.neonrain.com/agile-scrum-web-development/>
- Sistema de Créditos Transferible Chile*. (n.d.). Retrieved from http://sct-chile.consejoderrectores.cl/que_es_sct_chile.php
- Sitio Web Ingeniería - PUC2*. (n.d.). Retrieved from <http://www.ing.uc.cl/alumnos/plan-de-estudios/descripcion/>
- User Story Format*. (n.d.). Retrieved from <http://yodiz.com/help/agile-user-stories-and-groomed-product-backlog/>

¿Qué es la integración Continua. (n.d.). Retrieved from <https://aws.amazon.com/es/devops/continuous-integration/>

ANEXO

A. GLOSARIO

(i) ***Refactoring***

Es el proceso de reconstrucción de código existente sin cambiar su comportamiento. Mejora los aspectos no funcionales del *software*, por ejemplo, rendimiento.

(ii) ***Test Driven Development***

Proceso de desarrollo de software el cual se basa en la repetición de un ciclo de tres pasos. Primero, el desarrollador escribe un test automático que define el comportamiento deseado del software; segundo, se escribe la mínima cantidad de código que hace el test pasar y ,finalmente, se hace un proceso de *refactor* del código.

(iii) **Programación a pares**

Proceso de programación donde dos programadores participan en un esfuerzo combinado de desarrollo en un sitio de trabajo. Cada miembro realiza una acción que el otro no está haciendo actualmente: Mientras que uno codifica las pruebas de unidades el otro piensa en la clase que satisfará la prueba, por ejemplo.

(iv) **Integración continua**

Práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas. La integración continua se refiere en su mayoría a la fase de creación o integración del proceso de publicación de software y conlleva un componente de automatización (p. ej., CI o servicio de versiones) y un componente cultural (p. ej., aprender a integrar con frecuencia). Los objetivos clave de la integración continua consisten en encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software (*¿Qué es la integración Continua*, n.d.).

(v) Focus Group

Tipo de técnica de estudio empleada en las ciencias sociales y en trabajos comerciales que permite conocer y estudiar las opiniones y actitudes de un público determinado (*Definición de Focus Group*, n.d.).

(vi) Epic

Agrupación de relatos de usuarios. También son considerados relatos de usuarios más grandes (*Agile User Stories,Epics and Themes*, n.d.).

(vii) Minimal Viable Product

Producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro (*Minimal Viable Product*, n.d.).

(viii) Ingeniería Ontológica

es un campo de las ciencias de la computación y ciencias de la información que estudia los métodos y metodologías para construir esquemas conceptuales (ontología): ésta corresponde a la representación formal de un grupo de conceptos dentro de un dominio y de las relaciones entre esos conceptos. Una representación a gran escala de conceptos abstractos como acciones, tiempo, objetos físicos y creencias podría ser un ejemplo de ingeniería ontológica (*Ingeniería Ontológica*, n.d.).

(ix) Script

Corresponde a una serie un conjunto de instrucciones programadas bajo un ambiente de programación único o mixto. Las instrucciones se escriben en función del lenguaje utilizado por el ambiente definido en el proyecto en el cual se utiliza.

B. AN INTERESTING SHORT STORY

Let us enjoy reading this story of Hunting With The Lion.

It was a dry summer. The animals in the forest were beginning to find it difficult to get food.

A bear, a wolf and a jackal thought it would be better to join hands with a lion and do the hunting. They approached lion and he too agreed. The four of them went off hunting.

The hunting party came across a buffalo. The fox and wolf chased the buffalo. The bear intercepted the buffalo. The lion killed him.

The fox made shares out of the buffalo. When they were about to take their shares the lion roared and said, "Well friends, the first share is mine for my leadership. The second share is mine for, it is I who killed. The third share is also mine for I need it for my cubs. Anyone who needs a share can take the fourth. But before that you will have to win me."

All the three left the place without a single word.

MORAL : If you are might, you are right.