

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO
“SISTEMA WEB DE ADMINISTRACIÓN PARA
EVALUACIÓN DOCENTE Y CURSOS DE TEMPORADA
(SAURON)”

CASO: CARRERA DE INGENIERÍA INDUSTRIAL

Proyecto de Grado para obtener el Título de Licenciatura en Informática

Mención: Ingeniería de Sistemas Informáticos

POR: ISRAEL JOSE HUALLPARA MENCIAS

TUTOR METODOLÓGICO: M.SC. ALDO VALDEZ ALVARADO

ASESOR: LIC. JOSE LUIS ZEBALLOS ABASTO

LA PAZ – BOLIVIA

Noviembre, 2017



UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mi mamá Nancy que con todo su amor pudo comprenderme, apoyarme y guiarme, nunca tendré con que agradecerte mamá.

A mi papá Roberto (†) por su amor, comprensión y por ser una pieza importante en mi vida. Te quiero mucho papá.

A mi hija Shecid y esposa Raquel que son el motor de mi vida, y el motivo de superación constante.

A mi familia por el apoyo y la confianza brindada todo este tiempo.

Israel José Huallpara Mencias

AGRADECIMIENTOS

Especialmente a mis amigos Lic. Omar Quispe, Lic. Jesús Perez, Lic. Wilson Alarcon, Lic. Efrain Rodriguez, Lic. Sergio Flores, Lic. Rolando Condori que en los últimos años de este ciclo fueron actores fundamentales para la culminación de la carrera y de este trabajo.

Muchas Gracias.

Al director de la Carrera de Ingeniería Industrial Ing. Oswaldo Terán por la confianza puesta para el desarrollo de este proyecto.

Al M.Sc. Aldo Valdez Alvarado Docente Tutor Metodológico por la paciencia, gracia y confianza que tuvo en la revisión, colaboración y recomendaciones para el desarrollo de este proyecto.

Al Lic. José Luis Zeballos Abasto Docente Asesor por brindarme los consejos necesarios y la confianza depositada en mi persona para el desarrollo de este proyecto.

A mi familia por todo el apoyo moral y constructivo que coadyuvaron en el desarrollo de mi vida.

¡Muchas gracias!!!

RESUMEN

La carrera de Ingeniería Industria de la U.M.S.A. enfrenta diversos procesos administrativos en todo el ciclo académico, que causan en sus actores molestias si en algún proceso este tiene retrasos o desorganización. El proyecto “Sistema Web de Administración para Evaluación Docente y Cursos de Temporada (SAURON)” tiene por objetivo colaborar a administrativos, docentes y estudiantes para mejorar el tiempo y esfuerzo de los procesos mencionados, además ser base centrar para futuros sistemas.

El sistema cuenta con un módulo central que es base para el funcionamiento de los otros módulos, que alberga la información esencial que la carrera maneja para los diversos procesos que enfrenta. El módulo Curso de Temporada se considera las pre inscripciones e inscripciones, además la habilitación automática de las materias y los reportes necesarios que reflejan el desarrollo del mismo. Para el módulo de Evaluación Docente son considerados los autores del proceso con sus diferentes responsabilidades, de igual forma el reporte los reportes de la autoevaluación docente o memoria, las ponderaciones efectuadas por el director de carrera como por el tribunal y las asignaciones de los últimos mencionados por el administrador del sistema.

Para el desarrollo de sistema se usa la metodología Scrum combinándolo con la metodología UWE para el análisis y diseño en base a los requerimientos recolectados de los procesos mencionados.

Para determinar la calidad el sistema, se emplea la norma ISO 9126 con todas las características necesarias y también cuenta con todas las medidas de seguridad para el buen funcionamiento y custodia de la información.

ÍNDICE

CAPÍTULO I	1
MARCO INTRODUCTORIO.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. ANTECEDENTES INSTITUCIONALES	2
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES.....	5
1.3. PLANTEAMIENTO DEL PROBLEMA	6
1.3.1. PROBLEMA CENTRAL	6
1.3.2. PROBLEMAS SECUNDARIOS.....	7
1.4. DEFINICIÓN DE OBJETIVOS	8
1.4.1. OBJETIVO GENERAL.....	8
1.4.2. OBJETIVOS ESPECÍFICOS.....	8
1.5. JUSTIFICACIÓN.....	9
1.5.1. JUSTIFICACIÓN ECONÓMICA.....	9
1.5.2. JUSTIFICACIÓN SOCIAL	10
1.5.3. JUSTIFICACIÓN TECNOLÓGICA	10
1.6. ALCANCES Y LÍMITES	11
1.6.1. ALCANCES	11
1.6.2. LÍMITES	12
1.7. APORTES	12
1.7.1. APORTE PRÁCTICO	12
1.7.2. APORTE TEÓRICO.....	13
1.7.3. METODOLOGÍA.....	14
CAPÍTULO II.....	17
MARCO TEÓRICO.....	17

2.1.	INGENIERÍA DE SOFTWARE	17
2.2.	METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE	19
2.2.1.	MANIFIESTO ÁGIL	20
2.2.2.	VALORES	20
2.2.3.	PRINCIPIOS.....	21
2.3.	SCRUM	22
2.3.1.	GESTIÓN DE LA EVOLUCIÓN DEL PROYECTO	23
2.3.2.	ROLES DE SCRUM	24
2.3.3.	ARTEFACTOS.....	27
2.3.4.	FLUJO DE TRABAJO	35
2.3.5.	MODELO DE PROCESO	38
2.3.5.1	PRE – GAME	38
2.3.5.2	GAME	40
2.3.5.3	POST – GAME	41
2.4.	INGENIERÍA WEB	41
2.5.	DESARROLLO WEB BASADA EN U. M. L. (U. W. E.)	42
2.5.1.	LENGUAJE UNIFICADO DE MODELADO (U.M.L.).....	42
2.5.2.	INGENIERÍA WEB BASADA EN U.M.L. (U.W.E.).....	43
2.5.2.1	FASES DE DESARROLLO.....	44
2.5.2.2	ANÁLISIS DE REQUISITOS	45
2.5.2.3	DISEÑO CONCEPTUAL	46
2.5.2.4	DISEÑO NAVEGACIONAL	47
2.5.2.5	DISEÑO DE PRESENTACIÓN	48
CAPÍTULO III		50
MARCO APlicativo		50
3.1.	INTRODUCCIÓN	50

3.1.1. IDENTIFICACIÓN DE ROLES SCRUM	51
3.1.2. NÚMERO DE ITERACIONES	51
3.2. PRE – GAME.....	52
3.2.1. RECOPILACIÓN DE REQUERIMIENTOS	52
3.2.2. MODELADO DE CASOS DE USO	54
3.2.3. DEFINICIÓN DEL CRONOGRAMA DE TRABAJO	67
3.2.4. ANÁLISIS DE RIESGOS.....	67
3.3. GAME.....	69
3.3.1. PRIMERA ITERACIÓN.....	70
3.3.1.1 MODELO CONCEPTUAL.....	71
3.3.1.2 MODELO DE NAVEGACIÓN	74
3.3.1.3 MODELO DE PRESENTACIÓN	74
3.3.2. SEGUNDA ITERACIÓN	79
3.3.2.1 MODELO DE NAVEGACIÓN	80
3.3.2.2 MODELO DE PRESENTACIÓN	81
3.3.3. TERCERA ITERACIÓN	86
3.3.3.1 MODELO DE NAVEGACIÓN	87
3.3.3.2 MODELO DE PRESENTACIÓN	88
3.4. POST – GAME	93
CAPÍTULO IV.....	97
CALIDAD Y SEGURIDAD DEL SOFTWARE	97
4.1. CALIDAD	97
4.1.1. FUNCIONALIDAD.....	97
4.1.2. USABILIDAD	100
4.1.3. MANTENIBILIDAD.....	102
4.1.4. PORTABILIDAD	103

4.1.5. CALIDAD TOTAL	103
4.2. SEGURIDAD	104
 4.2.1. POLÍTICAS DE SEGURIDAD (USUARIOS)	104
 4.2.2. POLÍTICAS DE SEGURIDAD ACCESO AL SOFTWARE	105
 4.2.3. ENCRIPCIÓN DE CONTRASEÑAS	105
 4.2.4. CONTROL DE ACCESOS Y RESTRICCIONES	106
CAPÍTULO V	107
ANÁLISIS COSTO – BENEFICIO	107
 5.1. ANÁLISIS DE COSTOS	107
 5.2. COCOMO II	107
 5.2.1. MÉTRICAS DE SOFTWARE	107
 5.2.2. ESTIMACIÓN DEL ESFUERZO	109
 5.2.3. COSTO DE SOFTWARE	111
 5.2.4. COSTO DE ELABORACIÓN DEL PROYECTO	111
 5.2.5. COSTE DE IMPLEMENTACIÓN DEL PROYECTO	111
 5.2.6. COSTE TOTAL	112
 5.3. ANÁLISIS DE BENEFICIOS	112
 5.3.1. VALOR ACTUAL NETO	113
 5.3.2. COSTO/BENEFICIO	114
 5.3.3. TASA INTERNA DE RETORNO	115
CAPÍTULO VI	117
CONCLUSIONES Y RECOMENDACIONES	117
 6.1. CONCLUSIONES	117
 6.2. RECOMENDACIONES	118
BIBLIOGRAFÍA	119
ANEXOS	121

ÍNDICE DE FIGURAS

Figura 1.1: Estructura Organizacional de la Carrera de Ingeniería Industrial.....	3
Figura 2. 1 Descripción resumida de los Roles en Scrum	27
Figura 2. 2: Ejemplo de una Pila de Sprint	31
Figura 2. 3: De la Pila del Sprint al Gráfico de Avance.	32
Figura 2. 4: Gráfico de Avance Previsto.	32
Figura 2. 5: Gráfico de Avance Real	33
Figura 2. 6: Gráfica de Avance Subestimado	33
Figura 2. 7: Gráfica de Avance Sobrestimado	34
Figura 2. 8: Descripción resumida de los Artefactos Scrum	34
Figura 2. 9: Descripción Resumida de los Eventos Scrum	37
Figura 2. 10: Flujo de trabajo del Marco Técnico Scrum	38
Figura 2. 11: Ejemplo de un Diagrama de Casos de Uso.	46
Figura 2. 12: Ejemplo de un modelo Conceptual – Diagrama de Clases	47
Figura 2. 13: Ejemplo de un modelo de Navegación.	48
Figura 2. 14: Ejemplo de un Modelo de Presentación.	49
Figura 3. 1: Diagrama de Casos de Uso de Alto nivel del Sistema.....	55
Figura 3. 2: Caso de Uso Administrar roles	55
Figura 3. 3: Caso de Uso Administrar Curso de Temporada	57
Figura 3. 4: Caso de Uso Administrar Evaluación Docente.....	58
Figura 3. 5: Caso de Uso Registra Informe Docente.....	60
Figura 3. 6: Caso de Uso Pondera y observa informe docente.....	61
Figura 3. 7: Caso de Uso Pondera actividad docente	62
Figura 3. 8: Caso de Uso habilita materia para inscripción.....	63
Figura 3. 9: Caso de Uso Registra pre inscripción	64

Figura 3. 10: Caso de Uso Registra inscripción	66
Figura 3. 11: Modelo conceptual autenticación del usuario.....	71
Figura 3. 12: Modelo conceptual módulo central del sistema.....	72
Figura 3. 13: Modelo conceptual módulo Cursos de temporada.....	73
Figura 3. 14: Modelo conceptual módulo Cursos de temporada.....	73
Figura 3. 15: Modelo navegacional para el Sprint 1	74
Figura 3. 16: Modelo de presentación módulo central de administrativos.....	75
Figura 3. 17: Modelo de presentación módulo central de docentes	75
Figura 3. 18: Modelo de presentación módulo central de estudiantes.	75
Figura 3. 19: Modelo de presentación módulo central de materias y carreras	76
Figura 3. 20: Modelo de presentación módulo central de acceso al sistema.....	76
Figura 3. 21: Pantalla de Inicio de Sesión	77
Figura 3. 22: Pantalla de módulo central de administradores	77
Figura 3. 23: Pantalla del formulario de registro y edición de administradores.....	78
Figura 3. 24: Pantalla de validación del eliminado de un administrador	78
Figura 3. 25: Modelo navegacional para el Sprint 2	80
Figura 3. 26: Modelo de presentación administración del curso de temporada	81
Figura 3. 27: Modelo de presentación módulo pre inscripción estudiantes	81
Figura 3. 28: Modelo de presentación módulo pre inscripción administrativo	82
Figura 3. 29: Modelo de presentación inscripción para estudiantes.....	82
Figura 3. 30: Modelo de presentación inscripción para administrativos.....	83
Figura 3. 31: Modelo de presentación inscripción para administrativos.....	83
Figura 3. 32: Pantalla de pre inscripciones del administrador.....	84
Figura 3. 33: Pantalla de materias habilitadas	84
Figura 3. 34: Pantalla de inscripción del administrador	85
Figura 3. 35: Pantalla de inscripción del administrador	85
Figura 3. 36: Pantalla de pre inscripción del estudiante.....	85
Figura 3. 37: Pantalla de inscripción del estudiante	86
Figura 3. 38: Modelo navegacional para el Sprint 3	88

Figura 3. 39: Modelo de presentación módulo de administración evaluación docente.....	88
Figura 3. 40: Modelo de presentación módulo de registro de logros por el docente titular	89
Figura 3. 41: Modelo de presentación módulo de ponderación por el director de carrera	89
Figura 3. 42: Modelo de presentación de ponderación y observación por el tribunal.....	90
Figura 3. 43: Pantalla de lista de evaluaciones docentes.....	90
Figura 3. 44: Formulario de registro de evaluación docente.....	90
Figura 3. 45: Formulario de asignación de tribunal	91
Figura 3. 46: se ve la pantalla del reporte en pdf del informe final.....	91
Figura 3. 47: Formulario de registro del informe docente	92
Figura 3. 48: Formulario de registro de ponderación por el director	92
Figura 3. 49: Formulario de registro de ponderación por el tribunal	93
Figura 3. 50: Configuración para pruebas con Jmeter.....	95
Figura 3. 51: Configuración para pruebas al servidor oficial	95
Figura 3. 52: Configuración para pruebas al servidor oficial	96

ÍNDICE DE TABLAS

Tabla 2. 1 Comparación de Metodologías de desarrollo de software	19
Tabla 2. 2: Ejemplo de una Pila de Producto	29
Tabla 3. 1: Identificación de Roles Scrum	51
Tabla 3. 2: Requerimientos Institucionales	54
Tabla 3. 3: Descripción del Caso de Uso “Administra Roles”	56
Tabla 3. 4: Descripción del Caso de Uso “Administra Cursos de temporada”	58
Tabla 3. 5: Descripción del Caso de Uso “Administra Evaluación Docente”.....	59
Tabla 3. 6: Descripción del Caso de Uso “Registra Informe Docente”	61
Tabla 3. 7: Descripción del Caso de Uso “Pondera y observa informe docente”	62
Tabla 3. 8: Descripción del Caso de Uso “Pondera y observa informe docente”	63
Tabla 3. 9: Descripción del Caso de Uso “Habilita materias para inscripción”	64
Tabla 3. 10: Descripción del Caso de Uso “registra preinscripción”	65
Tabla 3. 11: Descripción del Caso de Uso “registra inscripción”	67
Tabla 3. 12: Análisis de Riesgos	69
Tabla 3. 13: Backlog del primer sprint.....	70
Tabla 3. 14: Backlog del segundo sprint	79
Tabla 3. 15: Backlog del tercer sprint	87
Tabla 4. 1: Cálculo Punto Función.....	97
Tabla 4. 2: Valores de ajuste de complejidad.....	98
Tabla 4. 3: Ajuste de Complejidad Punto Función	99
Tabla 4. 4: Escala de ajuste de usabilidad	100
Tabla 4. 5: Evaluación de usabilidad	101
Tabla 4. 6: Factores de ajuste de mantenibilidad	102
Tabla 4. 7: Resultados Calidad Total norma ISO 9126.....	103

Tabla 5. 1: Factor LCD/PF de lenguajes de programación	108
Tabla 5. 2: Modelo básico para tipos de proyectos.	110
Tabla 5. 3: Coste de elaboración	111
Tabla 5. 4: Coste total	112
Tabla 5. 5: Cálculo del VAN.....	114

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

La presencia de las diversas instituciones ofreciendo servicios a su entorno mediante aplicaciones web va creciendo exponencialmente, debido a la comunicación constante que se debe de tener con el cliente, el cual desea consultar información en cualquier momento, lugar y dispositivo con el que se pueda conectar.

Por tal razón las instituciones deben de estar preparadas tecnológicamente para poder generar servicios de acuerdo a la demanda de sus clientes, esto genera bastantes beneficios tanto para la institución como para sus clientes los cuales tienden a requerir información general de la institución y específica de acuerdo al proceso a realizar.

Además, el hablar del crecimiento actualmente es una necesidad que se toma en cuenta al elaborar un sistema, por lo que se cuentan con diversas arquitecturas, en el presente proyecto se pondrá en ejecución la arquitectura de micro servicios ya que la institución busca que el proyecto pueda ser escalable con facilidad, además de integrable en ambas direcciones.

La Carrera de Ingeniería Industrial de la Universidad Mayor de San Andrés, realiza diversos procesos administrativos entre estos están la evaluación docente que se realiza por normativa cada año en el cual participan docentes titulares, a tiempo completo o a tiempo horario, además de los evaluadores entre ellos el director de carrera, centro de estudiantes y otros, de igual forma en todo el año se realizan dos cursos de temporada los cuales necesitan una organización en todo el proceso del mismo, desde generar una lista de las materias que los estudiantes desearían inscribirse, pasando por las inscripciones al curso hasta la publicación de los resultados de las inscripciones.

El objetivo de la Carrera de Ingeniería Industrial es el de agilizar los procesos anteriormente mencionados en las distintas unidades y usuarios, de igual forma reducir considerablemente la cantidad de documentación física generada, tiempo en cada uno de los procesos y estar preparados para el crecimiento de sus servicios informáticos para su comunidad docente, estudiantil y universitaria.

En este sentido el presente trabajo es el desarrollar un “SISTEMA WEB DE ADMINISTRACIÓN PARA EVALUACIÓN DOCENTE Y CURSOS DE TEMPORADA” que permitirá automatizar los procesos en ambos casos teniendo en cuenta similitudes y diferencias de ambos procesos para así llegar a un sistema escalable y predisposta a la integración con otros sistemas.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INSTITUCIONALES

La carrera de Ingeniería Industrial es una de las previstas en el plan de estudios aprobado por el consejo universitario el 10 de febrero de 1939, pero aún no se habían establecido sus estudios.

La carrera se basa en una sólida formación en las ciencias básicas, aplicadas y complementarias de ingeniería que permite adquirir profundos conocimientos en los aspectos técnicos, económicos y humanos inherentes a la marcha y evolución de las empresas, tanto las de producción como las de servicios.

Para una mejor comprensión de las unidades que conforman el estatuto orgánico de la carrera podemos observar la siguiente estructura orgánica (ver Figura 1.1).

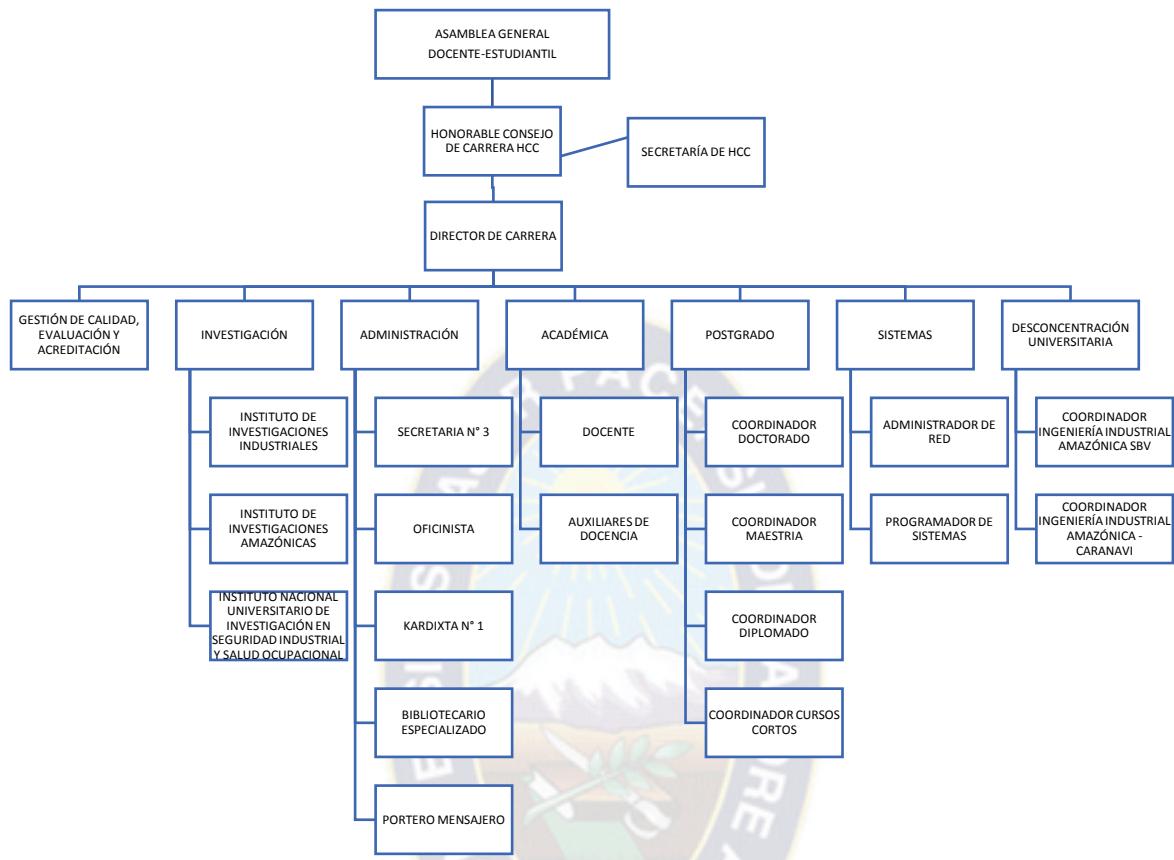


Figura 1.1: Estructura Organizacional de la Carrera de Ingeniería Industrial

Fuente: (Carrera de Ingeniería Industrial, 2017)

Misión. - Producir conocimiento científico y tecnológico, pertinente y relevante, que responde a la demanda y necesidades locales, regionales y nacionales, promoviendo el desarrollo industrial, con responsabilidad y compromiso social. Formar INGENIEROS INDUSTRIALES altamente calificados, con reflexión y pensamiento crítico; emprendedores y constructores de una sociedad justa e inclusiva. Promover la interacción con el Estado, la empresa, la sociedad, la comunidad científica y académica internacional, impulsando la transformación y búsqueda conjunta de soluciones innovadoras a la problemática social.

Visión. - Sostener el liderazgo a nivel nacional, en la formación de los mejores profesionales ingenieros industriales del Sistema de Universidad Boliviana, tanto a nivel público como privado, y comenzar el liderazgo en la especialización del Ingeniero Industrial a través de sus programas de Ingeniería en Producción Industrial, Ingeniería en Gestión Industrial, Ingeniería en Diseño Industrial e Ingeniería en Seguridad Industrial y Salud Ocupacional.

Objetivo de la carrera.- Formar profesionales Ingenieros Industriales, Ingenieros en Gestión Industrial, Ingenieros en Producción Industrial, Ingenieros en Diseño Industrial e Ingenieros en Seguridad Industrial y Salud Ocupacional de clase mundial con espíritu lógico, analítico, innovador, práctico, visionario y emprendedor con capacidad básica, científica, instrumental y de gestión, capaz de identificar variables fundamentales para el diseño, optimizar y gestionar procesos, haciendo investigaciones y optimizando sistemas de producción, generando bienes y servicios con responsabilidad, ética profesional y buenas relaciones humanas, coadyuvando en la preservación del medio ambiente y desarrollo regional y nacional en los niveles de Grado y Magister en el Posgrado.

Se pudo constatar que en la evaluación docente participan 35 docentes para ser evaluados, además actualmente se cuenta aproximadamente con 690 estudiantes activos y en promedio 500 de ellos se inscriben a cursos de temporada.

Además, mediante un meticuloso seguimiento en la evaluación docente se pudo constatar que la Carrera de Ingeniería Industrial no cuenta con algún sistema informático especializado que pueda ayudar en la administración del mismo, por lo cual el registro se realiza a través de formularios elaborados en alguna herramienta de ofimática, y copias pre elaboradas para ser llenadas posteriormente de forma manual.

Los cursos de temporada, de igual forma son administradas con registros manuales, solicitando al estudiante que registre su pre inscripción e inscripción en libros físicos, para posteriormente transcribirlo en una hoja electrónica de cálculo (Microsoft Excel), y poder generar listas de estudiantes y reportes para toma de decisiones.

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

Existen diversos proyectos realizados que pueden ser usados como referentes para la administración de procesos universitarios, sin embargo, cada uno presenta características particulares con una aplicación específica debida a los requerimientos que cada institución presenta.

Proyectos que fueron elaborados en la Carrera de Informática de la Universidad Mayor de San Andrés:

- Sistema Web Integrado de Gestión Académica Administrativa Caso: C.E.C.O.M.P.
Diego Paredes Mendoza, 2015.
Con el propósito de realizar un sistema bajo la plataforma web que permite gestionar el área administrativa y académica del instituto CECOMP. Utilizando SCRUMBAN como metodología de desarrollo y UWE como metodología de modelado.
- Sistema de Control y Generación de Partes de Asistencia de Docentes y Carga Horaria Académica Caso: Facultad de Ciencias Puras y Naturales
Sergio Adrián Flores Gutiérrez, 2016,
Con el propósito de realizar un sistema web que permite el control de partes de asistencia de los docentes de la Facultad de Ciencias Puras y Naturales. utilizando SCRUM como metodología de desarrollo y WebML como metodología de modelado,

desarrollado mediante el framework Laravel 5.0, con el gestor de base de datos MySQL.

- Sistema Web de Control y Seguimiento de Documentación Caso: Departamento de Infraestructura de la U.M.S.A.

Wilmer Mijhael Yucra Lecoña, 2016,

Con el propósito de realizar un sistema web que permite el control y seguimiento de la documentación del Departamento de Infraestructura de la U.M.S.A., Utilizando como metodología de desarrollo SCRUM y para el modelado UWE, desarrollado con el framework CodeIgniter 2 y el gestor de base de datos MySQL.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. PROBLEMA CENTRAL

Mediante una serie de entrevistas y visitas se pudo observar que la administración de la evaluación docente y de los cursos de temporada son realizados manualmente, desde los registros hasta los informes finales.

Todo este proceso lleva a la acumulación de papel que posteriormente es innecesario, en el caso de la evaluación docente se tienen normas establecidas para la evaluación por lo que se duplican esfuerzos al ponderar documentación que en gestiones anteriores ya fueron revisados, además la falta de comunicación de los actores tiende a demorar el proceso. En los cursos de temporada no se cuenta con datos suficientes en las pre inscripciones como para tomar una buena toma de decisiones para la habilitación de materias debido a la pre inscripción manual y presencial que se debe de realizar, así mismo la acumulación de

estudiantes para las inscripciones tiende a generar molestias tanto para los administrativos como para los estudiantes debido al tiempo, registro manual y presencial al mismo.

Tomando como base lo anteriormente mencionado se plantea el problema de la siguiente manera:

¿Cómo se puede administrar la evaluación docente y los cursos de temporada en la Carrera de Ingeniería Industrial de manera que se simplifiquen los procesos y se mantenga informados a los usuarios?

1.3.2. PROBLEMAS SECUNDARIOS

- Llenado manual de diversos formularios oficiales, lo que ocasiona errores de transcripción (errores humanos) y demora por correcciones.
- Acumulación de gran cantidad de documentación física que posteriormente no son útiles para la carrera.
- Omisión o mala interpretación de normas y/o ponderaciones establecidas para dicho proceso por el Reglamento de Evaluación Docente, lo que ocasiona una mala y desigual evaluación.
- Retraso de resultados finales por cantidad de documentación a transcribir, que conlleva a demoras en otros procesos administrativos externos y molestias para todos los involucrados.

- Procesos manuales y presenciales en los cursos de temporada, lo que ocasiona largas filas y un ambiente estresante tanto para estudiantes como administrativos.

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un sistema web de administración para evaluación docente y cursos de temporada para la Carrera de Ingeniería Industrial, que ayude a simplificar el proceso y mantenga informados a los actores del mismo, teniendo en cuenta la escalabilidad y posterior integración con otros sistemas.

1.4.2. OBJETIVOS ESPECÍFICOS

- Diseñar interfaces de registro acorde a formularios oficiales para su correspondiente llenado de manera simplificada y correcta.
- Registrar la documentación de forma digital en el sistema de manera que el documento físico sea validado para posteriores procesos y no haya la necesidad de solicitar una copia física del mismo.
- Desarrollar un adecuado mecanismo de registro y control basado en las normas establecidas por el Reglamento de Evaluación Docente.

- Emitir reportes finales y medios para así palear alguna demora y poder proporcionar información para una correcta toma de decisiones en ambos procesos.
- Implementar una serie de procesos para facilitar la administración, organización y control de los cursos de temporada además de disminuir los procesos presenciales.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN ECONÓMICA

Con el sistema propuesto, la administración de la evaluación docente y de los cursos de temporada que actualmente generan gastos considerables por el uso de material de escritorio, se reducirá considerablemente debido a que toda la información se mantendrá en formato digital y solo se generarán reportes físicos cuando sea necesario, además se reducirá el tiempo invertido en los diferentes procesos para todos los involucrados.

En relación a los costos por el desarrollo e implementación del nuevo sistema, estos serán mínimos, puesto que la propuesta hecha anteriormente se enfoca a trabajar con herramientas libres y certificadas, de igual forma se reducirán los costos para su mantenimiento y/o crecimiento ya que se manejará una estructura basada en micro servicios.

En cuanto a los requerimientos de hardware actualmente se cuenta con servicios de alojamiento por consumo, el cual reducirá los costos por el alojamiento y uso del sistema sin la necesidad que la carrera realice una inversión en servidores. Así mismo el estudiante ya no tendrá que realizar gastos innecesarios en la copia de documentos y formularios ni en acudir a la carrera para informarse sobre el estado del curso, ya que poseerá su cuenta en el cual se mantendrá informado constantemente.

1.5.2. JUSTIFICACIÓN SOCIAL

El proyecto beneficiará al personal administrativo en sus diferentes unidades ya que deben de administrar ambos procesos, además de los docentes y estudiantes designados como evaluadores ya que reducirá el tiempo invertido en la revisión de documentos y llenado de formularios en las distintas etapas.

Así mismo será mayormente beneficiada la población estudiantil ya que estará constantemente informada del estado del curso de temporada y los procesos que realizan será no necesariamente serán presenciales.

En cuanto a imagen se beneficiará la Carrera de Ingeniería Industrial por la tecnología de punta que está manejando el sistema.

1.5.3. JUSTIFICACIÓN TECNOLÓGICA

La Carrera de Ingeniería Industrial cuenta con una conexión estable y eficiente de internet en todos sus ambientes, para poder conectarse al sistema.

Las unidades de Kardex y dirección de carrera cuentan con equipos con procesadores Intel i5 y discos duros de hasta 1 TB, y memorias RAM de 4 GB.

En el desarrollo del sistema se utilizará para los servicios del Back-end NodeJs que permite una comunicación sin pérdida de velocidad sin importar la cantidad de usuarios que estén conectados por su procesamiento por hilos, el Front-end será desarrollado en Angular por su velocidad, así mismo se trabajará con el gestor de base de datos PostgreSQL.

1.6. ALCANCES Y LÍMITES

1.6.1. ALCANCES

Los alcances del proyecto de administración para evaluación docente y cursos de temporada se resumen en los siguientes módulos.

- Módulo de administración de usuarios docentes, estudiantes y administrativos que se encargará del registro correcto de los usuarios al sistema.
- Módulo de registro de informe-memoria por parte de docentes, que permitirá el registro simple y adecuado del mismo.
- Módulo de evaluación, que permitirá el registro de las ponderaciones por parte de los distintos actores de la evaluación.
- Módulo de preinscripción e inscripción estudiantil, que permitirá realizar el registro de preinscripción e inscripción por parte del estudiante.
- Módulo de administración de curso de temporada y evaluación docente, que se encargará de administrar el proceso de los mismo durante sus desarrollos por parte de los administrativos, así teniendo la capacidad de modificar especialmente las fechas de presentación o fechas límite.
- Módulo de reportes, que se encargará de generar reportes necesarios para la toma de decisiones adecuados.

1.6.2. LÍMITES

Los límites del presente proyecto según un análisis de requerimientos en relación a la evaluación docente y cursos de temporada son:

- No digitaliza documentos referentes al desarrollo de la evaluación ni del curso.
- No gestionará notas del estudiante.
- No controlará ingresos generados por los cursos.
- No se cuenta con una firma digital, por lo que las autoridades y responsables deberán hacerlo manualmente si fuera necesario.

1.7. APORTES

1.7.1. APORTE PRÁCTICO

La implementación del sistema proporcionará ciertos aportes a la Carrera de Ingeniería Industrial, entre estas podemos mencionar:

- Construcción de un sistema flexible que pueda proporcionar servicios a otros sistemas.
- Contribuir con la población estudiantil de la Carrera de Ingeniería Industrial, en simplificar el proceso del curso de temporada.

- Reducir la cantidad de documentos físicos y cambiarlos por registros digitales almacenados en el sistema.
- Uso de una arquitectura moderna basada en micro servicios y no así una monolítica tradicional.
- Uso de herramientas para el desarrollo ágil, como NodeJs, Angular, servicios REST FULL.
- Estructura pensada en una futura integración para otras aplicaciones o plataformas.

1.7.2. APORTE TEÓRICO

Para todo el proceso de desarrollo del sistema se emplea la metodología ágil SCRUM, para la parte del diseño y análisis del sistema se usa la metodología UWE, una herramienta de modelado basada en UML para aplicaciones web y además se emplea un patrón de desarrollo muy conocido como MVC (Modelo – Vista - Controlador) implementado en muchos frameworks de los lenguajes de programación más populares ya que ayuda bastante a los desarrolladores por la organización, distribución del código fuente y facilidad de programación, además que la arquitectura estará basada en micro servicios así preparándose para futuras integraciones y su escalabilidad.

La fusión entre Scrum y UWE se aplicará en las iteraciones teniendo a Scrum predominando el desarrollo del proyecto.

1.7.3. METODOLOGÍA

El presente proyecto se sostiene bajo el método científico por adecuarse al desarrollo del proyecto de grado propuesto anteriormente. En este sentido describiremos las tareas que propiamente se desarrollan con el método científico.

- Observación. Para desarrollar el sistema es necesario observar la estructura interna de los flujos de trabajo que presentan en todos los actores del proceso con respecto a los pasos que siguen para cumplir satisfactoriamente la convocatoria y sus roles en ella.
- Medición de variables, para poder evaluar las variables que se presentan en el sistema a desarrollar, es necesario evaluarlas con datos reales de las postulaciones, calificaciones y/o ponderaciones que se generan en el proceso y con eso verificar y subsanar errores cuantificables.

Podemos señalar tres tipos de investigación: la investigación exploratoria, la investigación causal y la investigación descriptiva.

Para la propuesta de este proyecto nos basaremos en la investigación del tipo exploratorio, puesto que se emplea cuando se está buscando un conocimiento más profundo sobre el problema, sus alternativas de decisión y las variables que se deben considerar, en este caso podemos tener un conocimiento previo en el cual podemos apoyarnos.

Por otra parte, terminaremos con la investigación de tipo descriptiva que comprende la recolección de datos, en relación a los beneficios que el proyecto abarcara en cuanto a los servicios que el sistema pueda proporcionar a todas las unidades implicadas.

Para el lenguaje de modelado nos basaremos en la experiencia de UWE, puesto que el momento en el que las páginas web eran pequeñas ventanas de información quedó atrás hace décadas, actualmente, poderosos motores web, mueven los mecanismos de información que usan miles de usuarios simultáneamente al acceder a sus sitios web favoritos como Facebook o YouTube.

Antes el diseño y creación de dichas páginas se realizaba de formas estructuradas siguiendo los pasos de programación convencional o incluso técnicas como RAD o XP, estirando al máximo lo que es UML, pero llegó el momento en que se han quedado cortas las herramientas y se perfilan modelos de diseño web adaptadas a las características básicas de Internet. Este el caso de UWE, Lenguaje de Modelado Web.

La metodología ágil de desarrollo que se aplicará es SCRUM, que está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. Scrum alienta a los desarrolladores a responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo.

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce.

Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empíricos: transparencia, inspección y adaptación.

En cuanto a los gestores, plataforma, lenguajes y Framework utilizados para el desarrollo del proyecto son:

- Gestor de base de datos PostgreSQL 9.6.3
- Herramienta de desarrollo: Atom
- Framework Insac y Angular
- Lenguaje de programación: JavaScript y NodeJs
- Vistas para el lado del cliente con Material Design de Google.



CAPÍTULO II

MARCO TEÓRICO

2.1. INGENIERÍA DE SOFTWARE

La ingeniería de Software es una disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven temas de todo tipo. Esta definición permite incluir dentro de la ingeniería de software a un gran número de áreas en la informática, desde desarrollos de sistemas operativos, construcción de compiladores hasta los nuevos desarrollos de software para aplicaciones basadas en aplicaciones enriquecidas en la web. [Pressman, 2012]

La ingeniería de Software es una tecnología con varias capas (Herramientas, Métodos, Procesos, entre otros), cualquier enfoque de ingeniería debe basarse en un compromiso organizacional de calidad.

El proceso define una estructura que debe establecerse para la obtención eficaz de tecnologías de ingeniería de software. El proceso de software forma la base para el control de la administración de proyectos de software, y establece un contexto en el que se aplican métodos técnicos, se generan modelos de trabajo, se establecen puntos de referencia, se asegura la calidad de software y se adapta a cambios de manera apropiada.

Los métodos de la ingeniería de software proporcionan la experiencia técnica necesaria para elaborar software, que incluye un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo.

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado o semi-automatizado para el proceso y los métodos.

Cuando se integran las herramientas de modo que la información obtenida pueda ser utilizada por otros actores, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software; a continuación, se detallan las herramientas más importantes: [Van Vliet, 2002].

- **Requerimientos de Software.** Un requerimiento se define como una exigencia que debe ser cumplida para dar solución a un problema específico. Existen diferentes áreas de análisis como ser la especificación de requerimientos, análisis, validación, clasificación, negociación, entre otros.
- **Diseño del Software.** Es el proceso en el cual se define la arquitectura, componentes, interfaces y otras características relativas al desarrollo del software mismo.
- **Construcción del Software.** Referida a la creación en detalle del Software a través de la combinación de las diferentes herramientas de codificación, verificación, pruebas de unidad, pruebas de integración y depuración.
- **Pruebas del Software.** Proceso de verificación dinámica del comportamiento del Software ante un conjunto limitado de casos de prueba.
- **Mantenimiento del Software.** Estas actividades de mantenimiento comienzan generalmente cuando el Software entra en la etapa de pruebas de funcionalidad con datos reales, pruebas que son realizadas por parte de los usuarios finales con la finalidad de que estos puedan adaptarse a los cambios, así mismo puedan sugerir cambios para su posterior adaptabilidad.

2.2. METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE

Las metodologías ágiles surgieron como una alternativa a las metodologías tradicionales pensando que estas son excesivamente pesadas y rígidas, por tener un enfoque normativo y fuerte dependencia a las planificaciones detalladas a largo plazo previas al desarrollo, puesto que algunos proyectos requerían otro tipo de gestión en donde los entornos de negocio del cliente son susceptibles de cambio y está continuamente en evolución. En contrapartida de las metodologías tradicionales, las metodologías ágiles son adaptativas y flexibles, ya que los cambios sugeridos por el cliente se pueden adecuar al equipo de desarrollo y al proyecto mismo en cualquier momento [Navarro, Fernández, & Morales, 2013].

Entonces, resaltando las virtudes de las metodologías ágiles, un proyecto se divide en varios subproyectos, atendidos de forma independiente viendo cuál de ellos tiene prioridad y estableciendo un lapso de entrega entre 2 a 6 semanas. Existe una constante comunicación con el cliente, lo que genera un alto grado de colaboración con el equipo de desarrollo y que el proyecto tenga mejoras de forma frecuente. En la Tabla 2.1 se puede ver una tabla comparativa entre ambas metodologías.

Metodologías Tradicionales	Metodologías Agiles
Predictivos	Adaptativos
Orientado a procesos	Orientado a personas
Proceso rígido	Proceso Flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Tabla 2. 1 Comparación de Metodologías de desarrollo de software

Fuente: [Fernandez & Morales, 2013]

2.2.1. MANIFIESTO ÁGIL

El manifiesto Ágil se compone de 4 valores y 12 principios. [Alaimo,2013]

2.2.2. VALORES

- **Valoramos más a los individuos y su interacción, por encima de los procesos y las herramientas:** Por supuesto que los procesos ayudan al trabajo. Son una guía de operación.
Las herramientas mejoran la eficiencia, pero hay tareas que requieren talento y necesitan personas que lo aporten y trabajen con una actitud adecuada.
- **Valoramos más al software que funciona, por encima de la documentación exhaustiva:** Poder anticipar cómo será el funcionamiento del producto final, observando prototipos previos, o partes ya elaboradas ofrece una retroalimentación estimulante y enriquecedor, que genera ideas imposibles de concebir en un primer momento y difícilmente se podrían incluir al redactar un documento de requisitos detallado en el comienzo del proyecto.
- **Valoramos más la colaboración con el cliente, por encima de la negociación contractual:** El objetivo de un proyecto ágil no es controlar la ejecución conforme a procesos y cumplimiento de planes, sino proporcionar el mayor valor posible al producto.
Resulta por tanto más adecuada una relación de implicación y colaboración continua con el cliente, más que un contractual de delimitación de responsabilidades.

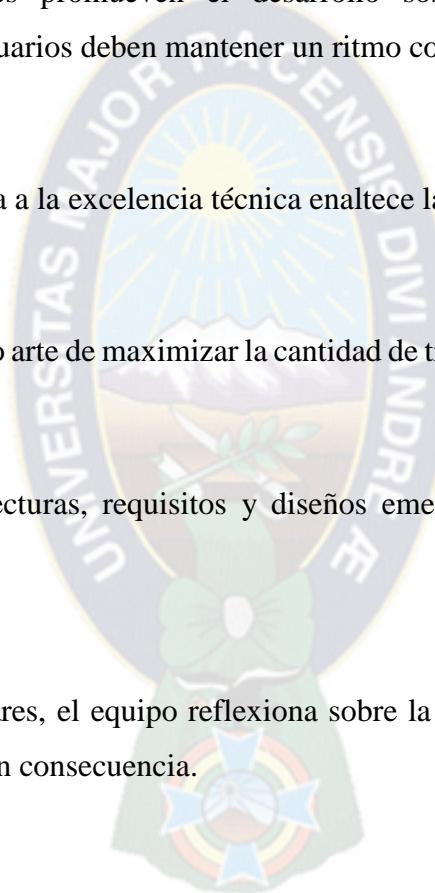
- **Valoramos más a la respuesta al cambio, por encima del seguimiento de un plan:**
Lo importante es adaptarse a los cambios que puedan ser sugeridos por el cliente, ya que en un inicio puede que no esté claro de lo que se quiere.

2.2.3. PRINCIPIOS

Vale aclarar que no se está desvalorizando las metodologías tradicionales, sino que se está dando un poco más de valor a las metodologías ágiles. Y en base a estos postulados se definieron 12 principios [Palacio, 2014]:

1. Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
2. Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblegan al cambio como ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con frecuencia en los períodos breves.
4. Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
5. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.

6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la principal medida de progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto organizan
12. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivos y ajusta su conducta en consecuencia.



2.3. SCRUM

Scrum es un modelo de desarrollo ágil caracterizado por [Palacio, 2014]:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más el conocimiento de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra de forma secuencial o en cascada.

Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para proyectos con requisitos que aún no están bien definidos y para los que requieran rapidez y flexibilidad, y esto se ve con frecuencia en el desarrollo de determinados sistemas de software.

2.3.1. GESTIÓN DE LA EVOLUCIÓN DEL PROYECTO

La metodología Scrum maneja la evolución del proyecto con las siguientes tácticas [Palacio, 2014]:

- **Revisión de las Iteraciones:** Al finalizar cada sprint se revisa funcionalmente el resultado con todos los implicados en el proyecto. Es por tanto la duración del sprint, el periodo máximo para descubrir planteamientos erróneos, mejorables o malinterpretaciones en las funcionalidades del producto.

- **Desarrollo incremental:** El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar. Durante la construcción se depura el diseño y la arquitectura, y no se cierran en una primera fase del proyecto. Las distintas fases que el desarrollo en cascada realiza de forma secuencial, en Scrum se solapan y realizan de forma continua y simultánea.
- **Autoorganización:** Son muchos los factores impredecibles en un proyecto. La gestión predictiva (métodos tradicionales) asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución. En Scrum (metodología ágil) los equipos son auto organizados, con un margen de maniobra suficiente para tomar las decisiones que consideren oportunas.
- **Colaboración:** Es un componente importante y necesario para que a través de la autoorganización se pueda gestionar con solvencia la labor que de otra forma realizaría el gestor de proyectos. Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

2.3.2. ROLES DE SCRUM

- **Propietario del producto (Product Owner):** El propietario del producto es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto. Decide en última instancia cómo será el resultado final, y el orden en el que se van construyendo los sucesivos incrementos (que se pone y que se quita de la pila del producto), y cuál es la prioridad de las funcionalidades. Conoce el plan del producto, sus posibilidades y plan de inversión, así como del retorno esperado a la inversión realizada, y se responsabiliza sobre fechas y funcionalidades de las diferentes versiones del mismo. Para ejercer este rol es necesario [Palacio, 2014]:

- Conocer perfectamente el entorno del negocio del cliente, las necesidades y el objetivo que se persigue con el sistema que se está construyendo.
 - Tener la visión del producto, así como las necesidades concretas del proyecto, para poder priorizar eficientemente el trabajo.
 - Recibir y analizar de forma continua retroinformación del entorno de negocio (evolución del mercado, competencia, alternativas) y del proyecto (sugerencias del equipo, alternativas técnicas, pruebas y evaluación de cada incremento).
-
- **Scrum Master:** Según Palacio [2014], el Scrum Master es el responsable del cumplimiento de las reglas del marco técnico Scrum, asegurándose de que se entiendan en la organización, y se trabaje conforme a ellas. El Scrum Master realiza su trabajo con un modelo de liderazgo servil, al servicio y en ayuda del equipo y del propietario del producto, proporcionando:
 - Asesoría y formación al equipo para trabajar de forma auto organizada y con responsabilidad de equipo.
 - Revisión y validación de la pila del producto.
 - Moderación de las reuniones.
 - Resolución de impedimentos que en el sprint puedan entorpecer la ejecución de las tareas.

- Gestión de las dinámicas del grupo en el equipo.
 - Configuración, diseño y mejora continua de las prácticas de Scrum en la organización, respeto de la organización y los implicados, con las pautas de tiempos y formas de Scrum.
-
- **Equipo de Desarrollo:** Según Palacio [2014], el Equipo de Desarrollo, lo forman el grupo de profesionales que realizan el incremento de cada sprint. Se recomienda que un equipo Scrum tenga entre 4 a 8 personas. No son considerados parte del grupo de desarrollo al Scrum Master ni al Propietario del producto. Cada uno tiene tareas asignadas y responsabilidades, siguiendo un proceso de ejecución. Es posible que algunos miembros sean especialistas en áreas concretas, pero la responsabilidad recae sobre el equipo de desarrollo en conjunto. Entonces en el equipo:
 - Todos conocen y comprenden la visión del propietario del producto.
 - Aportan y colaboran con el propietario del producto en el desarrollo de la pila del producto.
 - Comparten de forma conjunta el objetivo de cada sprint y la responsabilidad del logro.
 - Todos los miembros participan en las decisiones.
 - Se respetan las opiniones y aportes de todos.

En la Figura 2.1 se puede ver un gráfico como resumen de los roles expuestos:



Figura 2. 1 Descripción resumida de los Roles en Scrum

Fuente: [Palacio, 2014]

2.3.3. ARTEFACTOS

- **Pila del producto: (Product backlog):** La pila del producto es el inventario de funcionalidades (Historias de Usuario), mejoras y corrección de errores que deben incorporarse al producto a través de los sucesivos sprints. Representa todo aquello que esperan el cliente, los usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo debe estar reflejado en esta pila. Sin embargo, la pila de requisitos del producto nunca se da por completada; esta en continuo crecimiento y evolución. Al comenzar el proyecto incluye los requisitos inicialmente conocidos (generalmente se empieza con una tormenta de ideas o un proceso de exploración donde todo el equipo de desarrollo colabora en base a la visión del

propietario del producto) y mejor entendidos, y conforme avanza el desarrollo, y evoluciona el entorno en el que será usado, se va desarrollando [Palacio, 2014].

Los elementos de la pila del producto que pueden ser incorporados a un sprint se denominan “preparados” o “accionables” y son los que pueden seleccionarse en la reunión de planificación del sprint. La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo. Es recomendable que al menos tenga la siguiente información:

- Identificador Único de la funcionalidad o trabajo.
- Descripción de la funcionalidad/requisito, denominado “historia de usuario”.
- Campo o sistema de Priorización.
- Estimación del esfuerzo necesario.

Dependiendo del proyecto, y organización del equipo, pueden ser aconsejables otros campos más:

- Observaciones
- Persona Asignada
- Nº de Sprint en el que se realiza
- Modulo del sistema al que pertenece

En la Tabla 2.2 podemos ver un ejemplo de una pila de producto con los campos esenciales:

D	Prioridad	Descripción	Estimación
1	Muy Alta	Plataforma Tecnológica	30
2	Muy Alta	Interfaz de Usuario	40
3	Muy Alta	Un Usuario se registra en el sistema	40
4	Alta	El operador define el flujo y textos de un expediente	60
5	Alta	Xxx	999

Tabla 2. 2: Ejemplo de una Pila de Producto

Fuente: [Palacio, 2014]

- **Pila de sprint: (Sprint backlog):** La pila de sprint es la lista que descompone las funcionalidades de la pila del producto (historias de usuario) en tareas necesarias para construir un incremento: una parte completa y operativa del producto.

La realiza el equipo durante la reunión de planificación del sprint, asignando cada tarea a un miembro del equipo, e indicando en la misma lista cuanto tiempo o esfuerzo se prevé que falta para terminarla.

La pila del sprint descompone el trabajo en unidades de tamaño adecuado para monitorizar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos de gestión complejos.

Durante el sprint, el equipo actualiza a diario en ella los tiempos pendientes de cada tarea. Al mismo tiempo, con estos datos se traza el grafico de avance o trabajo consumido [Palacio, 2014].

- **Incremento:** El incremento es la parte del producto producida en un sprint, y tiene como característica el estar completamente terminada y operativa, es decir, en condiciones de ser entregada al cliente.

No se deben considerar como Incremento a propósitos, módulos o submódulos, ni partes pendientes de pruebas o integración. Sin embargo es una excepción frecuente el primer sprint. En el que objetivos del tipo “contrastar la plataforma y el diseño” pueden resultar necesarios, e implican trabajos de diseño o desarrollo de prototipos para contrastar las expectativas de la plataforma o tecnología que se va a emplear [Palacio, 2014].

- **Gráfico de Avance:** También se suele llamar a este gráfico con su nombre inglés: “burn-down”. Lo actualiza el equipo en el Scrum diario, para comprobar el ritmo de avance, y detectar desde el primer momento si es el previsto, o por el contrario se puede ver comprometida o adelantada la entrega prevista al final del sprint. La estrategia ágil para el seguimiento del proyecto se basa en [Palacio, 2014]:
 - Medir el trabajo que falta, no el realizado.
 - Seguimiento cercano del avance (diario de ser posible).

El equipo dispone en la pila del sprint, de lista de tareas que va a realizar, y en cada una se muestra el esfuerzo pendiente. Esto es, el primer día, en la pila de tareas se puede ver para cada tarea el esfuerzo que se ha estimado, puesto que aún no se ha trabajado en ninguna de ellas. Día tras día, cada miembro del equipo actualiza en la pila de sprint el tiempo que le queda a las tareas que va desarrollando, hasta que se terminan, es decir hasta que el tiempo pendiente sea 0 [Palacio, 2014].

La Figura 2.2 muestra un ejemplo de pila en el sexto día del sprint: las tareas terminadas ya no tienen esfuerzo pendiente, y de esfuerzo total previsto para el sprint: 276 puntos (A), para el sexto día quedan 110 (B).

SPRINT			INICIO		DURACIÓN		ESFUERZO													
1	1-mar-07	12	J	V	L	M	X	J	V	L	M	J	V	L	J	V	L			
			1-mar	2-mar	3-mar	4-mar	5-mar	6-mar	7-mar	8-mar	9-mar	10-mar	11-mar	12-mar	13-mar	14-mar	15-mar			
			23	23	19	16	16	13	9	9	9	9	9	9	9	9	9			
			276	246	216	190	178	158	110	110	110	110	110	110	110	110	110			
SPRINT BACKLOG																				
Tarea	Estado	Responsable	16	16	16	16	16	16	12	12	12	12	12	12	12	12	12	12		
Descripción de la tarea 1	Terminada	Luis																		
Descripción de la tarea 2	Terminada	Luis	12	8																
Descripción de la tarea 3	Terminada	Luis	4	4	4	4	4													
Descripción de la tarea 4	Terminada	Elena	8	4																
Descripción de la tarea 5	Terminada	Elena	16	16	4															
Descripción de la tarea 6	Terminada	Elena	6	6	2															
Descripción de la tarea 7	Terminada	Antonio	16	4																
Descripción de la tarea 8	Terminada	Antonio	16	16	20	12	4													
Descripción de la tarea 9	Terminada	Antonio	12	2																
Descripción de la tarea 10	En curso	Luis	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12		
Descripción de la tarea 11	Pendiente	Luis	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8		
Descripción de la tarea 12	Terminada	Luis	14	14	14	14	14	14	14											
Descripción de la tarea 13	En curso	Antonio	8	8	8	8	8	6												
Descripción de la tarea 14	Pendiente	Antonio	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16		
Descripción de la tarea 15	Pendiente	Antonio	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16		
Descripción de la tarea 16	Terminada	Elena	8	8	8															
Descripción de la tarea 17	Terminada	Elena	12	12	12	8	4													
Descripción de la tarea 18	En curso	Elena	15	16	16	16	16	10	10	10	10	10	10	10	10	10	10	10		
Descripción de la tarea 19	Terminada	Elena	12	12	12	12	12	12	12											
Descripción de la tarea 20	Pendiente	Elena	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16		
Descripción de la tarea 21	Pendiente	Elena	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12		
Descripción de la tarea 22	Pendiente	Antonio	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8		
Descripción de la tarea 23	Pendiente	Antonio	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12		

Figura 2. 2: Ejemplo de una Pila de Sprint

Fuente: [Palacio, 2014]



En la Figura 2.3 se puede ver una representación de cómo se actualiza el gráfico de avance en base a la información de la pila de sprint, poniendo cada día el esfuerzo pendiente total de todas las tareas que aún no se han terminado de tal manera refleje el avance del proyecto.

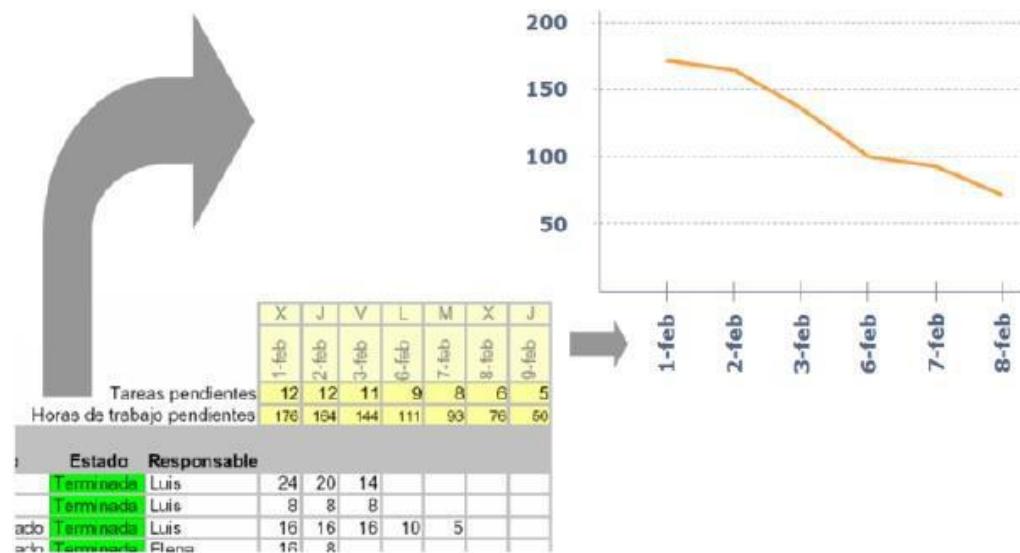


Figura 2. 3: De la Pila del Sprint al Gráfico de Avance.

Fuente: [Palacio, 2014]

En la Figura 2.4 se plasma el avance ideal de un sprint que estaría representado por la diagonal que reduce el esfuerzo pendiente de forma continua y gradual hasta completarlo el día que termina el sprint.

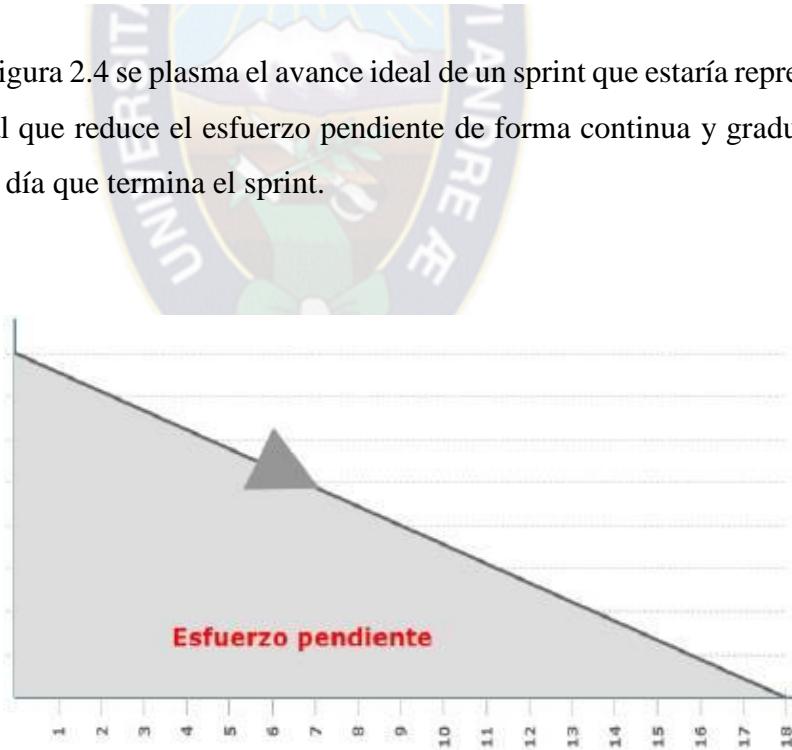


Figura 2. 4: Gráfico de Avance Previsto.

Fuente: [Palacio, 2014]

Las gráficas de la diagonal perfecta no son la habitual, en la Figura 2.5 se ve un ejemplo de un patrón de avance más normal.

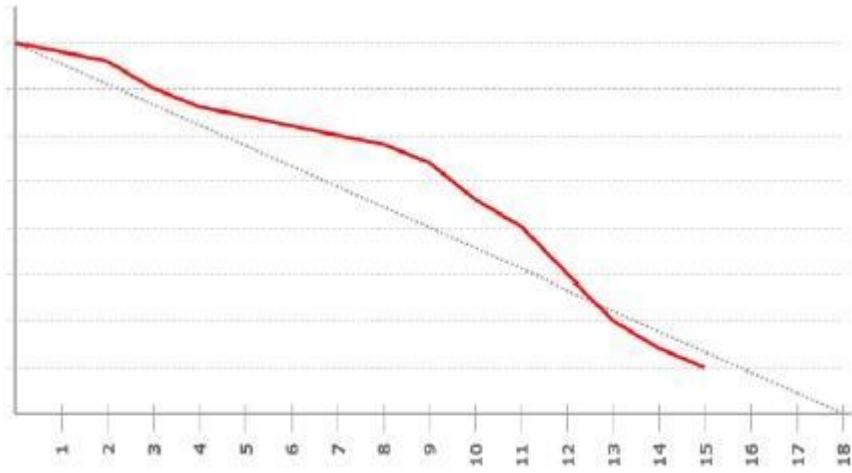


Figura 2. 5: Gráfico de Avance Real

Fuente: [Palacio, 2014]

El gráfico de la Figura 2.6 sería el aspecto de la gráfica de un sprint subestimado.



Figura 2. 6: Gráfica de Avance Subestimado

Fuente: [Palacio, 2014]

La estimación que realizó el equipo en la reunión de inicio del sprint es inferior al esfuerzo real que están requiriendo las tareas. Y esto se reflejaría en el patrón de su gráfica de un sprint sobreestimado como se puede ver en la Figura 2.7.

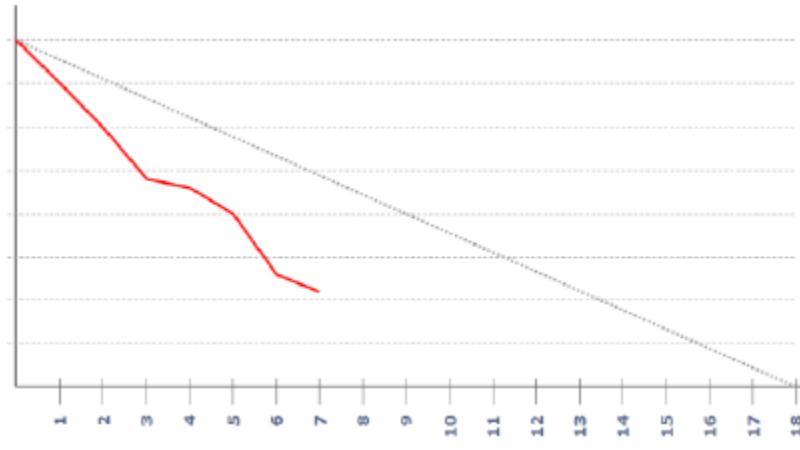


Figura 2. 7: Gráfica de Avance Sobreestimado

Fuente: [Palacio, 2014]

En la Figura 2.8 podemos ver un resumen grafico de los artefactos que se usan en la metodología Scrum.

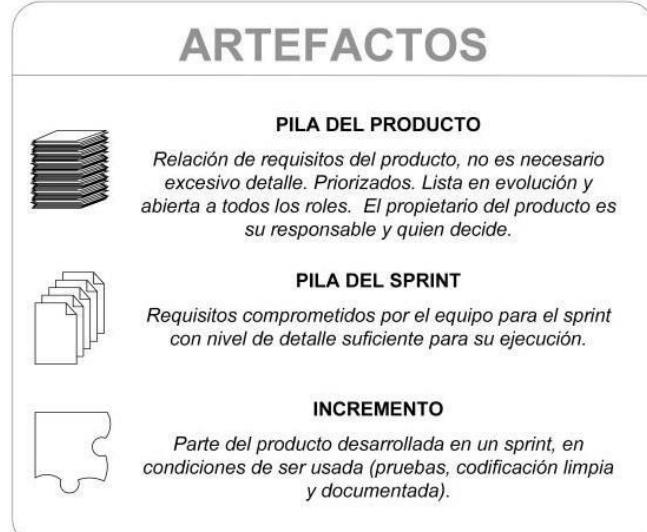


Figura 2. 8: Descripción resumida de los Artefactos Scrum

Fuente: [Palacio, 2014]

2.3.4. FLUJO DE TRABAJO

- **Planificación del Proyecto:** También es conocido como la preparación del proyecto, que corresponde a la definición de las historias de usuario para la pila del producto en base a los requerimientos y necesidades del cliente, y la determinación del tamaño y la duración de los sprints y cuántos de estos se realizarán, además conjuntamente también se realizan las actividades preparatorias de análisis, diseño, selección de las herramientas tecnológicas, identificación de políticas de calidad y seguridad para el sistema a desarrollar, y si es necesario la capacitación del equipo de desarrollo. Deben estar presentes el Propietario del Producto, el Scrum Master, el equipo de desarrollo, los interesados y dura de una a dos semanas [Menzinsky, 2015].
- **Reunión de planificación de Sprint:** En esta reunión se toman como base las prioridades y necesidades de negocio del cliente, y se determinan cuáles y cómo van a ser las funcionalidades que se incorporarán al producto en el siguiente sprint [Palacio, 2014].

El responsable de que se realice esta reunión es el Scrum Master y debe asegurarse de que estén presentes el propietario del producto, el equipo de desarrollo y si es necesario los implicados o interesados del proyecto. La reunión de planificación de Sprint dura aproximadamente una jornada de trabajo completa, cuando se trata de planificar un sprint largo (de un mes de duración) o un tiempo proporcional para planificar un sprint más breve [Palacio, 2014].

El propietario del producto presenta la pila de producto, exponiendo los requisitos de mayor prioridad que se necesita y prevé que se podrán desarrollar en el siguiente sprint. Si algún elemento de la pila del producto ha tenido cambios

significativos desde la anterior reunión, explica las causas que los han ocasionado [Palacio, 2014].

El equipo de desarrollo hace preguntas y pide aclaraciones necesarias, además pueden proponer sugerencias, modificaciones y soluciones alternativas, como respuesta el propietario del producto atiende las dudas y sugerencias del equipo asegurándose de que tengan un buen entendimiento de la idea y visión del producto.

El equipo desglosa cada funcionalidad en tareas, y estiman el tiempo para cada una de ellas, auto asignándose en base a sus conocimientos, intereses y una distribución homogénea de trabajo, componiendo así las tareas que forman la pila de sprint [Palacio, 2014].

- **Sprint:** Se denomina sprint a cada ciclo o iteración de trabajo que produce una parte del producto terminada y funcionalmente operativa (incremento). La duración de cada sprint puede ser desde una hasta seis semanas, aunque se recomienda que no excedan de un mes. El equipo monitoriza la evolución de cada sprint en reuniones breves diarias (reuniones de pie – stand-up meeting) que duran entre 5 a 15 minutos donde se revisa en conjunto las tareas realizadas por cada miembro el día anterior, y previsto para el día en curso [Palacio, 2014].
- **Reunión Diaria:** Las reuniones diarias, no debe durar más de 15 minutos, deben estar presentes todo el equipo.

En esta reunión cada miembro del equipo explica lo que ha logrado desde la anterior reunión diaria, lo que va a hacer en la jornada laboral actual y si está teniendo algún problema, o si prevé que puede encontrar algún impedimento, para que los demás miembros del equipo puedan colaborarle en solucionarlo [Palacio, 2014].

- **Revisión del sprint:** Se lo realiza al finalizar un sprint para comprobar el incremento. No debe durar más de 4 horas en caso de sprints largos. Para sprints de 1 o 3 semanas, de 1 a 2 horas es suficiente y deben estar presentes el equipo de desarrollo, el propietario del producto y el Scrum Master. El equipo expone el objetivo del sprint, las listas de funcionalidades deseadas, y las que se han desarrollado, haciendo una demostración de las partes construidas. Luego se hace una ronda de preguntas y sugerencias, donde se genera información valiosa respecto a la visión del producto. El propietario del producto comprobando el progreso del sistema, ve si se puede considerar si se ha logrado el objetivo y el Scrum Master, de acuerdo a las agendas del propietario del producto y el equipo, cierra la reunión para la preparación del siguiente sprint [Palacio, 2014].
- **Retrospectiva del sprint:** Se los realiza tras la revisión de cada sprint, y antes de la reunión de planificación del siguiente sprint, puede durar de 1 a 3 horas, según la duración del sprint terminado. En ella el equipo realiza un autoanálisis de su forma de trabajar, identificando fortalezas para consolidarlos y puntos débiles para tomar acciones de mejoras [Palacio, 2014].

En la Figura 2.9 un resumen gráfico de los eventos Scrum que ya fueron expuestos.



Figura 2. 9: Descripción Resumida de los Eventos Scrum

Fuente: [Palacio, 2014]

Por último, en la Figura 2.10 se puede ver un resumen del flujo de trabajo del marco técnico Scrum.

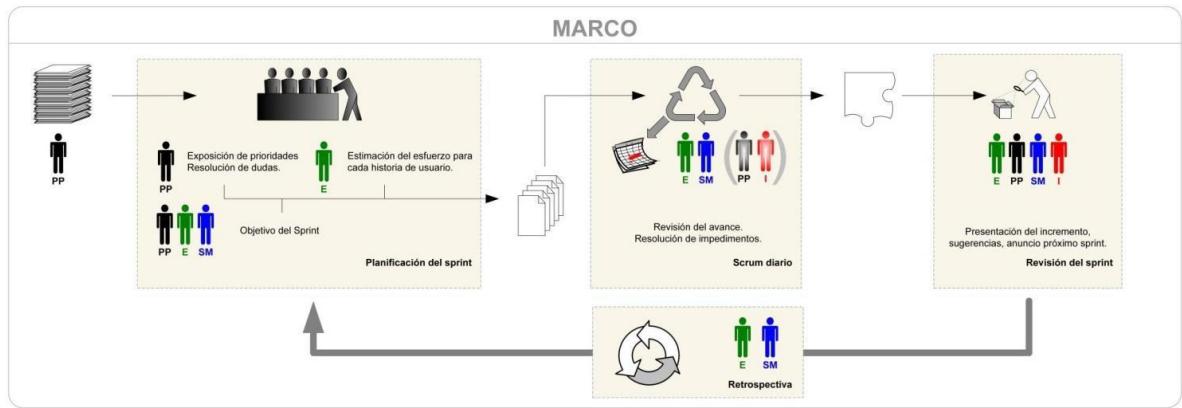


Figura 2. 10: Flujo de trabajo del Marco Técnico Scrum

Fuente: [Palacio, 2014]

2.3.5. MODELO DE PROCESO

Scrum es un método de desarrollo muy simple, que requiere mayor trabajo ya que no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. El ciclo de vida de Scrum está compuesto por tres tareas o etapas, Pre-Game, Game y Post-Game.

2.3.5.1 PRE – GAME

Antes de llevar a cabo el desarrollo del proyecto, se especifica lo que se va a realizar en las iteraciones, además de la prioridad con la que se lo hará, ésta fase consta de dos puntos importantes:

- **Planeación.** Durante la planeación todos los miembros del equipo incluyendo el cliente contribuyen a la creación de una lista de características del software, para el análisis y conceptualización del problema

Las tareas que se realizan en ésta primera etapa son:

- Recopilación de requerimientos para conformar el backlog del producto, priorizados de acuerdo a una evaluación del cliente.
 - Definición de las fechas de entrega de los sprints y sus funcionalidades.
 - Análisis de riesgos y controles apropiados para los riesgos.
 - Selección de las herramientas y de la infraestructura de desarrollo.
 - Cálculo y estimación del costo de cada iteración.
- **Arquitectura.** El objetivo de esta etapa es diseñar el cómo los elementos del backlog del producto serán puestos en ejecución. Esta fase incluye una revisión de la arquitectura del software y diseño de alto nivel.

En esta etapa se realizan las tareas de:

- Revisión de los ítems del backlog del producto
- Análisis de dominio para reflejar el nuevo contexto y requisitos del sistema.

- Revisión de la arquitectura del software de acuerdo a los requisitos definidos.
- Revisión del diseño de alto nivel.

2.3.5.2 GAME

Una vez realizada la especificación correspondiente, se lleva a cabo la elaboración del proyecto con un continuo seguimiento a cargo del mismo grupo de desarrollo.

En cada iteración del Game se realizan las siguientes tareas:

- **Planeación del Sprint.** Antes de comenzar cada Sprint o Iteración, se lleva a cabo dos reuniones consecutivas, en la primera se mejora y se prioriza nuevamente el backlog del producto, además de elegir las metas de la iteración. En la segunda reunión se deben considerar el cómo alcanzar los requerimientos y crear el backlog del Sprint.
- **Desarrollo del Sprint.** El trabajo generalmente se organiza en iteraciones de 30 días (Sprints). El Sprint es el desarrollo de la nueva funcionalidad para el producto. Esta fase provee la siguiente documentación. Backlog del Sprint con las actividades desarrolladas, los responsables y la duración de cada actividad.
- **Revisión del Sprint.** Al final de cada iteración se lleva a cabo una reunión de revisión, donde se presenta la nueva funcionalidad del software, las metas e inclusive la información de las funciones, diseño, ventajas, inconvenientes y esfuerzos del equipo.

2.3.5.3 POST – GAME

Luego de haber culminado todas las iteraciones, solo resta la revisión final, denominada según la metodología Scrum, el cierre.

- **Cierre.** En ésta última etapa se realiza la preparación operacional, incluyendo la documentación final necesaria para la presentación. También en esta etapa se debe realizar dependiendo del tipo del producto, ya sea el entretenimiento del usuario o el marketing para la venta del nuevo software.

2.4. INGENIERÍA WEB

La ingeniería Web es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.

El proceso de Ingeniería Web está relacionado con la inmediatez y evolución, estos crecen continuamente para conllevar un proceso incremental y evolutivo, que permite al usuario involucrarse activamente, facilitando el desarrollo de productos que se ajustan a lo que se está buscando en cuanto a la funcionalidad del producto.

El proceso de desarrollo de aplicaciones hipermedia con lleva a la realización de una serie de actividades que son aplicables a cualquier Sistema Web, independientemente del tamaño y complejidad del mismo, la Ingeniería Web aplica el desarrollo, operación y mantenimiento de todo el ciclo de vida del desarrollo de Software.

Es importante mencionar que el impacto de los sistemas y aplicaciones basados en metodologías Web es el suceso más significativo en la era de las Tecnologías de Información

y Comunicación. Conforme a la importancia de las Aplicaciones Web que crece, surge la necesidad de enfocar la ingeniería Web a un entorno disciplinado y bien estructurado.

Visualizando la evolución para las personas que están en el área de la informática, esta es significativamente grande y en ocasiones confusa; pasamos de páginas estáticas hechas con HTML, Java Script, CMS y estándares como XHTML, CSS, P3P por nombrar algunas de ellas.

Todo enfocado hacia el usuario final, a su interacción y producción. La Web crece a medida que los usuarios colaboran usando estas plataformas tecnológicas.

Esta evolución tecnológica es lo que muchos llaman ahora la Web 2.0; no es una versión, sino un concepto y una forma diferente de ver las cosas [Escalona, 2002].

2.5. DESARROLLO WEB BASADA EN U. M. L. (U. W. E.)

2.5.1. LENGUAJE UNIFICADO DE MODELADO (U.M.L.)

UML (Unified Modeling Language) es un lenguaje gráfico para visualización, especificación, construcción y documentación de componentes de sistemas grandes y complejos, como también para modelar negocios y otros sistemas que no son de software [Pérez, 2010].

Como lenguaje de especificación, significa la construcción de modelos precisos, completos y no ambiguos. En particular, maneja la especificación de todo el análisis, diseño y decisiones de implementación que puede hacerse en la etapa de desarrollo e implantación de un sistema grande y complejo de software [Pérez, 2010].

Como lenguaje de documentación, maneja la documentación de la arquitectura de un sistema y todos los detalles. Introduce diagramas que representan la parte dinámica de los procesos, logrando de esta manera identificar fallas de diseño en los procesos y por consiguiente generadoras de errores. Permite estereotipar sus elementos para que tengan un comportamiento particular [Pérez, 2010].

Para entender esta metodología, se necesita construir un modelo conceptual del lenguaje y este requiere tres elementos especiales: construcción de bloques en UML, las reglas que indican como esos bloques construidos se pueden interrelacionar, y algunos mecanismos comunes que se aplican en todo el lenguaje [Pérez, 2010].

La construcción de bloques en el UML se divide en tres categorías que son: cosas, relaciones y los diagramas. UML proporciona un gran número de diagramas, algunos son más necesarios dependiendo el tipo proyecto [Pérez, 2010].

2.5.2. INGENIERÍA WEB BASADA EN U.M.L. (U.W.E.)

U.W.E. es una metodología que permite especificar de mejor manera una aplicación web, para el proceso de creación de aplicaciones y describe que es lo que se debe utilizar. Este proceso iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el proceso unificado de modelado [Pérez, 2010].

La metodología U.W.E. define vistas especiales representadas gráficamente por diagrama en UML. Los principales aspectos en los que se fundamenta U.W.E. son los siguientes:

- Notación estándar: el uso de la metodología U.M.L. para todos los modelos.

- Métodos definidos: pasos definidos para la construcción de cada modelo.
- Especificación de restricciones: recomendables de manera escrita, para que la exactitud en cada modelo aumente.

2.5.2.1 FASES DE DESARROLLO

Las actividades base de modelado de U.W.E. son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como son el modelo de usuario, modelo de adaptación y modelo de tareas para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. De esta manera se obtiene una colección de modelos y diagramas que describen una aplicación web de manera integral [Pérez, 2010].

- **Análisis de requisitos:** plasma los requerimientos funcionales de la aplicación web, mediante modelos de casos de uso.
- **Diseño conceptual:** se define mediante un modelo de dominio, considerando los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle.
- **Diseño navegacional:** Comprende la construcción del modelo de navegación en dos pasos:
 - **Modelo de espacio de navegación:** su objetivo es especificar qué objetos pueden ser visitados a través de la aplicación.

- **Modelo de estructura de navegación:** amplía el modelo con un modelo de estructuras de acceso necesarias para la navegación como los índices, consultas y visitas guiadas.
- **Diseño de presentación:** permite la especificación lógica de la aplicación web. Basada sobre este modelo lógico, una presentación física puede ser construida. Representa las interfaces de usuario por medio de vistas estándares de interacción U.M.L. Dentro de este modelo se distinguen dos diferentes vistas:
 - **Estructura de vista:** muestra la estructura del espacio de presentación.
 - **Interfaz de usuario (UI por sus siglas en inglés de User Interface):** Se refleja los detalles acerca de los elementos de la interfaz de usuario dentro de las páginas.

2.5.2.2 ANÁLISIS DE REQUISITOS

El análisis de requisitos se expresa a través de la especificación de los casos de uso del sistema. Un caso de uso en UML es una unidad coherente de la funcionalidad proporcionada para la aplicación que obra recíprocamente con unos o más actores de la aplicación. Describe una parte del comportamiento de la aplicación sin revelar la estructura interna. [Novilos, Coronel & Campaña, 2010]

De esta manera, los requisitos para una aplicación web se pueden especificar con un modelo de casos de uso. Un ejemplo de Casos de Uso se puede ver en la Figura 2.11.

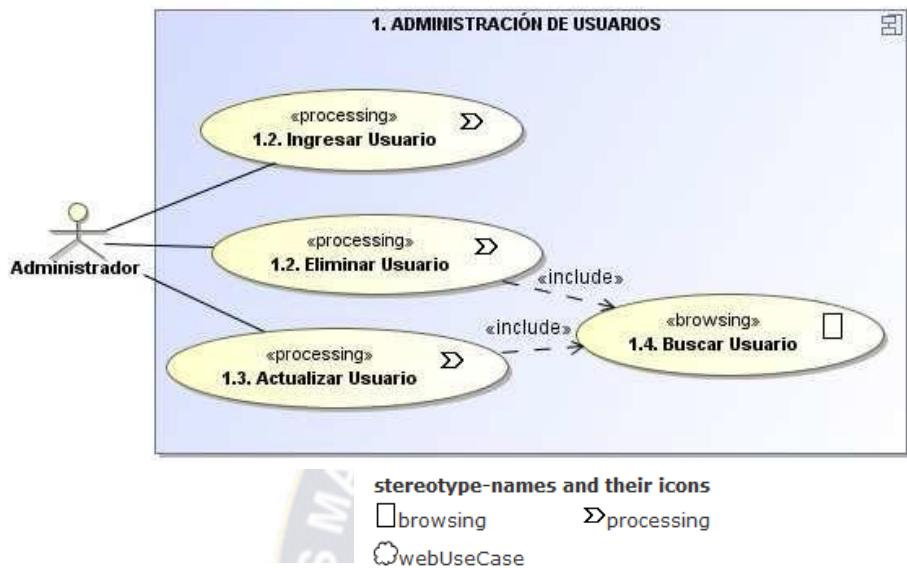


Figura 2. 11: Ejemplo de un Diagrama de Casos de Uso.

Fuente: [Novilos, Coronel & Campaña, 2010]

2.5.2.3 DISEÑO CONCEPTUAL

El modelado conceptual para aplicaciones web dentro de UWE no difiere del modelado conceptual de las aplicaciones normales. Se utiliza un diagrama de clases para representar gráficamente un modelo conceptual como visión estática que demuestre una colección de los elementos estáticos del dominio [Nolivos, Coronel, & Campaña, 2010].

La construcción del modelo conceptual se lleva a cabo de acuerdo con los casos de uso que se definieron en la especificación de los requerimientos de usuario. En la Figura 2.12 podemos ver un ejemplo de un modelo conceptual con diagrama de clases.

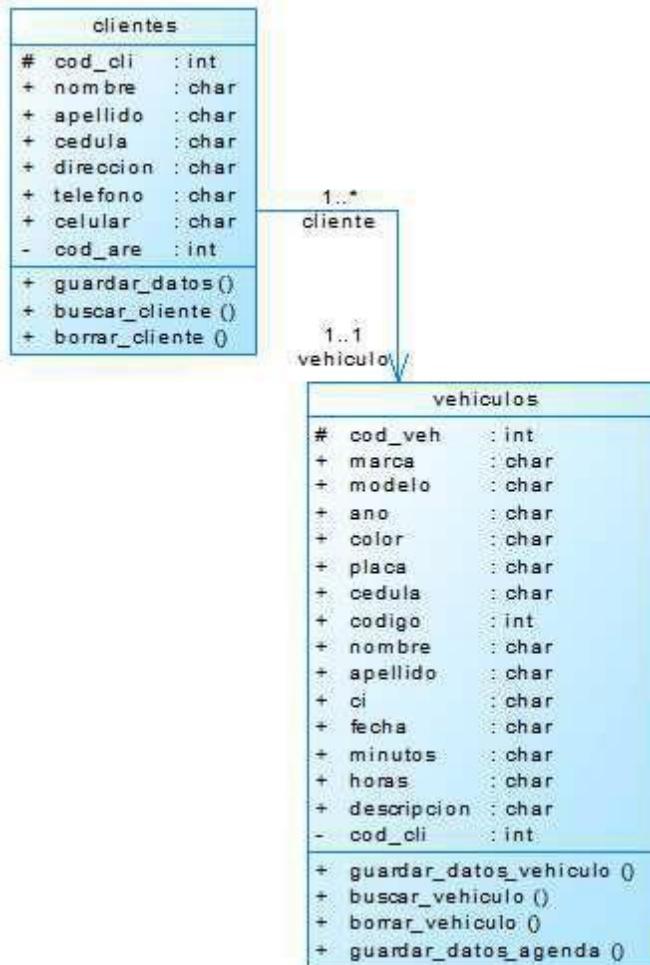


Figura 2. 12: Ejemplo de un modelo Conceptual – Diagrama de Clases

Fuente: [Novilos, Coronel & Campaña, 2010]

2.5.2.4 DISEÑO NAVEGACIONAL

El modelo de navegación de una aplicación web comprende la especificación de qué objetos pueden ser visitados mediante la navegación, a través del sistema y las asociaciones entre ellos. Mediante estos diagramas se representa el diseño y la estructura de las rutas de

navegación al usuario para evitar la desorientación en el proceso de navegación. Los elementos básicos en el modelo de navegación son los nodos y enlaces como se puede apreciar en la Figura 2.13 [Nolivos, Coronel & Campaña, 2010].

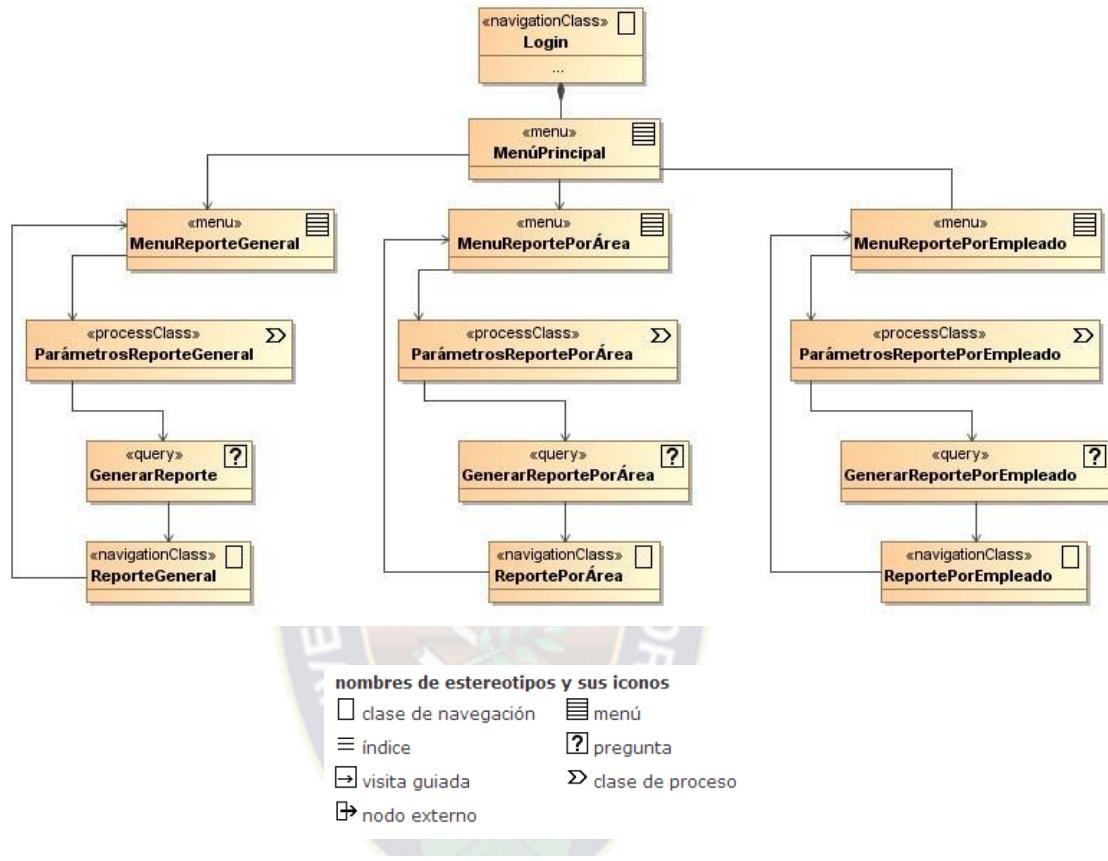


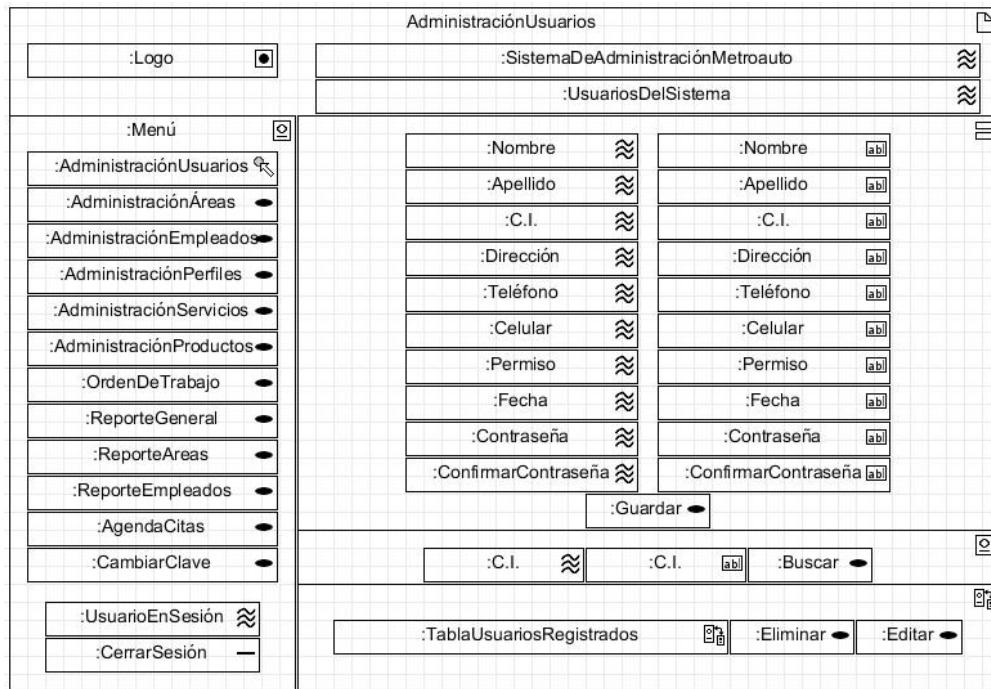
Figura 2. 13: Ejemplo de un modelo de Navegación.

Fuente: [Novilos, Coronel & Campaña, 2010]

2.5.2.5 DISEÑO DE PRESENTACIÓN

El modelo de presentación en U.W.E. está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos

de UML. Los elementos del modelo de presentación son: ventanas, entradas de texto, imágenes, audio y botones [Nolivos, Coronel, & Campaña, 2010]. Como se puede ver en la Figura 2.14, estos diagramas permiten especificar dónde y cómo los objetos de navegación serán presentados al usuario, una representación esquemática visible para el usuario.



nombres de estereotipos y sus iconos

□ grupo de presentación	□ página de presentación
≋ texto	abl entrada de texto
— ancla	fileUpload
● botón	○ imagen
□ formulario	□ componente de cliente
□ alternativas de presentación	mouse cursor selección

Figura 2. 14: Ejemplo de un Modelo de Presentación.

Fuente: [Novilos, Coronel & Campaña, 2010]

CAPÍTULO III

MARCO APLICATIVO

3.1. INTRODUCCIÓN

En este capítulo se describe la planificación y el desarrollo del sistema web de administración para evaluación docente y cursos de temporada SAURON de la Carrera de Ingeniería Industrial. aplicando la metodología Scrum y complementando con la metodología U.W.E.

Para tal efecto el proyecto se dividirá en tres grandes módulos, con una iteración para cada una:

- Módulo central, que concentrará la administración de información base.
- Módulo Curso de temporada, que se centrará en su propia administración y solicitud de información para su funcionamiento.
- Módulo Evaluación Docente, que se centrará en su propia administración y solicitud de información.

Será utilizado el modelo de procesos de Scrum, teniendo la siguiente refactorización con UWE:

- Pre – game, donde será realizada la recopilación de requerimientos, y fortalecida con el modelado de casos de uso que nos proporciona UWE, además del cronograma y el análisis de riesgos del proyecto.

- Game, en este apartado serán realizadas las iteraciones donde se desarrollarán los módulos mencionados anteriormente y cada una de ellas contendrá el modelo de navegación y de presentación, únicamente en la primera iteración consideraremos al modelo conceptual.
- Post-game, ya realizaremos pruebas al software y su documentación.

Previamente a lo mencionado identificaremos a los actores del proyecto en Scrum.

3.1.1. IDENTIFICACIÓN DE ROLES SCRUM

En la tabla 3.1 se tiene una lista de las personas implicadas e identificadas con sus roles de Scrum para el desarrollo del proyecto.

ROL		NOMBRE
Propietario del Producto		Ing. Oswaldo Teran
Scrum Master		Israel Huallpara
Equipo Scrum	Analista Diseñador Desarrollador Tester	Israel Huallpara

Tabla 3. 1: Identificación de Roles Scrum

Fuente: [Elaboración propia]

3.1.2. NÚMERO DE ITERACIONES

De acuerdo a los requerimientos institucionales evidenciados en la Tabla 3.2 se manejaron 3 iteraciones que se observan con más detalle en el Game del proyecto.

- Primera iteración. Administración de datos generales del sistema (Módulo Central).
- Segunda iteración. Administración del curso de temporada (Módulo Cursos de temporada).
- Tercera iteración: Administración de la evaluación docente (Módulo evaluación docente).

3.2. PRE – GAME

3.2.1. RECOLGIDA DE REQUERIMIENTOS

Se estiman las prioridades de los módulos a desarrollar en relación a los requerimientos de los usuarios.

Un Backlog Eficiente es aquel donde se puede obtener el mayor beneficio con el menor esfuerzo posible.

Este concepto llevado al ProductBacklog significa invertir el esfuerzo de exploración y especificación de la manera más inteligente posible para evitar re-trabajos y desperdicios innecesarios.

Por esto se recomienda un ProductBacklog donde sus ítems más prioritarios están expresados con un nivel de detalle mucho mayor que los ítems de menor prioridad, los cuales estarán descriptos a un nivel más alto, ya que son los más susceptibles de ser alterados o reemplazados.

Por todo aquello, a continuación, se menciona algunas características del flujo de trabajo en relación a los cursos de temporada y evaluación docente.

La carrera de Ingeniería Industrial se ve en la obligación de organizar y agilizar el proceso de habilitación de materias y la inscripción a las mismas para los dos cursos de temporada anuales que son publicados.

La inscripción actual es mediante Kardex que se encarga de toda la administración de los cursos, se evidenció que además el proceso de pre inscripción es manual, y por ende acarrea retraso en la atención y desinformación sobre todo en la población estudiantil.

A finales de cada gestión es realizada la evaluación docente, que se pudo evidenciar que es centralizado mediante hojas de cálculo donde para realizar la validación y ponderación por el tribunal estos son nuevamente cargados acarreando posibles errores de transcripción.

Por lo expuesto anteriormente se llega a la Tabla 3.2 donde son centralizados los requerimientos establecidos y solicitados por los usuarios mediante reuniones en cada sprint.

ID	DESCRIPCIÓN	MÓDULO	PRIORIDAD	ESTADO
R1	El usuario puede restaurar su contraseña por correo electrónico.	Central	Media	Terminado
R2	El administrador consulta, crea, publica, edita, y elimina cursos de temporada.	Temporada	Alta	Terminado
R3	El administrador habilita materias para inscripción.	Temporada	Alta	Terminado

R4	El administrador y estudiante habilita materias para inscripción en base a resultados de preinscripción.	Temporada	Alta	Terminado
R5	El usuario cuenta con control y acceso seguro.	Central	Alta	Terminado
R6	El estudiante debe de registrar su inscripción	Temporada	Alta	Terminado
R7	El administrador debe de validar y/o registrar la inscripción.	Temporada	Alta	Terminado
R8	El administrador genera el reporte de cantidad de preinscritos por materia.	Temporada	Media	Terminado
R9	El administrador genera reporte de inscritos por materia.	Temporada	Media	Terminado
R10	El docente registra su informe memoria.	Evaluación	Alta	Terminado
R11	El director de carrera pondera a cada docente titular	Evaluación	Alta	Terminado
R12	El tribunal ponderación y/u observación el informe memoria registrado por el docente.	Evaluación	Alta	Terminado
R13	El tribunal genera reportes finales de ponderaciones alcanzadas.	Evaluación	Alta	Terminado
R14	El administrador asigna tribunal.	Evaluación	Media	Terminado
R15	El docente titular visualiza ponderación y observaciones del tribunal	Evaluación	Media	Terminado

Tabla 3. 2: Requerimientos Institucionales

Fuente: [Elaboración Propia]

3.2.2. MODELADO DE CASOS DE USO

Realizando la refactorización entre las metodologías SCRUM y UWE, se genera:

En la Figura 3.1 se puede ver el diagrama de Caso de Uso de Alto Nivel del sistema a desarrollar:

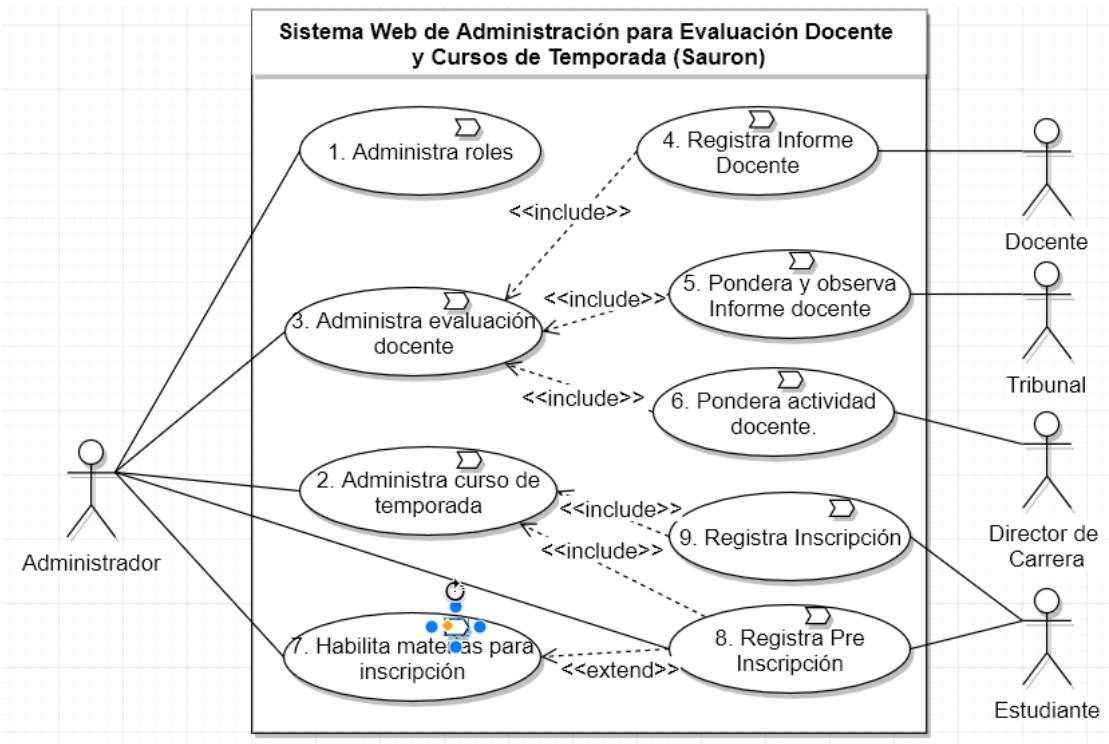


Figura 3. 1: Diagrama de Casos de Uso de Alto nivel del Sistema

Fuente: [Elaboración Propia]

Se muestra en el caso de uso para la Administración de roles en la Figura 3.2 y en la Tabla 3.3 sus especificaciones.

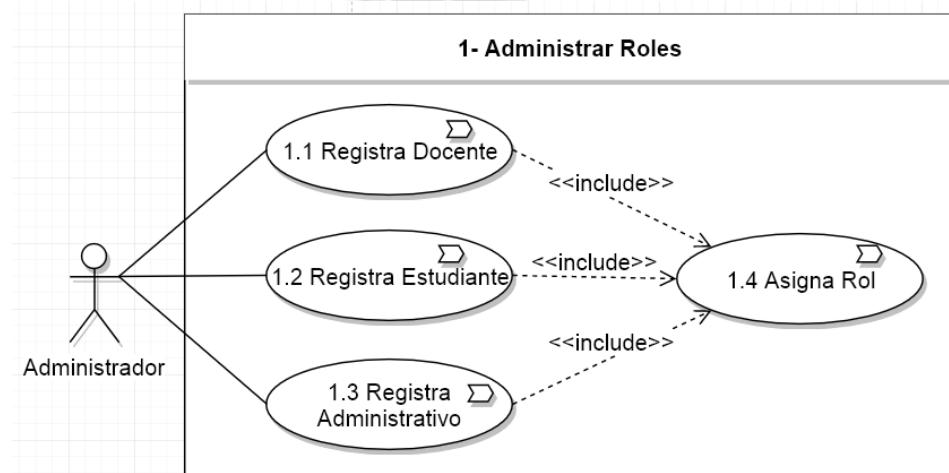


Figura 3. 2: Caso de Uso Administrar roles

Fuente: [Elaboración Propia]

A continuación, en la tabla 3.3 se describe en detalle el caso de uso administrar roles.

Caso de Uso: Administrar roles
Actores: Administrativo
Descripción: Permite crear, editar y eliminar usuarios administrativo, docente y estudiante con su respectivo rol para el acceso al sistema.
Precondiciones: El usuario no puede estar registrado previamente en el sistema.
Flujo normal: 1. El actor ingresa al módulo de docente, estudiante o administrativo. 2. Llena un nuevo registro en el formulario. 3. El actor guarda el registro.
Postcondición: El sistema automáticamente asigna al nuevo registro con el mismo rol que del módulo ingresado, despliega un mensaje y lo muestra en la lista del módulo.

Tabla 3. 3: Descripción del Caso de Uso “Administra Roles”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.3 se muestra el diagrama de caso de uso para la administración del curso de temporada junto con sus especificaciones en la tabla 3.4, que detallan el comportamiento para el módulo del curso de temporada.

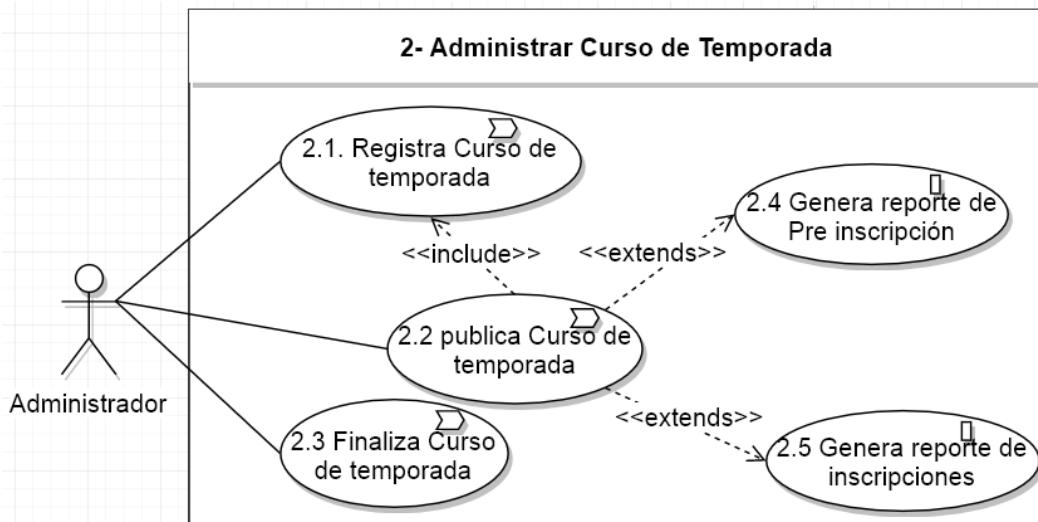


Figura 3. 3: Caso de Uso Administrar Curso de Temporada

Fuente: [Elaboración Propia]

En la tabla siguiente es reflejado en detalle el caso de uso administrar curso de temporada

Caso de Uso:
Administrar Curso de temporada
Actores:
Administrativo
Descripción:
Permite crear, publicar y finalizar cursos de temporada, además la generación de reportes de pre inscripción e inscripción.
Precondiciones:
El curso de temporada no debe de estar registrado previamente.
Flujo normal:
1. El actor ingresa al módulo y registra el nuevo curso de temporada.

- | |
|---|
| <p>1.1. El curso puede ser publicado presionando un botón.</p> <p>1.1.1 Se encuentra disponible para registros de pre inscripción e inscripción.</p> <p>1.1.2 Se puede generar el reporte de pre inscripciones.</p> <p>1.1.3 Se puede generar el reporte de inscripciones.</p> <p>1.2 El curso puede ser finalizado presionando un botón.</p> <p>1.2.1 Ya no se encuentra disponible para nuevos registros.</p> |
|---|

Postcondición:

El sistema al crear un curso es generado en estado borrador, al publicarlo es visible para los estudiantes, dependiendo las fechas registradas se habilitará el módulo de pre inscripción e inscripción.

Tabla 3. 4: Descripción del Caso de Uso “Administra Cursos de temporada”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.4 se muestra el diagrama de caso de uso para la administración de la evaluación docente junto con sus especificaciones en la tabla 3.5.

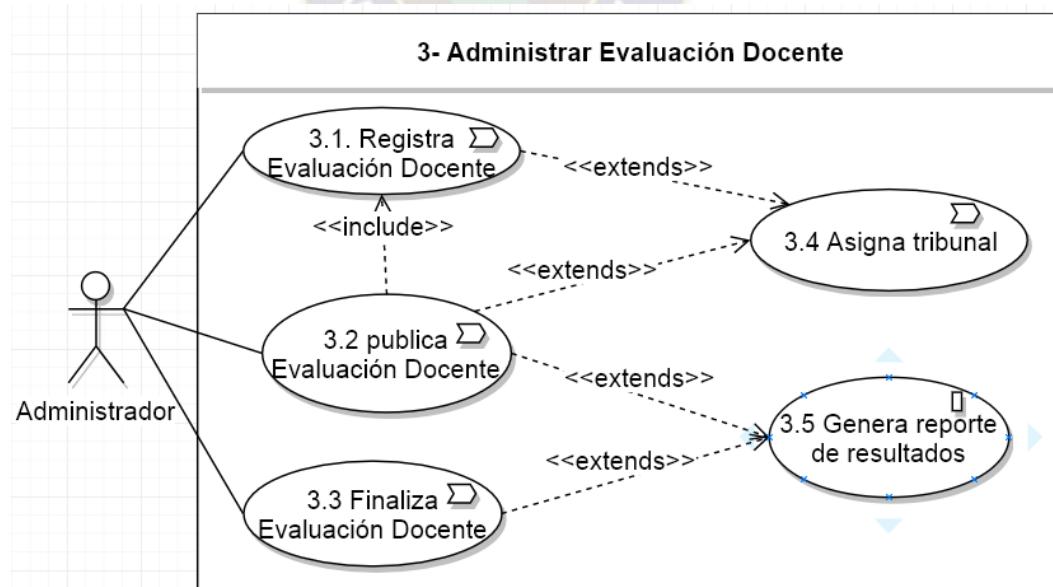


Figura 3. 4: Caso de Uso Administrar Evaluación Docente

Fuente: [Elaboración Propia]

A continuación, la tabla refleja en detalle el caso de uso administrar evaluación docente.

Caso de Uso: Administrar Evaluación docente
Actores: Administrativo
Descripción: Permite crear, publicar y finalizar evaluaciones docentes, además la asignación del tribunal y generación del reporte de resultados.
Precondiciones: La evaluación docente no debe de estar registrada previamente.
Flujo normal: 1. El actor ingresa al módulo y registra la nueva evaluación docente. 1.1 La evaluación puede ser publicada presionando un botón. 1.1.1 Es habilitado el registro de tribunales. 1.1.2 Es habilitada la generación de reportes. 1.2 La revaluación puede ser finalizada presionando un botón. 1.3 Ya no se encuentra disponible para nuevos registros.
Postcondición: El sistema al crear una evaluación es generado en estado borrador, al publicarlo es visible para los docentes, tribunales y director de carrera.

Tabla 3. 5: Descripción del Caso de Uso “Administra Evaluación Docente”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.5 se muestra el diagrama de caso de uso para el registro del informe docente junto con sus especificaciones en la tabla 3.6.

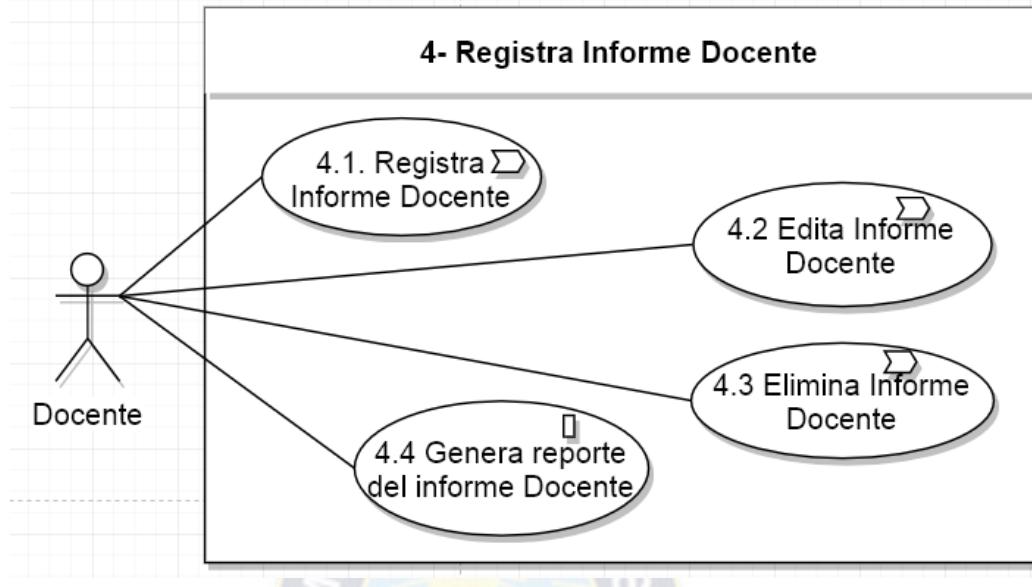


Figura 3. 5: Caso de Uso Registra Informe Docente

Fuente: [Elaboración Propia]

A continuación, la tabla que refleja a detalle el caso de uso registra informe docente.

Caso de Uso:
Registra informe docente
Actores:
Docente
Descripción:
Permite crear, editar y eliminar registros para el informe docente y la generación del reporte.
Precondiciones:
El actor debe tener como tipo de contrato docente titular.
Flujo normal:

1. El actor ingresa al módulo y registra sus actividades en el formulario dependiendo la solicitud, que debe estar basada en el informe docente.
2. Cada registro insertado puede ser editado como eliminado presionando un botón.
3. En cualquier estado del informe puede ser generado el reporte.

Postcondición:

Los registros son vistos únicamente por el actor y el tribunal asignado.

Tabla 3. 6: Descripción del Caso de Uso “Registra Informe Docente”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.6 se muestra el diagrama de caso de uso para la ponderación y observación del informe docente junto con sus especificaciones en la tabla 3.7.

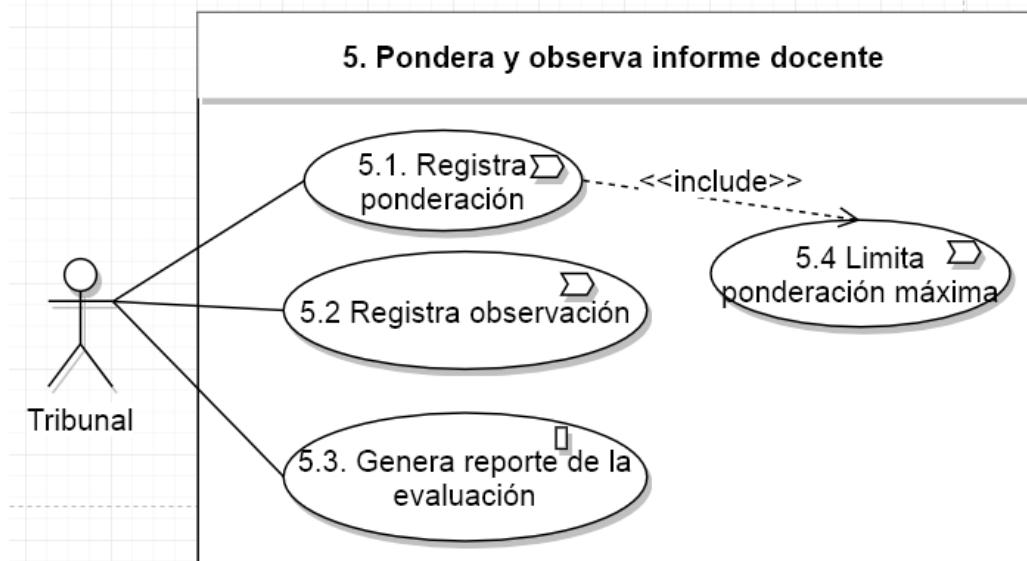


Figura 3. 6: Caso de Uso Pondera y observa informe docente

Fuente: [Elaboración Propia]

A continuación, la tabla describe con detalle el caso de uso de la Figura 3.6.

Caso de Uso:
Pondera y observa informe docente
Actores:
Tribunal
Descripción:
Permite ponderar y observar actividades del informe docente.
Precondiciones:
El actor debe de estar asignado como tribunal.
Flujo normal:
<ol style="list-style-type: none"> 1. El actor ingresa al módulo y por cada sección se encuentra el recuadro para registrar la ponderación, que es limitada mediante la categoría del docente a ser evaluado. 2. Para cada registro del docente se encuentra el recuadro para ingresar la observación. 1A, Si no existen registros en alguna sección la ponderación es automática a cero.
Postcondición:
Las observaciones y ponderaciones son vistas por el docente evaluado.

Tabla 3. 7: Descripción del Caso de Uso “Pondera y observa informe docente”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.7 se muestra el diagrama de caso de uso para la ponderación de la actividad docente junto con sus especificaciones en la tabla 3.8.

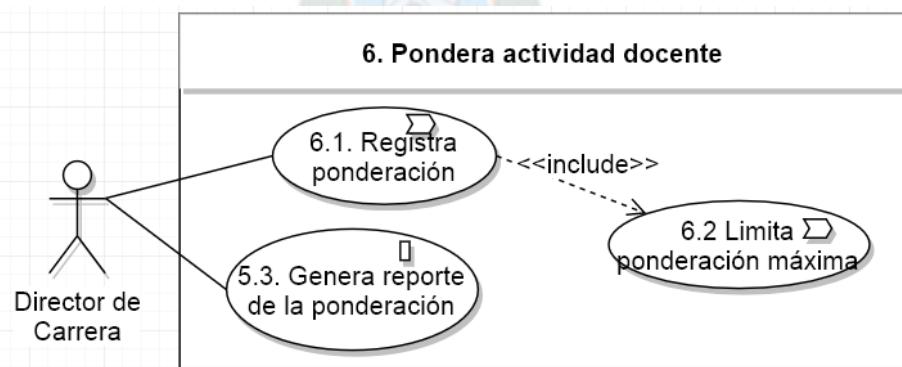


Figura 3. 7: Caso de Uso Pondera actividad docente

Fuente: [Elaboración Propia]

Caso de Uso:
Pondera actividad docente
Actores:
Director de carrera
Descripción:
Permite ponderar la actividad docente por parte del actor y generar su reporte.
Precondiciones:
El actor debe de estar con tipo de contrato director de Carrera.
Flujo normal:
<ol style="list-style-type: none"> 1. El actor ingresa al módulo y selecciona al docente a calificar. 2. Cada consulta es limitada mediante la nota máxima por la categoría del docente evaluado. 3. Al concluir con la calificación de cada consulta, es habilitado el botón de generación de reporte.
Postcondición:
La ponderación es visible para el docente y el tribunal.

Tabla 3. 8: Descripción del Caso de Uso “Pondera y observa informe docente”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.8 se muestra el diagrama de caso de uso para la habilitación de materias junto con sus especificaciones en la tabla 3.9.

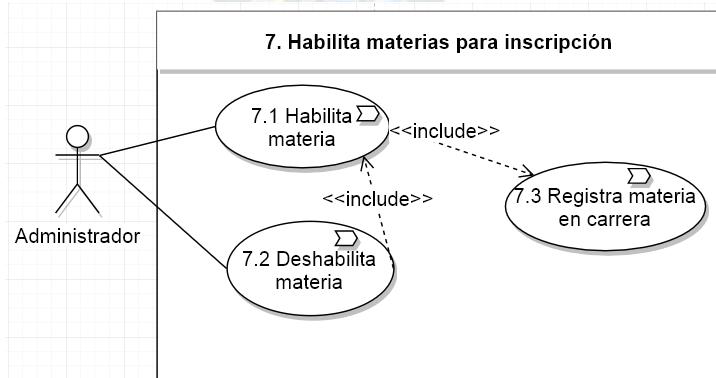


Figura 3. 8: Caso de Uso habilita materia para inscripción

Fuente: [Elaboración Propia]

Caso de Uso:
Habilita materia para inscripción
Actores:
Administrador
Descripción:
Permite habilitar materias para que el estudiante pueda realizar su inscripción.
Precondiciones:
La materia debe de estar registrada como parte de la carrera.
Flujo normal:
<ol style="list-style-type: none"> 1. El actor ingresa al módulo y visualiza la lista de materias sin habilitar y las habilitadas 2. Al presionar un botón, habilita o deshabilita la materia.
Postcondición:
Es mostrado un mensaje de confirmación de la acción y la materia es agregada a la lista de habilitados o deshabilitados.

Tabla 3. 9: Descripción del Caso de Uso “Habilita materias para inscripción”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.9 se muestra el diagrama de caso de uso para el registro de pre inscripción al curso de temporada junto con sus especificaciones en la tabla 3.10.

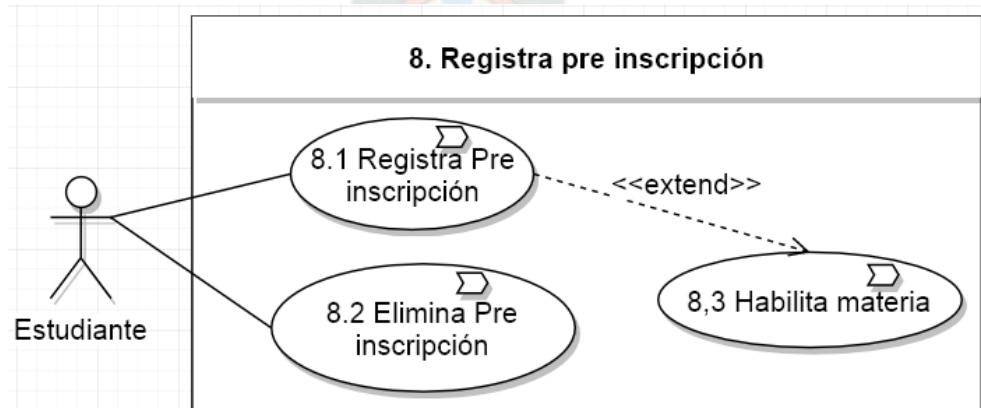


Figura 3. 9: Caso de Uso Registra pre inscripción

Fuente: [Elaboración Propia]

Caso de Uso: Registra pre inscripción
Actores: Estudiante
Descripción: Permite registrar la preinscripción a dos materias.
Precondiciones: <ul style="list-style-type: none"> • El actor debe estar registrado como estudiante. • El curso de temporada debe de estar con estado publicado. • La fecha límite de pre inscripciones debe ser mayor a la fecha en la que realiza el proceso.
Flujo normal: <ol style="list-style-type: none"> 1. El actor ingresa al módulo y visualiza la lista de materias para ser pre inscrito. 2. En cada materia se encuentra un botón para realizar la pre inscripción o la anulación de la misma.
Postcondición: Es mostrado un mensaje de confirmación de la acción y si la cantidad de pre inscritos sobre pasa la solicitada la materia es agregada a la lista de habilitados para la inscripción.

Tabla 3. 10: Descripción del Caso de Uso “registra preinscripción”

Fuente: [Elaboración Propia]

A continuación, en la figura 3.10 se muestra el diagrama de caso de uso para el registro de la inscripción al curso de temporada junto con sus especificaciones en la tabla 3.11.

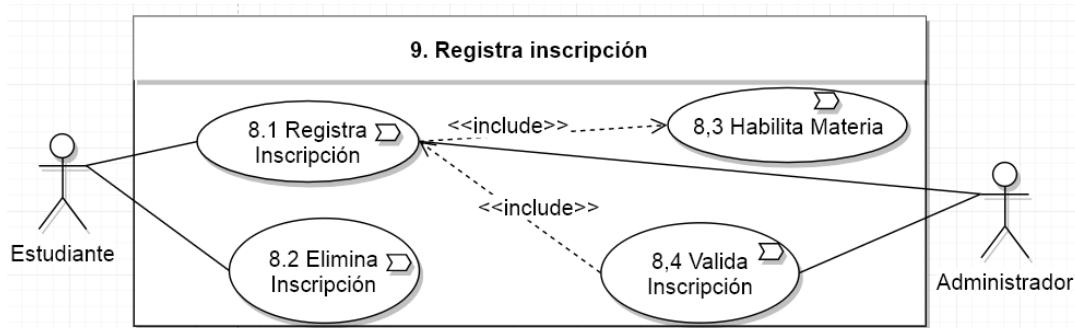


Figura 3. 10: Caso de Uso Registra inscripción

Fuente: [Elaboración Propia]

A continuación, la tabla refleja en detalle el caso de uso registra inscripción.

Caso de Uso: Registrar inscripción
Actores: Estudiante, Administrador
Descripción: Permite registrar la inscripción a materias.
Precondiciones: <ul style="list-style-type: none"> • El actor debe estar registrado como estudiante o administrador. • El curso de temporada debe de estar con estado publicado • La fecha límite de inscripciones debe ser mayor a la fecha en la que realiza el proceso.
Flujo normal: <ol style="list-style-type: none"> 1. El actor estudiante ingresa al módulo de inscripción y registra una nueva inscripción. 2. En el formulario selecciona la materia a inscribirse y guarda el registro, como máximo dos registros. 3. Para validar la inscripción, el actor administrador busca por el CI del estudiante y visualiza la lista de materias a confirmar la inscripción.

- | |
|--|
| <p>4. El actor administrador valida la inscripción presionando un botón.</p> <p>1A El actor administrador ingresa al módulo de inscripción.</p> <p>2A Busca al estudiante por su CI, y selecciona a las materias a inscribir directamente.</p> |
|--|

Postcondición:

Es mostrado un mensaje de confirmación de la acción, y es visualizado en la cuenta del estudiante como inscrito

Tabla 3. 11: Descripción del Caso de Uso “registra inscripción”

Fuente: [Elaboración Propia]

3.2.3. DEFINICIÓN DEL CRONOGRAMA DE TRABAJO

El cronograma de trabajo se definió en base al ciclo de vida de la metodología SCRUM, y se identifican 3 etapas principales, ya mencionadas anteriormente (pre-game, game, post-game), este cronograma se puede apreciar en un diagrama de Gantt en los Anexos de este documento.

3.2.4. ANÁLISIS DE RIESGOS

Es la probabilidad de ocurrencia de un acontecimiento adverso, problema o daño y las consecuencias del mismo y el análisis de riesgo es la aplicación de los métodos y herramientas existentes para determinar tal probabilidad, entre ellas tenemos 3 tipos de riesgo:

- Riesgo del Proyecto.** Que afecta a la ejecución de los módulos en ciertos períodos de tiempo ya definidos por el grupo de trabajo.

2. **Riesgos del Producto.** Afectan a la calidad o rendimiento del software que se está desarrollando.

3. **Riesgo de Negocio.** Afectan a la Organización o Institución que desarrolla o suministra el Software.

Riesgo	Tipo	Descripción	Probabilidad	Efecto	Estrategia
No se cumplen con las fechas establecidas en el cronograma.	Proyecto	Es probable que las fechas descritas en el diagrama de Gantt no se cumplan.	Alta	Tolerable	Realizar un segundo cronograma más flexible.
Cambios frecuentes en los requerimientos del cliente	Proyecto Producto	El riego más común es generalmente el miedo al cambio en relación a los usuarios finales	Alta	Serio	Realizar reuniones de capacitación constante.
No se cumplen los plazos de entrega del software.	Producto	Los plazos de entrega están sujetos al Scrum	Moderada	Serio	Agilizar procesos de desarrollo

		Master del proyecto			
No existe el incentivo adecuado para la elaboración del proyecto.	Producto Proyecto	Disminución de activos, presión innecesaria.	Baja	Tolerable	Entregas de versiones funcionales en cortos periodos de tiempo.

Tabla 3. 12: Análisis de Riesgos

Fuente: [Elaboración Propia]

3.3. GAME

Durante esta etapa del proyecto cada iteración adoptó los modelos de la metodología UWE que posteriormente fueron implementados basándose principalmente en la Base de Datos de Software (Gestor de Base de Datos PostgreSQL).

En cuanto a la persistencia de objetos se optó por hacer corresponder cada clase del modelo conceptual con alguna tabla de la Base de Datos, donde las filas representarían las instancias de los objetos y las columnas a los atributos de las clases.

Las clases y sus métodos fueron implementados en el lenguaje NodeJs

Finalmente se desarrollaron las vistas de entornos Web en base a los modelos de presentación y navegación.

3.3.1. PRIMERA ITERACIÓN

Durante la primera iteración se desarrollan los elementos necesarios y base para los módulos del curso de temporada y evaluación docente. Las actividades realizadas durante esta iteración se observan en la siguiente tabla, que constituye el BackLog del Sprint.

ID	Tareas	Sprint	Inicio	Duración
		Tipo	Días de trabajo	Estado
1.1	Realizar la planificación de la iteración.	Planificación	2	Terminado
1.2	Analizar los requerimientos del Backlog del producto.	Planificación	2	Terminado
1.3	Analizar los requerimientos de la interacción con historias de usuario	Desarrollo	2	Terminado
1.4	Diseñar la base de datos	Desarrollo	5	Terminado
1.5	Construir el módulo central de administrativos.	Desarrollo	3	Terminado
1.6	Construir el módulo central de docentes.	Desarrollo	3	Terminado
1.7	Construir el módulo central de estudiantes.	Desarrollo	3	Terminado
1.8	Construir el módulo central de materias y carreras	Desarrollo	3	Terminado
1.9	Construir el módulo central de acceso al sistema	Desarrollo	3	Terminado
1.10	Construir el módulo central	Desarrollo	3	Terminado
1.11	Probar los módulos desarrollados en la iteración.	Desarrollo	1	Terminado

Tabla 3. 13: Backlog del primer sprint

Fuente: [Elaboración Propia]

A continuación, se describe las funcionalidades correspondientes al incremento de esta iteración:

- Base de Datos funcional de acuerdo a los requerimientos del Servicio Web que será desarrollado posteriormente.
- Ingreso de datos del estudiante mediante el Registro Universitario, Carnet de Identidad.
- Consulta, registro, modificación y eliminación de estudiantes, docentes, administrativos, carreras, materias y ambientes.

3.3.1.1 MODELO CONCEPTUAL

Se genera el modelo conceptual en base a la estructura final de la base de datos considerando el manejo de relaciones del ORM del framework utilizado (Insac).

Previamente se muestra la estructura para la autenticación del usuario, ver Figura 3.11

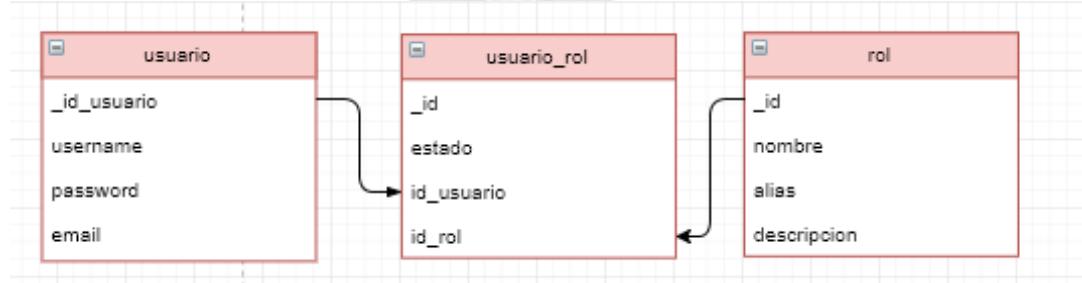


Figura 3. 11: Modelo conceptual autenticación del usuario.

Fuente: [Elaboración Propia]

Para la siguiente representación la base de datos es dividida por los módulos previamente mencionados, lo que no determina que la base de datos del proyecto es dividida.

De tal forma se muestra el módulo central del software, ver Figura 3.12.

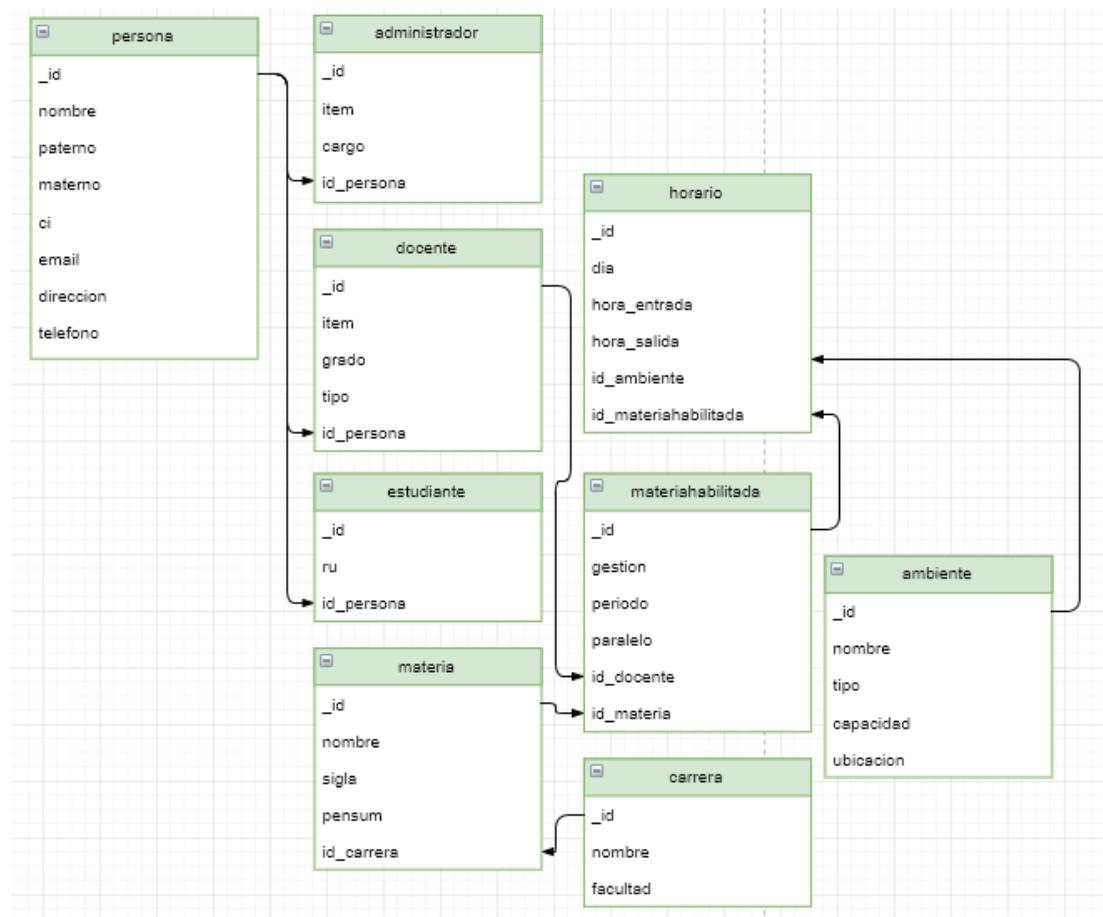


Figura 3. 12: Modelo conceptual módulo central del sistema.

Fuente: [Elaboración Propia]

En la figura 3.13, se muestra la estructura generada para el módulo Cursos de temporada.

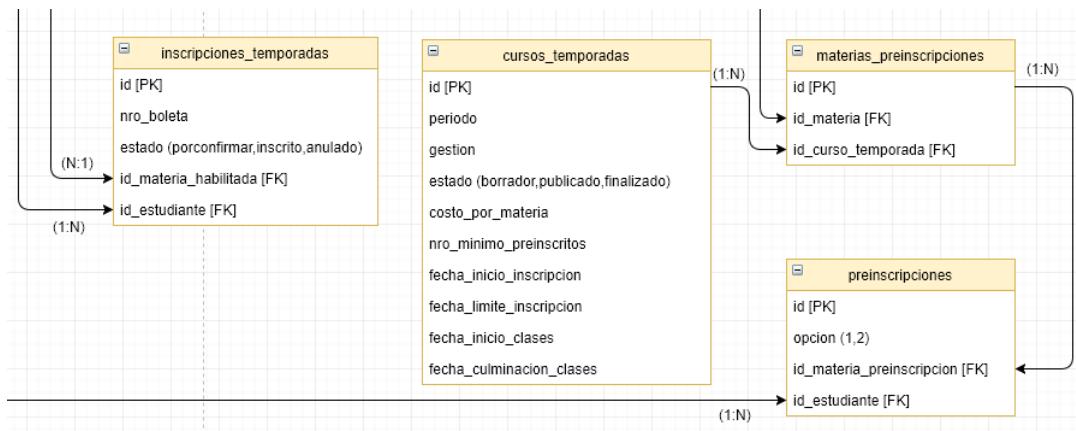


Figura 3. 13: Modelo conceptual módulo Cursos de temporada.

Fuente: [Elaboración Propia]

En la Figura 3.14 se muestra la estructura para la evaluación docente.

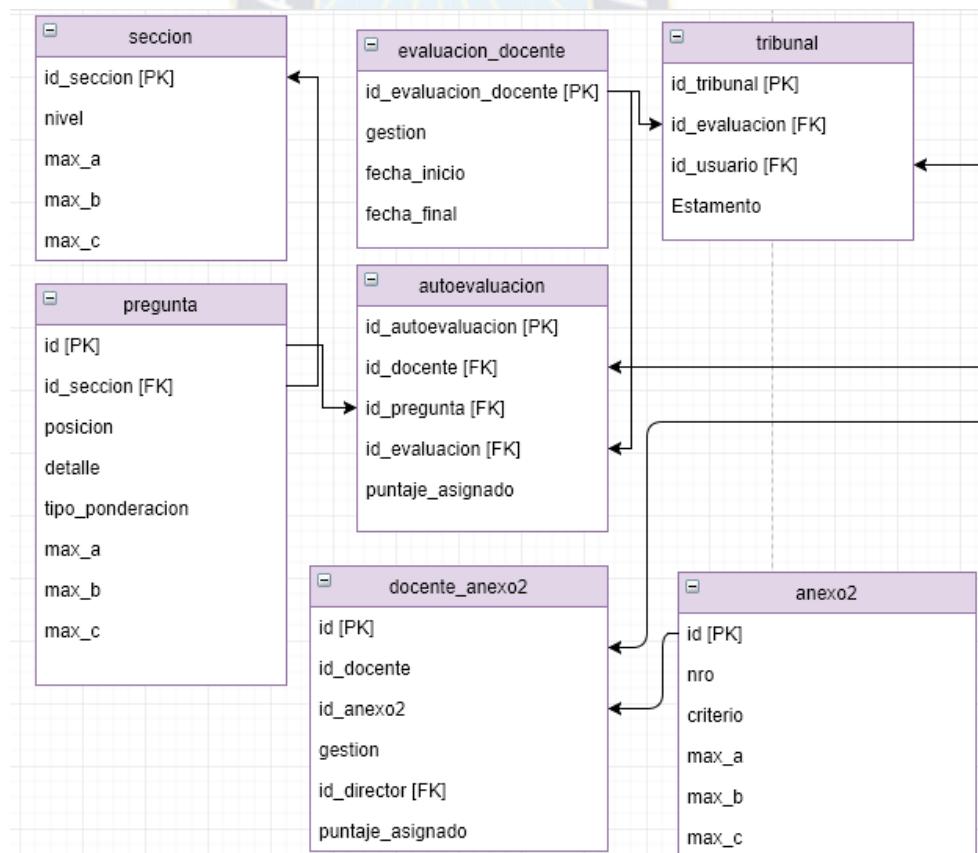


Figura 3. 14: Modelo conceptual módulo Cursos de temporada.

Fuente: [Elaboración Propia]

3.3.1.2 MODELO DE NAVEGACIÓN

La Figura 3.15 muestra el modelo de navegación en base a la pila del Sprint 1.

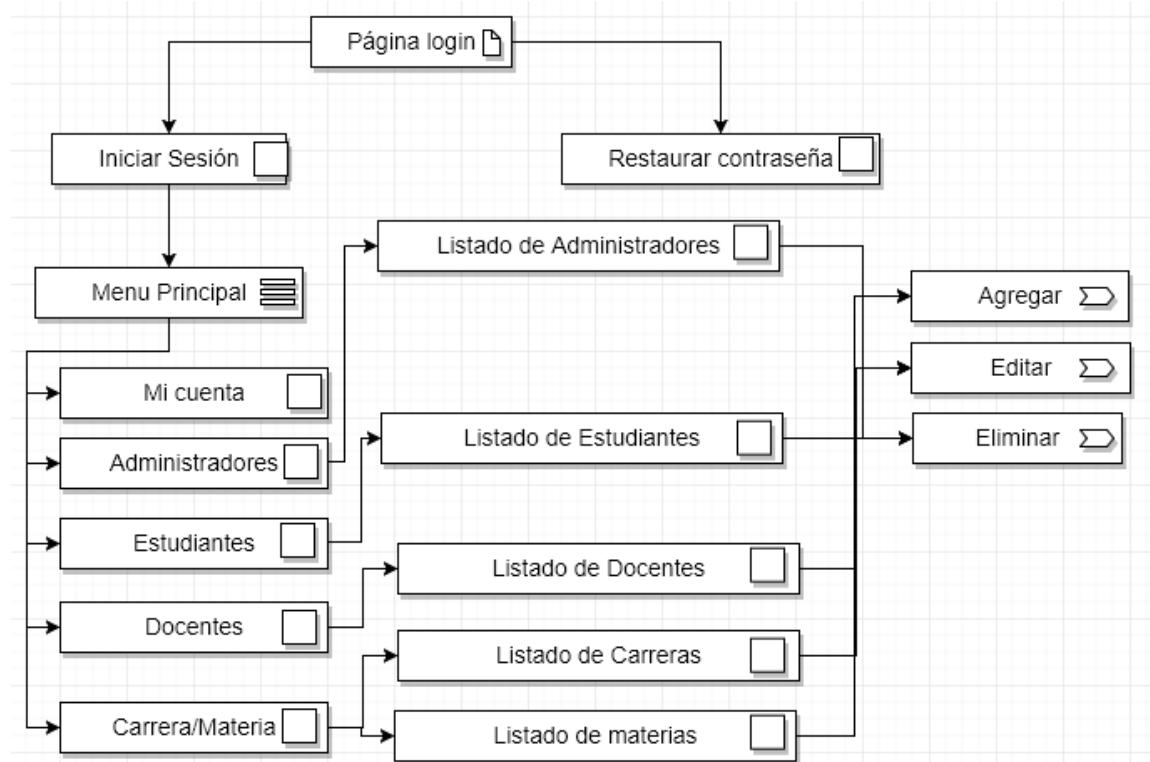


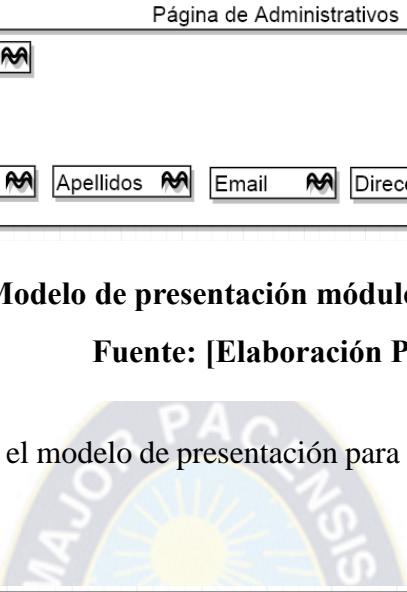
Figura 3. 15: Modelo navegacional para el Sprint 1

Fuente: [Elaboración Propia]

3.3.1.3 MODELO DE PRESENTACIÓN

En la Figura 3.16 se muestra el modelo de presentación para el módulo central de administrativos.

Página de Administrativos



Menu Sauron

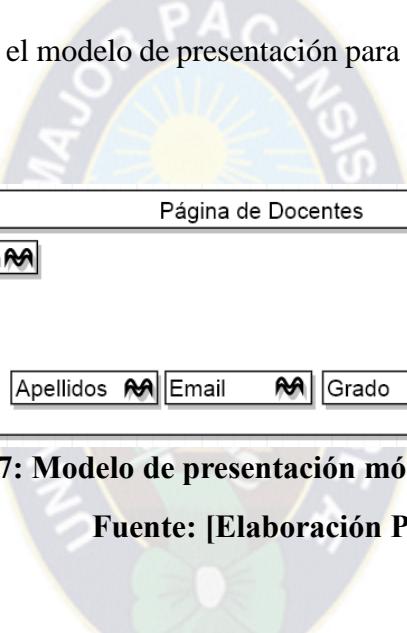
Nuevo Administrativo

CI Nombres Apellidos Email Dirección Editar Eliminar

Figura 3. 16: Modelo de presentación módulo central de administrativos

Fuente: [Elaboración Propia]

En la Figura 3.17 se ve el modelo de presentación para el módulo central de docentes.



Página de Docentes

Menu Sauron

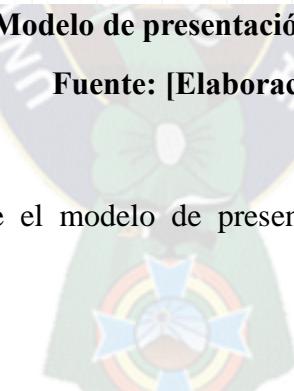
Nuevo Docente

CI Nombres Apellidos Email Grado Editar Eliminar

Figura 3. 17: Modelo de presentación módulo central de docentes

Fuente: [Elaboración Propia]

En la Figura 3.18 se ve el modelo de presentación para el módulo central de estudiantes.



Página de Estudiantes

Menu Sauron

Nuevo Estudiante

RUM CI Apellidos Nombres Carrera Editar Eliminar

Figura 3. 18: Modelo de presentación módulo central de estudiantes.

Fuente: [Elaboración Propia]

En la Figura 3.19 se ve el modelo de presentación para el módulo central de materias y carreras.

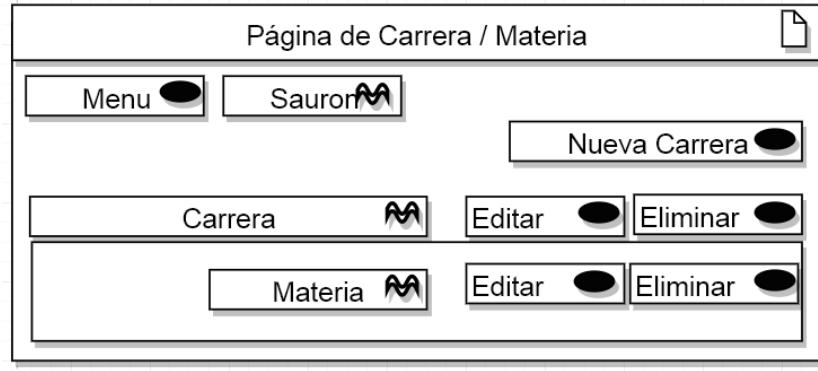


Figura 3. 19: Modelo de presentación módulo central de materias y carreras

Fuente: [Elaboración Propia]

En la Figura 3.20 se ve el modelo de presentación para el módulo central de acceso al sistema.

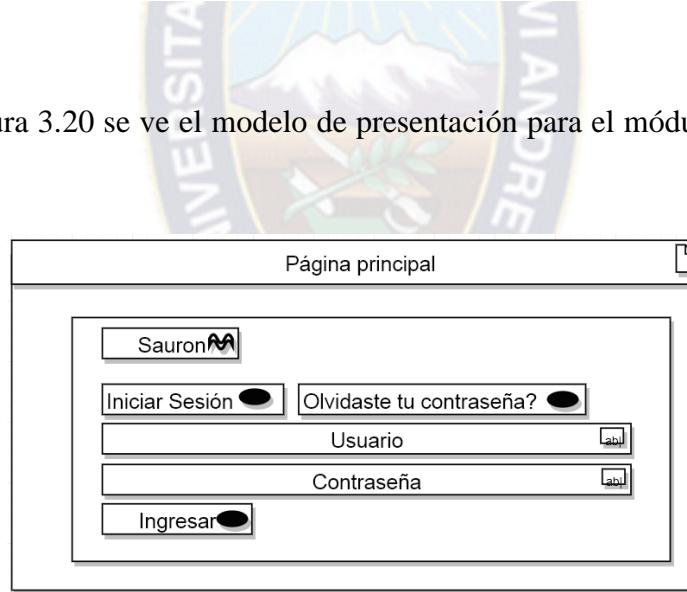


Figura 3. 20: Modelo de presentación módulo central de acceso al sistema

Fuente: [Elaboración Propia]

A continuación, se tiene las capturas de pantalla de las tareas implementadas en el Sprint 1

En la Figura 3.21 se muestra la pantalla de inicio de sesión.

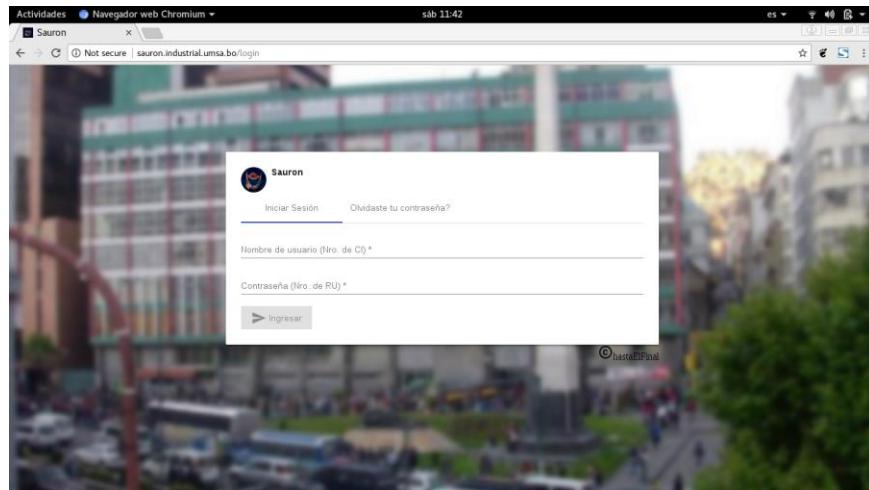


Figura 3. 21: Pantalla de Inicio de Sesión

Fuente: [Elaboración Propia]

En la Figura 3.22 se muestra la pantalla principal del módulo de administradores.

A screenshot of a web application interface titled "Administradores". At the top right is a blue button labeled "+ Agregar". The main area is a table with the following data:

Figura 3. 22: Pantalla de módulo central de administradores

Fuente: [Elaboración Propia]

En la figura 3.23 se muestra la pantalla del formulario de registro de docentes administradores.

Dore

Registro de Usuarios

Cargo del administrador *

Nombre del administrador *

Apellido Paterno del administrador

Apellido Materno del administrador

Carnet de identidad del administrador *

Correo electrónico del administrador *

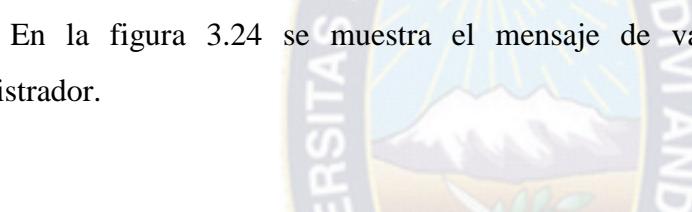
Dirección del administrador *

Teléfono del administrador *

Figura 3. 23: Pantalla del formulario de registro y edición de administradores

Fuente: [Elaboración Propia]

En la figura 3.24 se muestra el mensaje de validación para eliminar a un administrador.



Administradores

Está segur@ de eliminar al usuario AQUINO VICTOR?

C.I.	Nombres	Apellidos	E-mail	Dirección	Teléfono	Cargo	Acciones
123456	CARLOS	FLORES MAHAMI	israelhuilpara@gmail.com	Zona Obreras, Esq. Calle 8, #24	78674654	KARDEX	
4849175	ENRIQUE	OROSCO	enriqueorosco.ing@gmail.com	en algun lugar	79550203	SISTEMAS	
911911	RODRIGO	GARCIA	sty742@gmail.com	algun lugar	70563989	SISTEMAS tester	
10109	VICTOR	AQUINO	vamaquino@gmail.com	primer piso industrial	71241124	kardex	

Figura 3. 24: Pantalla de validación del eliminado de un administrador

Fuente: [Elaboración Propia]

Los módulos de estudiantes, docente, carrera y materia tienen la misma estructura y apariencia, para mantener al usuario en el mismo contexto y así en cada uso aumente su adecuación.

3.3.2. SEGUNDA ITERACIÓN

En esta iteración se desarrolló el módulo del curso de temporada tanto para el administrador como para los estudiantes.

ID	Tareas	Sprint	Inicio	Duración
		2	1/09/2017	21/09/2017
1.1	Realizar la planificación de la iteración.	Planificación	2	Terminado
1.2	Analizar los requerimientos del Backlog del producto.	Planificación	2	Terminado
1.3	Analizar los requerimientos de la interacción con historias de usuario	Desarrollo	2	Terminado
1.4	Construir el módulo de administración del curso de temporada.	Desarrollo	3	Terminado
1.5	Construir el módulo de preinscripción para estudiantes	Desarrollo	2	Terminado
1.6	Construir el módulo de preinscripción para administrador	Desarrollo	2	Terminado
1.7	Construir el módulo de inscripción para estudiantes.	Desarrollo	2	Terminado
1.8	Construir el módulo de validación e inscripción para administradores.	Desarrollo	2	Terminado
1.9	Construir el módulo de generación de reportes.	Desarrollo	3	Terminado
1.10	Probar los módulos desarrollados en la iteración.	Desarrollo	1	Terminado

Tabla 3. 14: Backlog del segundo sprint

Fuente: [Elaboración Propia]

A continuación, se detallan las funcionalidades correspondientes al incremento de esta segunda iteración.

- Administración de cursos de temporada.
- Pre inscripción e inscripción por parte de estudiantes.
- Habilitación automática y manual de materias para la inscripción.
- Inscripción por parte de estudiantes.
- Validación y/o inscripción por el administrador.
- Generación de reportes de pre inscripción e inscripción.

3.3.2.1 MODELO DE NAVEGACIÓN

La figura 3.25 muestra el modelo navegacional en base a la pila del sprint 2.

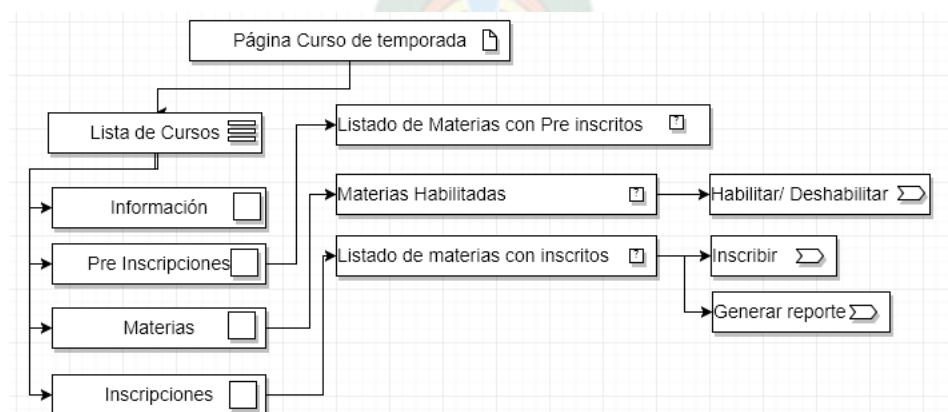


Figura 3. 25: Modelo navegacional para el Sprint 2

Fuente: [Elaboración Propia]

3.3.2.2 MODELO DE PRESENTACIÓN

En la Figura 3.26 se ve el modelo de presentación para el módulo administración del curso de temporada.

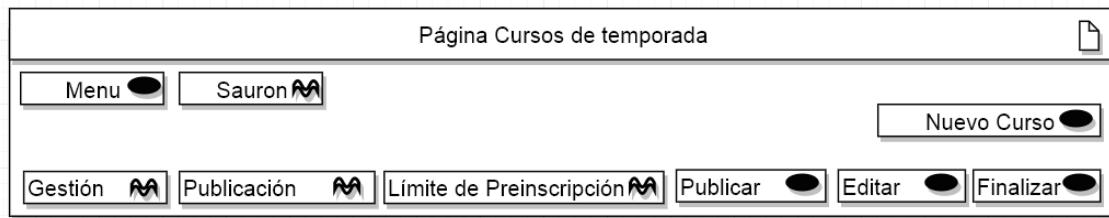


Figura 3. 26: Modelo de presentación administración del curso de temporada

Fuente: [Elaboración Propia]

En la Figura 3.27 se ve el modelo de presentación para el módulo pre inscripción para estudiantes.

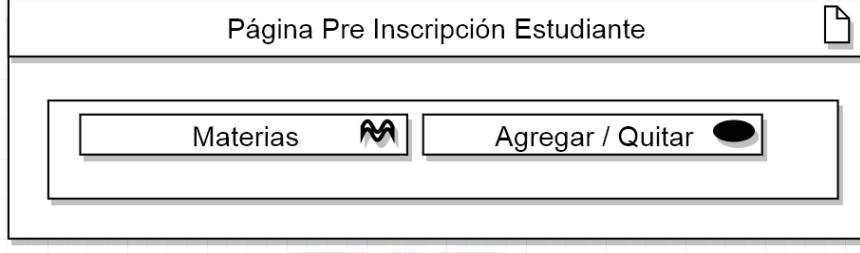


Figura 3. 27: Modelo de presentación módulo pre inscripción estudiantes

Fuente: [Elaboración Propia]

En la Figura 3.28 se ve el modelo de presentación para el módulo pre inscripción para administrativo.

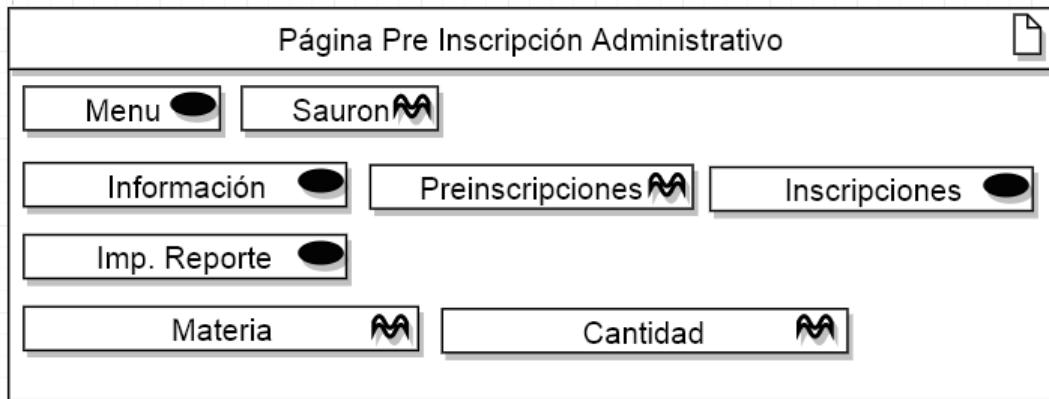


Figura 3. 28: Modelo de presentación módulo pre inscripción administrativo

Fuente: [Elaboración Propia]

En la Figura 3.29 se ve el modelo de presentación para el módulo de inscripción para estudiantes.

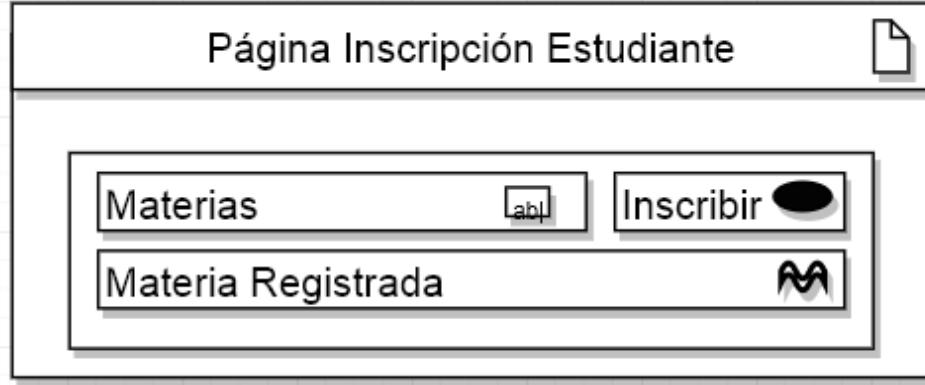


Figura 3. 29: Modelo de presentación inscripción para estudiantes

Fuente: [Elaboración Propia]

En la Figura 3.30 se ve el modelo de presentación para el módulo de inscripción para administrativos.

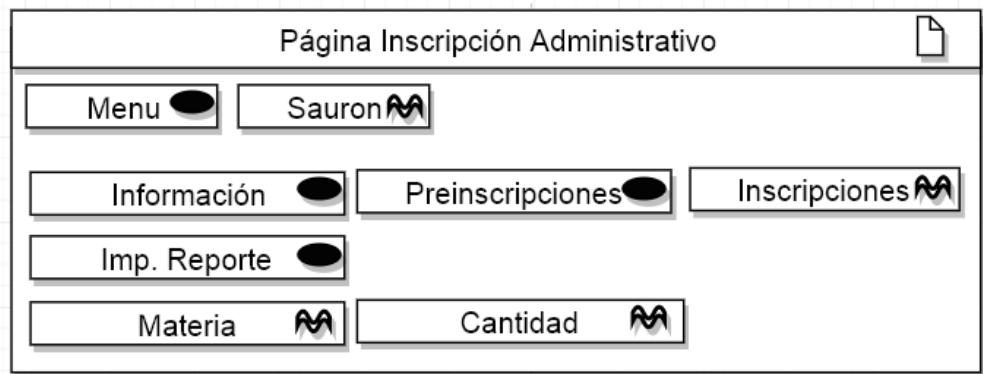


Figura 3. 30: Modelo de presentación inscripción para administrativos

Fuente: [Elaboración Propia]

A continuación, se tiene las capturas de pantalla de las tareas implementadas en el Sprint 2.

En la Figura 3.31 se muestra la pantalla de registro del curso de temporada.

A screenshot of a web form titled "Registro del curso de temporada". The form is part of a larger application window with a sidebar showing years from 2015 to 2018. The main form fields include: "INV" and "VER" buttons, a dropdown menu for "Gestión" showing "2017", "Fecha inicio de inscripción *", "Fecha límite de inscripción *", "Fecha inicio de clases *", "Fecha culminación de clases *", "Costo por materia *", and "Cantidad mínima de pre-inscritos *". There are also "Nuevo" and "Borrar" buttons on the right side of the form.

Figura 3. 31: Modelo de presentación inscripción para administrativos

Fuente: [Elaboración Propia]

En la Figura 3.32 se muestra la pantalla de pre inscripciones del administrador.

Materia	Sigla	Cantidad
INFORMATICA PARA INGENIERIA Y LABORATORIO	IND - 312	0
CONTABILIDAD	IND - 122	0
CONSTRUCCIONES E INSTALACIONES INDUSTRIALES	IND - 414	0
ELECTROTECNIA ELECTRONICA Y LABORATORIO	IND - 322	0
TEORIA ECONOMICA	IND - 222	0
CALCULO DE PROBABILIDADES	IND - 311	0
ESTADISTICA INFERENCIAL	IND - 411	0
PREPARACION Y EVALUACION DE PROYECTOS I	IND - 723	0
MARKETING	IND - 621	0
ECONOMETRIA	IND - 521	0
PREPARACION Y EVALUACION DE PROYECTOS II	IND - 823	0

Figura 3. 32: Pantalla de pre inscripciones del administrador

Fuente: [Elaboración Propia]

En la Figura 3.33 se muestra la pantalla de materias habilitadas del administrador.

Materias sin habilitarse		Materias Seleccionadas	
[IND - 322] ELECTROTECNIA ELECTRONICA Y LABORATORIO Ingeniería Industrial	✓	INFORMATICA PARA INGENIERIA Y LABORATORIO Ing. Industrial	☒
[IND - 311] CALCULO DE PROBABILIDADES Ingeniería Industrial	✓	CONSTRUCCIONES E INSTALACIONES INDUSTRIALES Ing. Industrial	☒
[IND - 411] ESTADISTICA INFERENCIAL Ingeniería Industrial	✓	TEORIA ECONOMICA Ing. Industrial	☒
[IND - 723] PREPARACION Y EVALUACION DE PROYECTOS I Ingeniería Industrial	✓	CONTABILIDAD Ing. Industrial	☒
[IND - 621] MARKETING Ingeniería Industrial	✓		
[IND - 521] ECONOMETRIA			

Figura 3. 33: Pantalla de materias habilitadas

Fuente: [Elaboración Propia]

En la Figura 3.34 se muestra la pantalla de inscripción del administrador.



Figura 3. 34: Pantalla de inscripción del administrador

Fuente: [Elaboración Propia]

En la Figura 3.35 se muestra el reporte de inscripciones del administrador.

This screenshot shows a print dialog box on the left and a report table on the right. The print dialog shows 'Total: 5 pages' and options to 'Save' or 'Cancel'. The report table is titled 'TEORIA ECONOMICA' and contains the following data:

R.U.	C.I.	Apellidos y Nombres	Carrera	Nro
1646325	9123122	HUALLPARA MENCHAS ISRAEL JOSE	Ingeniería Industrial	3

Figura 3. 35: Pantalla de inscripción del administrador

Fuente: [Elaboración Propia]

En la Figura 3.36 se muestra el formulario de pre inscripción del estudiante.

This screenshot shows a modal window titled 'Pre Inscripciones' with a list of courses and their current pre-enrollment counts. To the right of each count is a red circular button with a white question mark icon, likely for more details. The courses listed are:

- INFORMATICA PARA INGENIERIA Y LABORATORIO: Cnt. de pre inscritos: [8]
- CONTABILIDAD: Cnt. de pre inscritos: [0]
- CONSTRUCCIONES E INSTALACIONES INDUSTRIALES: Cnt. de pre inscritos: [8]
- ELECTROTECHA ELECTRONICA Y LABORATORIO: Cnt. de pre inscritos: [1]
- TEORIA ECONOMICA: Cnt. de pre inscritos: [0]
- CALCULO DE PROBABILIDADES: Cnt. de pre inscritos: [1]

Figura 3. 36: Pantalla de pre inscripción del estudiante

Fuente: [Elaboración Propia]

En la Figura 3.37 se muestra el formulario de inscripción del estudiante.

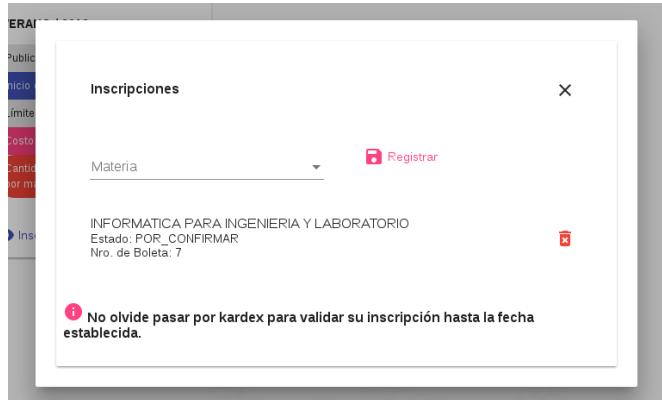


Figura 3. 37: Pantalla de inscripción del estudiante

Fuente: [Elaboración Propia]

3.3.3. TERCERA ITERACIÓN

En esta iteración se desarrolló el módulo de evaluación docente para docentes titulares, director de carrera y tribunal asignado.

		Sprint	Inicio	Duración
		3	22/09/2017	10/10/2017
ID	Tareas	Tipo	Días de trabajo	Estado
1.1	Realizar la planificación de la iteración.	Planificación	2	Terminado
1.2	Analizar los requerimientos del Backlog del producto.	Planificación	2	Terminado
1.3	Analizar los requerimientos de la interacción con historias de usuario	Desarrollo	2	Terminado
1.4	Construir el módulo de administración para evaluación docente.	Desarrollo	3	Terminado
1.5	Construir el módulo de registro de logros del docente titular.	Desarrollo	2	Terminado

1.6	Construir el módulo de ponderación del director de carrera.	Desarrollo	2	Terminado
1.7	Construir el módulo de ponderación y observaciones del tribunal.	Desarrollo	2	Terminado
1.8	Construir el módulo de generación de reportes.	Desarrollo	3	Terminado
1.9	Probar los módulos desarrollados en la iteración.	Desarrollo	1	Terminado

Tabla 3. 15: Backlog del tercer sprint

Fuente: [Elaboración Propia]

A continuación, se detallan las funcionalidades correspondientes al incremento de esta tercera iteración:

- Registro y generación de reporte en PDF de logros por parte del docente evaluado.
- Ponderación y reporte en PDF por parte del director de carrera.
- Ponderación y reporte en PDF por parte del tribunal.
- Reporte final en PDF por parte del administrador del sistema.

3.3.3.1 MODELO DE NAVEGACIÓN

La figura 3.38 muestra el modelo navegacional en base a la pila del sprint 3, que corresponde al módulo de evaluación docente.

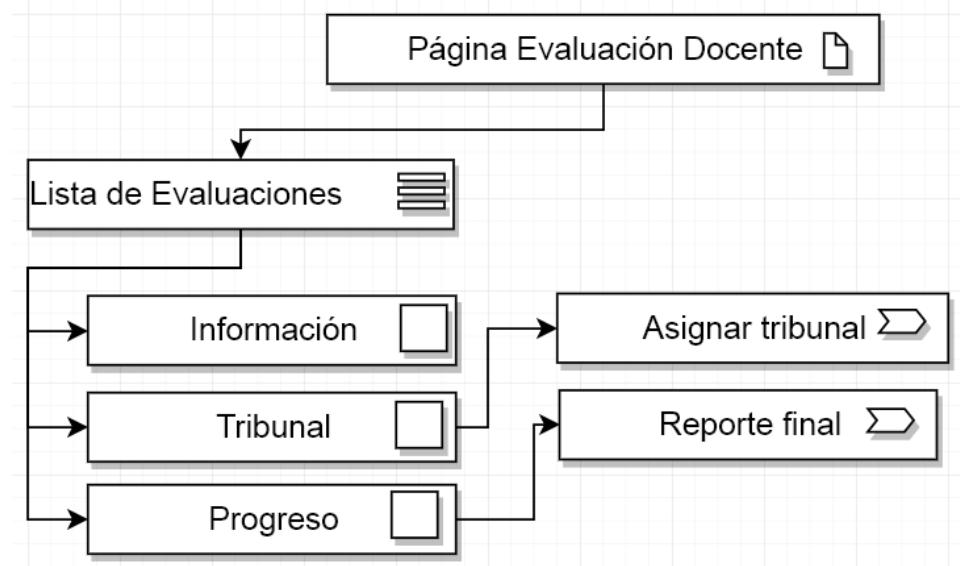


Figura 3. 38: Modelo navegacional para el Sprint 3

Fuente: [Elaboración Propia]

3.3.3.2 MODELO DE PRESENTACIÓN

En la Figura 3.39 es mostrado el modelo de presentación para el módulo de administración para evaluación docente.

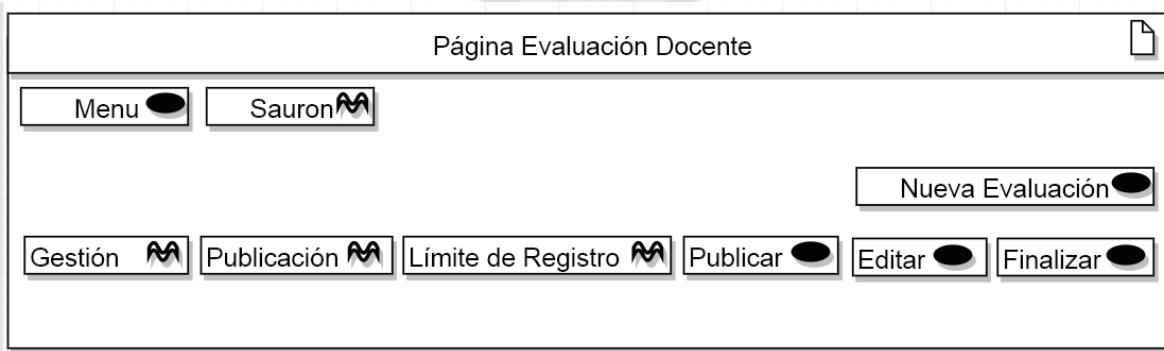


Figura 3. 39: Modelo de presentación módulo de administración evaluación docente

Fuente: [Elaboración Propia]

En la Figura 3.40 se ve el modelo de presentación para el módulo de registro de logros o informe docente por el docente titular.

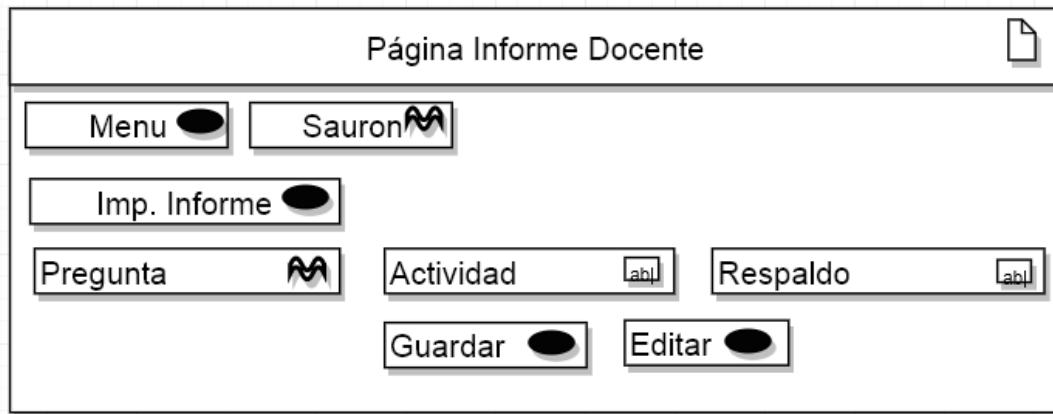


Figura 3. 40: Modelo de presentación módulo de registro de logros por el docente titular

Fuente: [Elaboración Propia]

En la Figura 3.41 se ve el modelo de presentación para el módulo de ponderación por el director de carrera.

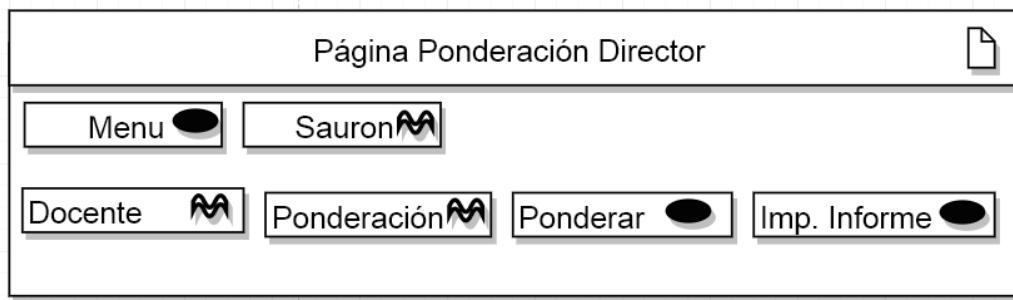


Figura 3. 41: Modelo de presentación módulo de ponderación por el director de carrera

Fuente: [Elaboración Propia]

En la Figura 3.42 se ve el modelo de presentación para el módulo de ponderación y observación por el tribunal.

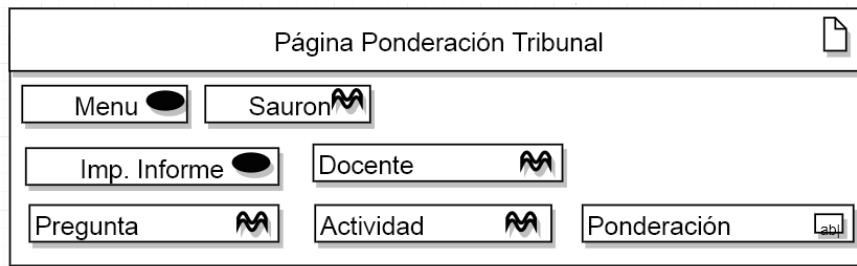


Figura 3. 42: Modelo de presentación de ponderación y observación por el tribunal

Fuente: [Elaboración Propia]

A continuación, se tiene las capturas de pantalla de las tareas implementadas en el Sprint 3.

En la Figura 3.43 se ve la pantalla del listado de evaluaciones docentes.

Evaluación Docente							Agregar
Gestión	Publicación	Límite autoevaluación	Revisión por el tribunal	Presentación de observaciones	Revisión de observaciones	Acciones	
2016-2017	07/11/2017	07/11/2017	14/11/2017	21/11/2017	28/11/2017		

Figura 3. 43: Pantalla de lista de evaluaciones docentes

Fuente: [Elaboración Propia]

En la Figura 3.44 se ve la pantalla del formulario de registro de evaluación docente.

Registro de la Evaluación Docente

Gestión *

Fecha límite autoevaluación por docente *

Fecha límite revisión por el tribunal *

Fecha límite observaciones y reclamos *

Fecha límite revisión final por el tribunal *

Figura 3. 44: Formulario de registro de evaluación docente.

Fuente: [Elaboración Propia]

En la Figura 3.45 se ve la pantalla del formulario de asignación de tribunal.

Registro de Tribunales

CI del tribunal
7089640

Nombres del tribunal
RAQUEL GLADYS

Apellidos del tribunal
YUJRA CHOQUETOPA

Estamento del tribunal
ESTUDIANTE

Figura 3. 45: Formulario de asignación de tribunal

Fuente: [Elaboración Propia]

En la Figura 3.46 se ve la pantalla del reporte en pdf del informe final.

EVALUACIÓN DOCENTE GESTIÓN 2016-2017

Nº	APELLIDOS Y NOMBRES	GRUPO	CARGA HORARIA	CATEGORÍA	EVALUACIÓN						PUNTAJE/PF CATE.	
					LABOR DOCENTE (40)	LABOR INVESTIGACIÓN (50)	LABOR EXTENSIÓN (10)	PRODUCCIÓN INTTELUAL (50)	PARTICIPACIÓN VIDA UNIV. (20)	CURSOS Y CONFERENCIAS (20)		TOTAL 180

La paz, 7 de Noviembre del 2017

CASTRO ORDÓÑEZ JOSE MANUEL
COMISION DE EVALUACION DOCENTE

Univ. HUALLPARA MENCIAS ISRAEL JOSE
COMISION DE EVALUACION DOCENTE

Figura 3. 46: se ve la pantalla del reporte en pdf del informe final.

Fuente: [Elaboración Propia]

En la Figura 3.47 se ve la pantalla registro del informe docente.

LABOR DOCENTE

1.1 ADMINISTRACIÓN DE LA CÁTEDRA: Evalúa el Director de Carrera de acuerdo con el formulario aprobado en el CAU sobre la base del Reglamento de la Evaluación Docente, aprobado mediante Resolución de HCU (Ver Anexo 2)

1.2 DEDICACIÓN A LA DOCENCIA:

ADMINISTRACION INDUSTRIAL – DOCENTE TITULAR A

Adjunto: CERT DIV. ESC. DOC. N° 1317/08

CONTABILIDAD - DOCENTE TITULAR

Adjunto: CERT

Documento Respaldo

Figura 3. 47: Formulario de registro del informe docente

Fuente: [Elaboración Propia]

En la Figura 3.48 se ve la pantalla del registro de ponderación por el director, donde el límite es por la categoría.

Asignaturas	
1	sistemas industriales IID-523
1.-	Presentación de Programas en el tiempo oportuno. Puntaje 0
2.-	Cumplimiento de horarios, participación y asistencia regular más del 80% Puntaje 0
3.-	Publicación de notas parciales de acuerdo a cronograma Puntaje 0
4.-	Presentación de informe académico Puntaje 0
5.-	Cumplimiento de Reglamento Régimen Docente Estudiantil. Relación Humana Docente Estudiante, de acuerdo a documentos probatorios si la relación fuera óptima debe tener el máximo puntaje Puntaje 0
6.-	Presentación de acta de notas finales Puntaje 0
7.-	Otros Puntaje 0
TOTAL: 0.00 pts	

Figura 3. 48: Formulario de registro de ponderación por el director

Fuente: [Elaboración Propia]

En la Figura 3.49 se ve la pantalla del registro de ponderación por el tribunal.

Docente:	CHAMBI YANA MARCOS
Carga horaria:	32 hrs.
LABOR DOCENTE	
1.1 ADMINISTRACIÓN DE LA CÁTEDRA: Evalúa el Director de Carrera de acuerdo con el formulario aprobado en el CAU sobre la base del Reglamento de la Evaluación Docente, aprobado mediante Resolución de HCU (Ver Anexo 2) máx: 17.5 [0/17.5] Aún no fue calificado Puntaje máximo: 50 [0.00/50]	
1.2 DEDICACIÓN A LA DOCENCIA: máx: 15 [0/15]	
1.2.1 Asignatura (carga horaria de 32 horas/mes) clu 2 puntos cu: 2 [0]	
1.2.2 Por ser profesor tutor de un trabajo de tesis, tesina, trabajo dirigido, proyecto de grado, presentado y aprobado. clu 5 puntos cu: 5 [0]	
CONTABILIDAD I Adjunto: <i>Memorandum de asignación</i> Observación	
Ponderación <input type="text" value="0"/>	

Figura 3. 49: Formulario de registro de ponderación por el tribunal

Fuente: [Elaboración Propia]

3.4. POST – GAME

En esta etapa final se realizaron las actividades de prueba del Software Web, se propusieron métricas de calidad que se pueden apreciar en el Capítulo IV y se desarrollaron los diferentes manuales de usuario.

La etapa de pruebas de un sistema presenta un elemento crítico para la garantía de la calidad de software, y representa una revisión final de las especificaciones del diseño y la codificación [Pressman, 2012].

Para hacer las pruebas del sistema se hizo uso de las pruebas de prototipado rápido, donde se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción del prototipo, el cual fue evaluado por la institución para una retroalimentación, gracias a esto se refinan los requisitos del software que se desarrolló.

La interacción ocurrirá hasta que el sistema se ajuste para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo que el desarrollador entienda lo que debe hacer y el cliente vea resultados en corto plazo.

- **Perfectivo:** son las acciones llevadas a cabo para mejorar la calidad interna de los sistemas en cualquiera de sus aspectos: reestructuración del código, definición más clara del sistema y optimización del rendimiento y eficiencia.
- **Evolutivo:** son las incorporaciones, modificaciones y eliminaciones necesarias en un producto para cubrir la expansión o cambio en las necesidades del personal.
- **Adaptativo:** son las modificaciones que afectan a los entornos en los que el sistema opera, por ejemplo, cambios de configuración del hardware, software de base, bases de datos, comunicaciones, entre otros.
- **Correctivo:** son aquellos cambios precisos para corregir errores del producto software.

También consideramos las pruebas de stress a los servicios HTTP que el back-end proporciona, estas pruebas fueron realizadas con Jmeter que se dedica a realizar todo tipo de pruebas a sistemas.

Se consideró realizar la prueba al servicio de inicio de sesión por ser el más utilizado, teniendo como datos de prueba:

- 100 usuarios conectados al mismo tiempo.

- 30 peticiones por usuario.

La configuración para la prueba se puede ver en la figura 3.50

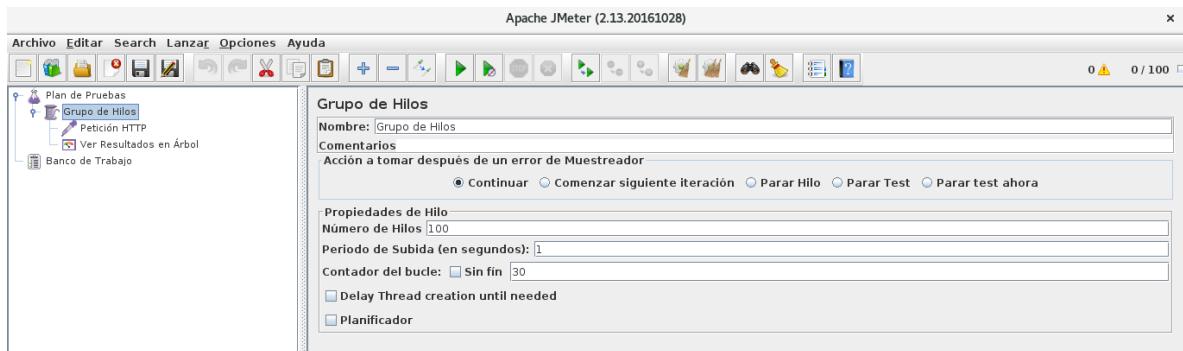


Figura 3. 50: Configuración para pruebas con Jmeter

Fuente: [Elaboración Propia]

Las pruebas fueron realizadas directamente en el servidor oficial del proyecto sauron.industrial.umsa.bo, como se ve en la figura 3.51.

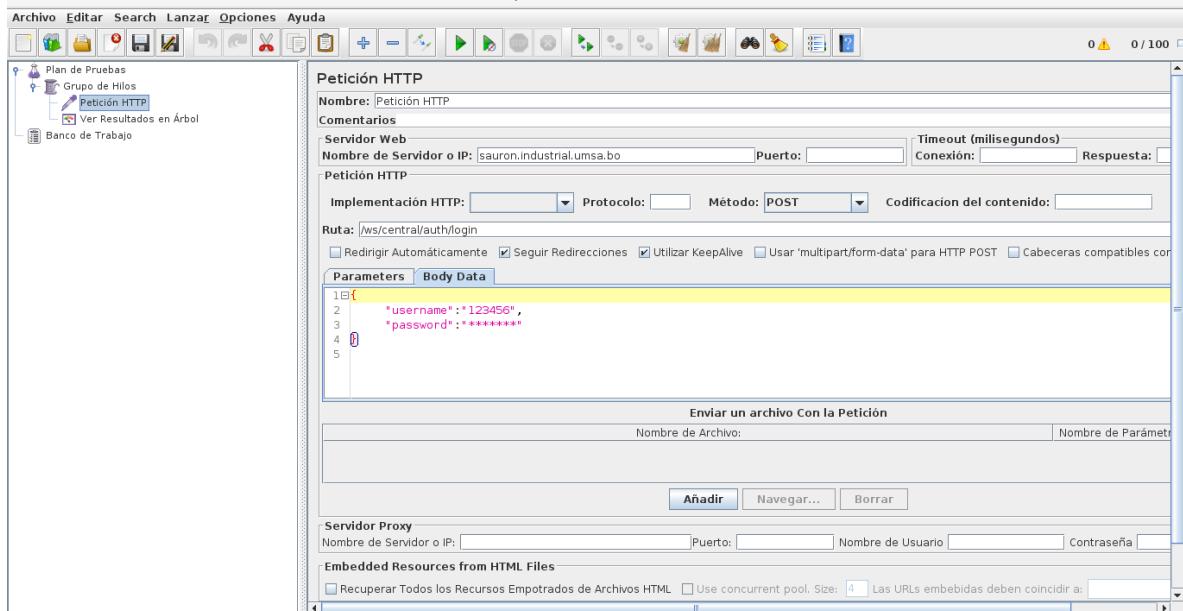


Figura 3. 51: Configuración para pruebas al servidor oficial

Fuente: [Elaboración Propia]

Con la configuración previa se llegó a obtener los resultados mostrados en la figura 3.52.

#	04:32:09.854	Grupo de Hilos...	PeticIÓN HTTP	2833	438	2833	465
20	04:32:09.853	Grupo de Hilos...	PeticIÓN HTTP	2833	438	2833	465
21	04:32:09.865	Grupo de Hilos...	PeticIÓN HTTP	2840	438	2840	458
22	04:32:09.863	Grupo de Hilos...	PeticIÓN HTTP	2844	438	2844	461
23	04:32:09.870	Grupo de Hilos...	PeticIÓN HTTP	2845	438	2845	458
24	04:32:09.886	Grupo de Hilos...	PeticIÓN HTTP	2829	438	2829	442
25	04:32:09.864	Grupo de Hilos...	PeticIÓN HTTP	2852	438	2852	461
26	04:32:09.863	Grupo de Hilos...	PeticIÓN HTTP	2853	438	2853	462
27	04:32:09.854	Grupo de Hilos...	PeticIÓN HTTP	2893	438	2893	468

Figura 3. 52: Configuración para pruebas al servidor oficial

Fuente: [Elaboración Propia]

Los datos generados fueron: de las 100 peticiones ninguna se perdió y su media en milisegundos es de 538 de respuesta por petición, mostrando que las respuestas son casi inmediatas de los servicios.



CAPÍTULO IV

CALIDAD Y SEGURIDAD DEL SOFTWARE

4.1. CALIDAD

El objetivo es llegar a la mejor calidad necesaria de evaluación bajo la norma ISO 9126, para ello se realiza los cálculos referentes a la funcionalidad, usabilidad, mantenibilidad y portabilidad del sistema.

4.1.1. FUNCIONALIDAD

La funcionalidad del Sistema se mide a través de su complejidad, la funcionalidad no puede ser medida directamente, entonces corresponde derivar, mediante otras medidas directas como Punto Función, que permite un resultado medible y cuantificable a partir de la siguiente fórmula: Para la medición de la funcionalidad, se toman en cuenta las siguientes características:

Parámetros de medida	Cuenta	Factor de peso			Total
		Simple	Medio	Complejo	
Número de entradas de Usuario	26	3	4	6	104
Número de Salidas de Usuario	13	4	5	7	65
Número de Peticiones de Usuario	8	3	4	6	32
Número de Archivos	23	7	10	15	230
Número de Interfaces	0	5	7	10	0
				Total	431

Tabla 4. 1: Cálculo Punto Función

Fuente: [Elaboración Propia]

Los valores de la variable F_i , se obtiene de los resultados que se aprecian en la Tabla 4.3, bajo la siguiente ponderación:

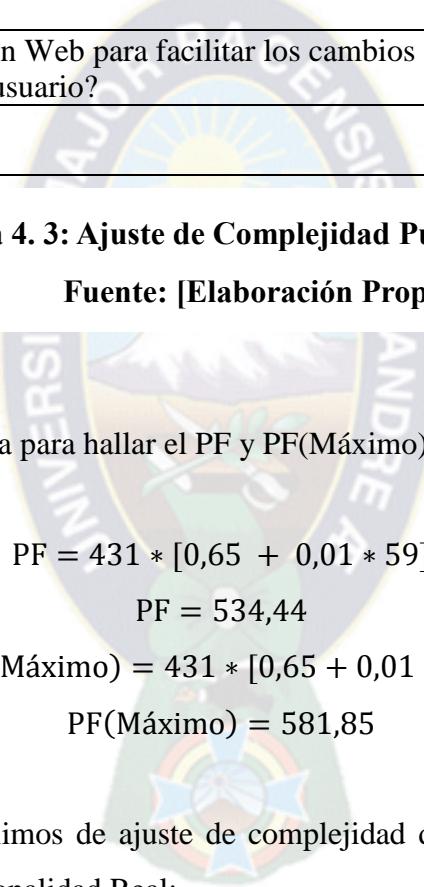
Escala	Complejidad
0	Sin influencia
1	Incidental
2	Moderado
3	Medio
4	Significativo
5	Esencial

Tabla 4. 2: Valores de ajuste de complejidad

Fuente: [Elaboración Propia]

Preguntas para la variable F_i se muestra en la tabla 4.3.

Pregunta	Ponderación
¿Requiere el Software copias de seguridad y de recuperación fiables?	5
¿Se requiere comunicación de datos?	5
¿Existen funciones de procesamiento distribuido?	3
¿Es crítico el rendimiento?	3
¿Se ejecuta el Software en un entorno operativo existente y fuertemente utilizado?	5
¿Requiere el Software entrada de datos interactiva?	3
¿Requiere la entrada de datos interactiva para que estas transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	5



¿Se actualizan los archivos maestros de forma interactiva?	5
¿Son complejos las entradas, las salidas, los archivos o las peticiones?	3
¿Es complejo el procesamiento interno?	3
¿Se ha diseñado el código para que sea reutilizable?	5
¿Están incluidas en el diseño la conversión y la instalación?	4
¿Se ha diseñado el Software para soportar múltiples instalaciones en diferentes organizaciones?	5
¿Se ha diseñado la Aplicación Web para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
P(Fi) Total	59

Tabla 4. 3: Ajuste de Complejidad Punto Función

Fuente: [Elaboración Propia]

Empleando la fórmula para hallar el PF y PF(Máximo):

$$PF = 431 * [0,65 + 0,01 * 59]$$

$$PF = 534,44$$

$$PF(\text{Máximo}) = 431 * [0,65 + 0,01 * 70]$$

$$PF(\text{Máximo}) = 581,85$$

Con los valores máximos de ajuste de complejidad de Punto Función, se tiene el siguiente resultado de Funcionalidad Real:

$$\text{FUNCIONALIDAD} = \left(\frac{534,44}{581,85} \right) * 100$$

$$\text{FUNCIONALIDAD} = 91,85 \%$$

Por tanto, la funcionalidad del Software se representa por el 91.85%, teniendo en cuenta el Punto Función (Máximo). Lo que quiere decir que el Software cumple con los requisitos funcionales de forma satisfactoria.

Es decir que en una escala de trabajo de 8 hrs. el software es capaz de ser operable aproximadamente 7 hrs. sin que este se caiga por razones de funcionalidad interna

4.1.2. USABILIDAD

Cuando se habla de usabilidad, se espera que el Software sea de fácil entendimiento y aprendizaje. Cabe resaltar que para la norma ISO 9126, la usabilidad no es afectada por la funcionalidad y eficiencia. La usabilidad está definida por los usuarios finales o asociados al Software.

Para la medición de la usabilidad se tiene el siguiente cuestionario con los ajustes necesarios que se pueden observar en las siguientes tablas:

Descripción	Escala
Pésimo	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Tabla 4. 4: Escala de ajuste de usabilidad

Fuente: [Elaboración propia]

Las preguntas utilizadas para medir la funcionalidad fueron las representadas en la tabla 4.5, con los siguientes valores obtenidos:

Factor	Valor
¿Se ha cubierto todos los requerimientos establecidos por el Software?	5
¿Es sencillo acceder a los datos?	5
¿Presenta suficiente ayuda durante el tiempo que accede al software?	4
¿Los informes son suficientemente representativos?	5
¿El Software tiene la seguridad necesaria?	5
¿Está de acuerdo con el funcionamiento del Software?	4
¿El Software facilita el trabajo que realiza?	5
Total	33

Tabla 4. 5: Evaluación de usabilidad

Fuente: [Elaboración Propia]

La usabilidad se calcula por medio de la siguiente fórmula:

$$\text{USABILIDAD} = \frac{\left[\left(\frac{\sum \text{Valor}}{n} \right) \times 100 \right]}{5}$$

$$\text{USABILIDAD} = \frac{\left[\left(\frac{33}{7} \right) \times 100 \right]}{5}$$

$$\text{USABILIDAD} = 94.28 \%$$

Por lo tanto, la usabilidad del software corresponde a un 94.28%, que se interpreta como la facilidad del usuario al interactuar con las interfaces.

Es decir que de 100 usuarios 94 fueron capaces de entender y operar el software, con un nivel de atracción alto en cuanto a la interfaz de usuario.

4.1.3. MANTENIBILIDAD

La mantenibilidad es la capacidad del sistema a ser modificado a nivel funcional, correcciones de mejoras y cambios en el entorno del Software.

Para la puntuación de la mantenibilidad, se desarrollaron las siguientes preguntas, mismas que fueron contestadas por el desarrollador del Software

Factor de ajuste	Valor
¿Puede ser modificado el Software?	95
¿Deja identificar las partes que deben ser modificadas?	95
¿Permite implementar una modificación específica?	97
¿No presenta efectos inesperados en posibles errores?	95
Total	95,5

Tabla 4. 6: Factores de ajuste de mantenibilidad

Fuente: [Elaboración Propia]

$$\text{MANTENIBILIDAD} = 95,5 \%$$

Al obtener un 95,5% aproximadamente en mantenibilidad podemos asegurar que el producto de Software contempla una cantidad de esfuerzo mínimo para cualquier cambio o modificación que requiera el Software.

Por lo tanto, existe una probabilidad de un 95,5 % de que un desarrollador pueda realizar cambios con facilidad en el código o estructura interna del software propuesto.

4.1.4. PORTABILIDAD

El hecho de que el producto Software esté desarrollado en el Lenguaje NodeJs y es en el entorno web nos ayuda de sobremanera en cuanto a su portabilidad, dado que únicamente es necesario tener un navegador instalado, por lo tanto la portabilidad es el de un 100%.

4.1.5. CALIDAD TOTAL

Para poder obtener la calidad total del Software, sacaremos la media de todas las medidas expresadas en porcentajes.

Características	Resultados
Funcionalidad	91,85
Usabilidad	94,28
Mantenibilidad	95,5
Portabilidad	100
Total	95,41

Tabla 4. 7: Resultados Calidad Total norma ISO 9126

Fuente: [Elaboración Propia]

La calidad total del Software corresponde al 95.41%, lo que se interpreta como la satisfacción que tiene el usuario al interactuar con el Software. En otras palabras, por cada 100 usuarios que manejaron el Software 95 a 96 quedaron satisfechos.

4.2. SEGURIDAD

4.2.1. POLÍTICAS DE SEGURIDAD (USUARIOS)

- **Administrador.** Puede acceder a todos los módulos del sistema, y tiene permisos de adición, actualización y eliminado de todos los registros.
Además, es el encargado de imprimir los reportes de pre inscripciones e inscripciones para los cursos de temporada, en el caso de la evaluación docente es quien asigna a los tribunales e imprime el reporte final de la evaluación.
- **Estudiante.** Puede registrar sus pre inscripciones e inscripciones al curso de temporada cuando esté habilitado, además de editar datos de su perfil.
- **Docente Titular.** Puede administrar su informe de docente anual cuando la evaluación esté habilitada, además de editar datos de su perfil.
- **Director de Carrera.** Puede registrar la ponderación del trabajo de los docentes titulares cuando la evaluación esté habilitada, además de editar datos de su perfil.
- **Tribunal.** Puede ponderar y observar registros de los informes docentes registrados cuando la evaluación esté habilitada, además de editar datos de su perfil, este únicamente puede ser un docente emérito.

4.2.2. POLÍTICAS DE SEGURIDAD ACCESO AL SOFTWARE

El sistema verifica la autenticidad del usuario y su contraseña, Los usuarios deben de guardar de forma segura su contraseña.

En caso de olvido este puede ser restaurado mediante el correo electrónico asociado al usuario.

El acceso a la base de datos es restringido para usuarios no autorizados mediante contraseña y nombre de usuario en el Servidor.

4.2.3. ENCRIPCIÓN DE CONTRASEÑAS

Es recomendable que las contraseñas de los usuarios estén encriptados, de esta forma se tiene asegurado que las acciones solamente sean únicamente responsables de los usuarios correctamente identificados. Para el sistema desarrollado se utilizó una función de NodeJs para encriptar cadenas de texto que hace uso de algoritmo de encriptación MD5 (Message digest 5).

Además, para la solicitud de datos por los servicios proporcionados se mantiene la autenticación del usuario mediante un token encriptado principal que es enviado por la cabecera de la petición a través del protocolo HTTPS, así mantenemos la seguridad en las peticiones.

4.2.4. CONTROL DE ACCESOS Y RESTRICCCIONES

Se desarrolló una funcionalidad para el control de accesos a ciertas partes del sistema y restricciones de acciones de acuerdo al rol que tenga asignado un usuario.



CAPÍTULO V

ANÁLISIS COSTO – BENEFICIO

5.1. ANÁLISIS DE COSTOS

En este punto cuantificaremos la inversión de los recursos que se empleó en el desarrollo del Software. Para el cálculo de esfuerzo y costo de desarrollo se utilizará el modelo COCOMO. II

5.2. COCOMO II

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito, pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. COCOMO II sigue los principios de apertura usados en el COCOMO original de esta manera todos sus algoritmos y relaciones están disponibles públicamente

5.2.1. MÉTRICAS DE SOFTWARE

Se utilizó como punto función $PF = 534,44$ que se obtuvo en el apartado 4.1.1.

A continuación, realizamos la conversión de punto función a miles de líneas de código mediante la siguiente tabla:

Lenguaje de Programación	Factor LDC/PF
Java	53
JavaScript	47
Visual Basic	46
ASP	36
Visual C++	34
PHP	12
Ensamblador	320
C	150

Tabla 5. 1: Factor LCD/PF de lenguajes de programación

Fuente: [QSM, 2017]

Ahora aplicamos la fórmula para calcular las líneas de código.

KLDC: Número estimado de líneas de código distribuidas.

LCD: PF * Factor LDC

$$LCD = 534,44 * 47 = 25118,68$$

Las líneas de código en su totalidad son 25118,68, de las cuales se estima que aproximadamente un 90% del código es reutilizable, en ese entendido el LCD total es:

$$KLCD = (LCD - LCD \text{ reutilizable}) / 1000$$

$$KLCD = (25118,68 - 22606,81) / 1000$$

Por tanto (Miles de líneas de código) KLDC = 2,51

5.2.2. ESTIMACIÓN DEL ESFUERZO

El esfuerzo necesario para concretar un proyecto de desarrollo de software, cualquiera sea el modelo empleado, se expresa en meses/persona (PM) y representa los meses de trabajo de una persona full-time, requeridos para desarrollar el proyecto.

El modelo usado para calcular el esfuerzo es el propuesto por COCOMO.

$$E = a_b(KLDC)^{bb}, \text{en peronas por mes}$$

E: Esfuerzo aplicado por persona (KLDC: Número líneas de código)

$$D = C_b(E)^{dd}, \text{en meses}$$

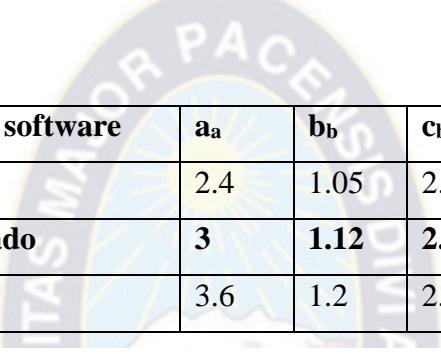
D: Tiempo de desarrollo en meses

Se utiliza para obtener una primera aproximación rápida del esfuerzo y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

Se debe de considerar el tipo de proyecto adecuado para el cálculo de las constantes, entre los cuales tenemos:

- **Orgánicos.** Relativamente sencillos y pequeños, en los que trabajan equipos pequeños con experiencia, sobre un conjunto de datos poco rígidos.

- **Semi-acoplados.** Proyectos intermedios (en tamaño y complejidad) en los que participan equipos con variados niveles de experiencia, deben satisfacer requisitos medio rígidos.
- **Empotrados.** Proyectos bastante complejos, en los que apenas se tienen experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.



Proyecto de software	a_a	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Tabla 5. 2: Modelo básico para tipos de proyectos.

Fuente: [Gomez y Migani, 2007]

Aplicando las fórmulas anteriormente descritas y tomando en cuenta un tipo de proyecto orgánico se tiene lo siguiente:

$$E = 3 * (2,51)^{1.12} = 8,42$$

$$D = 2.5 * (8,42)^{0.35} = 5,27 = 5$$

El personal requerido será igual a: Número de programadores = E/D

Por tanto: Número de Programadores = $8,42 / 5,27 = 1,59 = 2$

5.2.3. COSTO DE SOFTWARE

Tomando como base el salario aproximado de un programador 500 USD, esta cifra se toma en cuenta para estimar el costo del software:

$$\text{Costo Soft} = \text{Nro Programadores} * \text{salario Programador} * \text{duración} = 2 * 500 \text{ USD} * 6 = 6.000\text{\$}$$

5.2.4. COSTO DE ELABORACIÓN DEL PROYECTO

Los costos de elaboración del proyecto se detallan a continuación:

Descripción	Costo Total (USD)
Análisis y Diseño del proyecto	500
Bibliografía	30
Material de escritorio	40
Otros	30
Total	600

Tabla 5. 3: Coste de elaboración

Fuente: [Elaboración propia]

5.2.5. COSTE DE IMPLEMENTACIÓN DEL PROYECTO

La carrera como parte de la Universidad Mayor de San Andrés, se le fue proporcionada una máquina virtual en los servidores centrales, por esa razón el costo es 0.

5.2.6. COSTE TOTAL

El coste total del sistema en USD es la suma de los costos, los cuales se presentan en la siguiente tabla:

Descripción	Costo Total (USD)
Costo de desarrollo	6000
Coste de elaboración	600
Costo de implementación	0
Total	6600

Tabla 5. 4: Coste total

Fuente: [Elaboración propia]

5.3. ANÁLISIS DE BENEFICIOS

En un proyecto es muy importante analizar la posible rentabilidad de dicho proyecto y sobre todo si es viable o no. Cuando se elaborar un proyecto hay que invertir un capital y se espera obtener una rentabilidad a lo largo de los años. Esta rentabilidad debe ser mayor al menos que una inversión con poco riesgo.

El análisis de los beneficios del proyecto se emplea 5 criterios de evaluación, en el cual es posible comparar y evaluar el nuevo sistema con respecto al anterior.

- Incremento de velocidad
- Capacidad en el volumen de información

- Control de procesos
- Integración de la información
- Información para la toma de decisiones
- Valor neto actual

5.3.1. VALOR ACTUAL NETO

El Valor Actual Neto o (VAN), es un indicador financiero que mide los flujos de los futuros ingresos y egresos que tendrá un proyecto, para determinar si luego de desconectar la inversión inicial, nos quedara alguna ganancia. Si el resultado es positivo, el proyecto es viable.

De este modo si se tiene varios proyectos sobre la mesa el VAN nos permitirá determinar cuál proyecto es el más rentable entre varias opciones de inversión.

Si $VAN > 0$: El proyecto es rentable

Si $VAN = 0$: El proyecto es rentable, pero ya está incorporando ganancia de la TD.

Si $VAN < 0$: El proyecto no es rentable

La fórmula para calcular el van es la siguiente: $VAN = \sum_{t=1}^n \frac{Ganancias}{(1+k)^t} - \sum_{t=1}^n \frac{Costos}{(1+k)^t}$

N es el número de periodos considerados.

K la tasa de descuento o tasa de interés al préstamo.

Los Gastos y ganancias que se estiman en un lapso de 4 se muestran en la siguiente tabla, para el presente caso se utilizará una tasa de descuento del 10%.

Año	Costos	Ganancias	Costos/(1+k) ⁿ	Ganancias/(1+k) ⁿ	Resultado
0	6600	0	6600	0	-6600
1	0	3500	0	3181,82	3181,82
2	0	4500	0	3719,01	3719,01
Total	6600	8000	6600	6900,83	300,83

Tabla 5. 5: Cálculo del VAN

Fuente: [Elaboración propia]

Así,

$$VAN = \sum \frac{Ganancias}{(1 + 0,14)} - \sum \frac{Costos}{(1 + 0,14)}$$

$$VAN = 300,83$$

Por lo tanto, podemos afirmar que nuestro proyecto es rentable.

5.3.2. COSTO/BENEFICIO

Para hallar el costo – beneficio de nuestro proyecto aplicamos la siguiente ecuación:

$$\text{Costo Beneficio} = \frac{\sum \text{Ganancias}}{\sum \text{Costos}}$$

$$Costo Beneficio = 1,21$$

Es decir que por cada USD invertido en el proyecto de Software, la institución tendrá una ganancia de 0.21 USD.

5.3.3. TASA INTERNA DE RETORNO

Cuando el VAN toma un valor igual a 0, la variable k pasa a llamarse TIR. Que es la rentabilidad que nos proporcionará el proyecto.

La siguiente ecuación calcula el VAN:

$$VAN = \text{Inversión} + \sum \frac{\text{Ganancias}}{(1 - k)^n}$$

Tenemos:

$$VAN = 6600 + \frac{3500}{1 + TIR} + \frac{4500}{(1 + TIR)^2}$$

$$(1 + TIR)^2 * 0 = -6600 * (1 + TIR)^2 + 3500 * (1 + TIR) + 4500$$

$$1 + TIR = \frac{-3500 \pm \sqrt{3500^2 - 4 * (-6600) * 4500}}{2 * (-6600)}$$

$$1 + TIR = -0,60 \text{ y } 1,13$$

Se toma en cuenta el positivo ya que no es posible tener una tasa negativa.

$$TIR = 1,13 - 1 = 0,13$$

$$TIR = 0,13 * 100 = 13\%$$

El tipo de interés ofrecido debe de ser menor o igual al 13%.



CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

A partir del cumplimiento de los requerimientos establecidos por los Usuarios del Sistema pertenecientes a la carrera de ingeniería industrial, se ha cumplido con los objetivos planteados por medio de la implementación del Sistema de administración de cursos de temporada y evaluación docente (Sauron) a través de los diferentes módulos desarrollados para este proyecto.

Por tanto, se llega a las siguientes conclusiones:

- Se implementó sistema web de administración para evaluación docente y cursos de temporada, que ayuda a simplificar el proceso y mantiene informados a los actores del mismo.
- Se implementó un sistema que toma en cuenta la escalabilidad y posterior integración con otros sistemas utilizando web services para su comunicación.
- Se diseñaron las interfaces de registro acorde a formularios oficiales para su llenado de manera simplificada y correcta.
- Se desarrollaron los módulos con un adecuado mecanismo de registro basado en las normas establecidas por el Reglamento de Evaluación Docente.

- Se generaron reportes finales y medios que proporcionar información para una correcta toma de decisiones en ambos procesos.
- Se desarrollaron los módulos de pre inscripción e inscripción que facilitan la administración, organización y control de los cursos de temporada y disminuyen los procesos presenciales.
- En cuanto a la población estudiantil se redujo considerablemente la desinformación sobre los cursos de temporada y la consulta presencial en oficinas de Kardex.

6.2. RECOMENDACIONES

Para ampliar el presente proyecto de grado, se hacen las siguientes recomendaciones:

- Realizar ampliaciones en cuanto la funcionalidad del Software a más procesos y extenderlos a más áreas de la carrera y/o facultad.
- Aprovechar el uso de las nuevas herramientas de desarrollo aplicadas en este proyecto para futuros desarrollos de Software tanto web como móviles.
- Debido al manejo de Documentación física se recomienda la inclusión de políticas para el uso de documentos digitales en ambos procesos, y así acoplar el registro en el sistema.
- Desarrollar un módulo para la exportación y gestión de notas del curso de temporada hacia los sistemas académicos de la facultad de ingeniería.

BIBLIOGRAFÍA

- ALAIMO, M. 2013. Proyectos ágiles con scrum. Argentina: Keleer. 123p
- ARCONADA, F. 2016. Comunicación HTTP de Microservicios. La Rioja, Universidad de La Rioja. 63p
- BESTBIPRACTICES. 2015. Breve resumen sobre SCRUM [en linea] <<https://bestbipractices.wordpress.com/2013/07/10/breve-resumen-sobre-scrum/>> [consulta: 01 de junio de 2017]
- COCOMOII. (14 de Abril de 2016). Obtenido de TEMA 7: http://www.eici.ucm.cl/Academicos/ygomez/descargas/Ing_Sw2/apuntes/cocomo_anual_espanol.pdf
- ERASO, J. 2013. Aplicación para la gestión de proyectos ágiles con Scrum. La Rioja, Universidad de La Rioja. 51p
- FERNANDEZ, Y., & Díaz, Y. 2012. Patrón Modelo-Vista-Controlador. Obtenido de <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/15/10>
- FLORES, S. 2016, Sistema de control y generación de partes de asistencia de docentes y carga horaria académica. La Paz, Universidad Mayor de San Andrés, Facultad de Ciencias Puras y Naturales. 66p
- NOLIVOS, G., CORONEL, F., & CAMPAÑA, M. 2010. Implementación de un Sistema Web para el Control de un Taller Técnico Automotriz en Plataforma PHP - MySQL utilizando UWE para la Empresa METROAUTOCERFRAN Cia. Ltda. Ecuador: Escuela Politécnica del Ejército.
- PALACIO, J. 2014. Gestión de Proyectos Scrum Manager. Obtenido de http://www.scrummanager.net/files/sm_proyecto.pdf
- PEREZ-HERRERA, M. 2015, Arquitecturas basadas en microservicios. Madrid, E.T.S.I. Telecomunicación (UPM). 50p

PRESSMAN, R. 2012. Ingeniería de Software, un enfoque práctico. 7° ed. México, McGraw-Hill Interamericana Editores. 777p

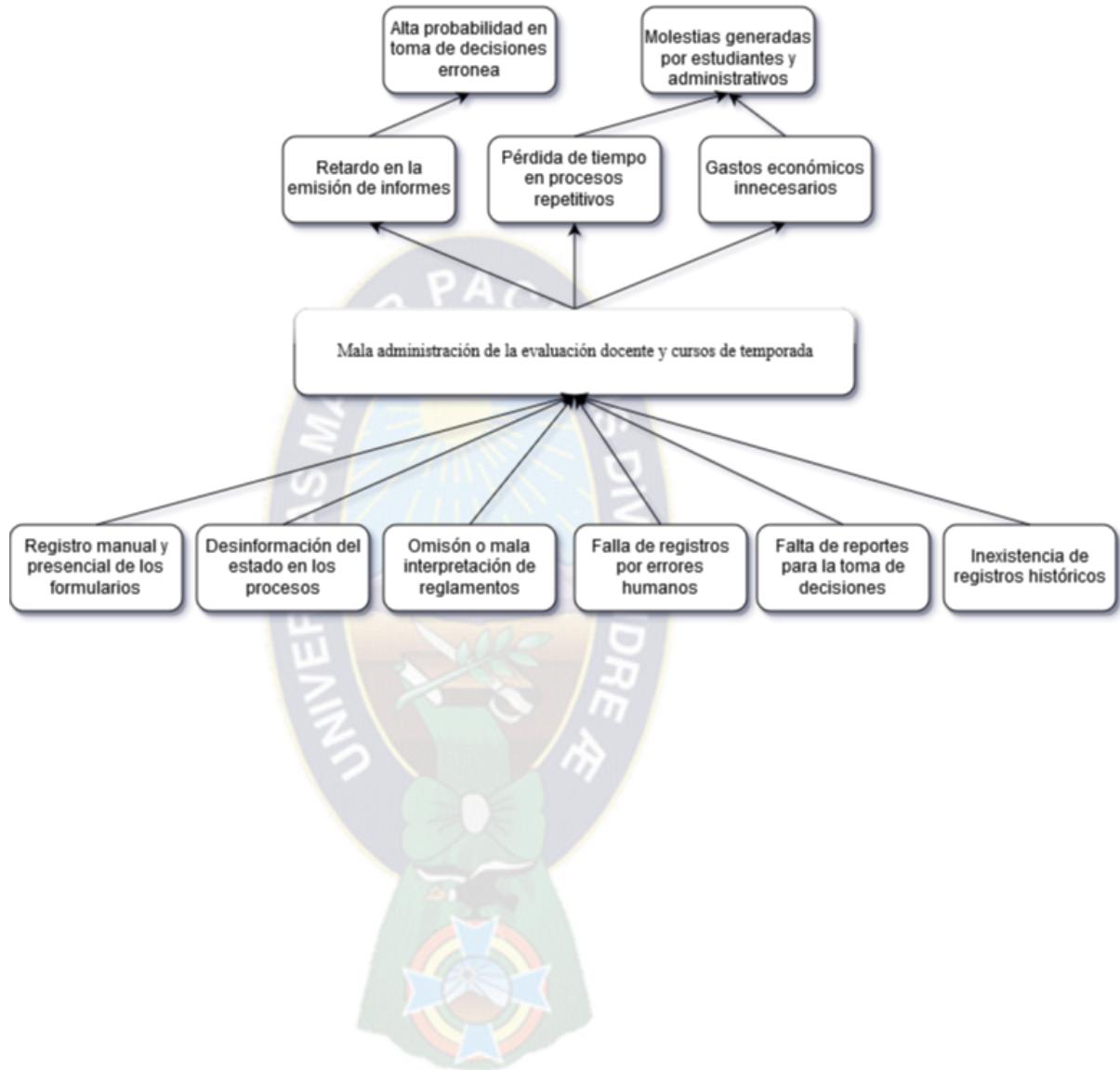
QSM, 2017. Quantitative Software Management. Obtenido de Quantitative Software Management: <http://www.qsm.com/resources/function-point-languages-table>

RODRIGUEZ, A. 2009. Metodologías de diseño usadas en la ingeniería web, su vinculación con las ntics. La Plata, Universidad Nacional de La Plata, Facultad de Informática. 58p

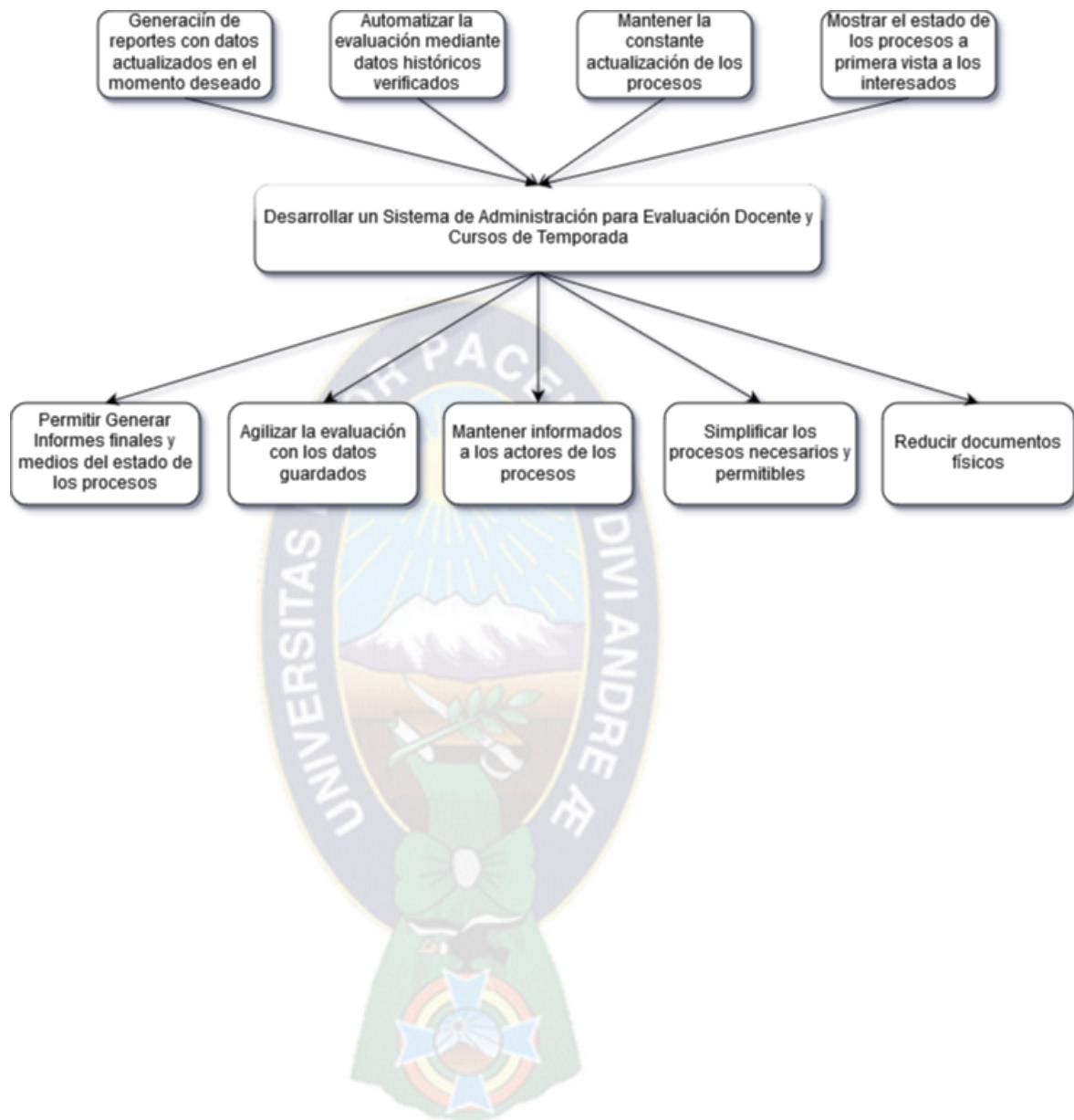


ANEXOS

ANEXO A - ÁRBOL DE PROBLEMAS



ANEXO B - ÁRBOL DE OBJETIVOS



CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	DURACIÓN EN DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																			
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Redacción del Capítulo II - Marco Teórico	10																				
Desarrollo del Capítulo III - Marco Aplicativo	60																				
Pre-Game	21																				
Game (1ra. iteración)	14																				
Game (2da. iteración)	14																				
Game (3ra. iteración)	14																				
Post-Game	14																				
Redacción del Capítulo III - Marco Aplicativo	21																				
Redacción del Capítulo IV - Calidad y Seguridad	14																				
Redacción del Capítulo V - Análisis Costo Beneficio	14																				
Redacción del Capítulo VI - Conclusiones y Recomendaciones	7																				



DOCUMENTOS

