



“INSTITUTO SUPERIOR MINERO METALÚRGICO”

Dr. Antonio Núñez Jiménez

Facultad Metalurgia - Electromecánica

Moa, Holguín

Trabajo de diploma para optar por el título de
Ingeniero en Informática

“SISTEMA DE GESTIÓN DE LOS CORTES EVALUATIVOS”

Autor:

Joandris Rivera Legrá

Tutor:

Ing. Dabiel González Ramos

Moa-Cuba

Julio – 2011

Año 53 de La Revolución

DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez” para que hagan el uso que estimen pertinente del mismo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del 2011.

Joandris Rivera Legrá

Firma del autor

Ing. Dabiel González Ramos

Firma del tutor



“En tiempos teológicos universidad teológica, en tiempos científicos universidad científica, al mundo nuevo corresponde la universidad nueva”.

José Martí

Agradecimiento:

Muchas son las personas que de una forma u otra me han ayudado a que se haga realidad la realización de este trabajo. Profundo agradecimiento quiero expresarle a mi tutor Dabiel González, por guiarme en lo correcto con sus sugerencias, exigencia y confianza y por contribuir con mi gran sueño de verme hecho Ingeniero. No puedo olvidar el claustro de profesores que día a día me fueron llenando de conocimientos con sus clases y se preocuparon por que yo fuera en el futuro un excelente profesional.

La amistad es algo muy valioso en el mundo, un amigo siempre está ahí en los momentos buenos y malos. Entre tantas personas quiero agradecerle ahora y siempre a todos los compañeros de mi grupo, a mis más sinceros amigos: Arnold Torrell Delgado, Yasmany Chávez, Ernesto Laffita, Zeydis Borges. Un saludo muy especial a mi gran amigo, que más que amigo es mi hermano, Getsy Marzo Méndez por ayudarme en todo lo que pudo y estar ahí siempre que lo necesité...

La madre es lo más grande que uno tiene en la vida por eso quiero agradecerle a mi mamá por confiar siempre en mí, por estar ahí, preocupada en todo lo que de mí se trata y por todo lo que me inculcó y me aconsejó.

Quiero agradecerle a mi papá por estar pendiente de mis actos, aconsejarme siempre que hacía algo que no fuese lo correcto, por preocuparse y darme todo el apoyo que necesité....

Dedicatoria:

Agradezco haber seguido los consejos de mi madre, que me ayudó a ver el camino y a ser lo que soy hoy, por eso le dedico mi esfuerzo, mi trabajo y mi existir.

No se puede vivir sin amor de madre pero tampoco sin el amor de la pareja, por eso quiero dedicarle este trabajo que es mi ser a mi esposa, quien me brinda su amor, su tiempo y la que merece más que un beso, lo merece todo por parte mía en esta vida.

Le dedico este trabajo a mi gran amigo Getsy Marzo quien ha depositado en mí todo lo que un amigo debe creer y me ha juzgado y señalado cuando hago algo mal.

Quiero dedicarles esto a todas las personas que de una forma u otra me han ayudado y han contribuido a que yo sea lo que soy hoy, y a toda mi familia que siempre está conmigo para apoyarme en lo que necesite y en todo momento.

Gracias.

Resumen

Con el desarrollo de las tecnologías a nivel mundial, se ha convertido en una necesidad la utilización de aplicaciones web y portales para las empresas, instituciones sociales y demás entidades, ya que les permite entregar información actualizada a sus clientes, gestionando la misma para hacerla accesible al usuario que la necesite. En el Departamento Informática del Instituto Superior Minero Metalúrgico de Moa (ISMMM) surgió la necesidad de informatizar el proceso del Corte Evaluativo de los estudiantes mediante la creación de una aplicación Web, ya que esta actividad no es realizada de forma automatizada lo que provoca demora durante el proceso de gestión de la información. Con este sistema trataremos de solucionar los problemas que existen actualmente sobre los Cortes Evaluativos. La aplicación muestra funcionalidades muy importantes como la recogida de datos de todos los estudiantes de la carrera que se presenta en la actualidad, en Excel y documentos Word (nombre, apellidos y las dificultades que presentan), datos generales de todos los años. El sistema debe brindar reportes de los resultados. Un módulo de administración, debe presentar una propuesta con datos iniciales que sirva como guía para los Cortes Evaluativos.

Abstract

With the development of them technology to level worldly, the utilization of web application and porch for them enterprise, social institution and other entity, has become in a necessity, since that them permit to deliver present information to her client, to procure it is making accessible at user that the need. At the Department Computer Superior Institute Minero Metalúrgica of Moa's (ISMMM) rose the need to computerize the process of the evaluative cut of the students intervening the creation of Web application, since this activity is not accomplished of automated form what he provokes takes time in the process of steps of information. With this system we will try to solve of the present-day problems on the evaluative cut. Application evidences functionalities very important. Example: the retirement of date of all them students of the career that it presents in the actuality in excel and document word (name, last name, deep that present) general date of all them years. System must offer reports the result. A module of administration, should to present a proposition with initials date that he be good at the evaluative cut.

Contenido

Introducción.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 ¿Qué son los Cortes Evaluativos?	6
1.2 Funcionamiento de los Cortes Evaluativos.....	6
1.2.1 Estructura de los Cortes Evaluativos.....	7
1.3 Estrategia de los Cortes Evaluativos:.....	8
1.4 Beneficios y resultados esperados:.....	9
1.5 Antecedentes	9
1.6 Propuesta de Solución	9
1.7 Tendencias y tecnologías actuales:	10
1.7.1 Lenguaje de programación:.....	10
1.7.2 Justificación del lenguaje de programación a utilizar.....	12
1.7.3 Framework de desarrollo.....	13
1.7.4 Sistemas Gestores de Base de Datos (SGBD)	13
1.7.5 Justificación del SGBD a utilizar.....	16
1.7.6 Entorno de trabajo.....	18
1.8 Metodologías de desarrollo de software.....	18
1.8.1 Metodologías ágiles.....	18
1.8.2 XP (Extreme Programming)	20
1.8.2.1 Valores de la metodología XP	20
1.8.2.2 Prácticas de la metodología XP.....	21
1.8.2.3 Fases de la metodología XP.....	23
1.8.3 Scrum.....	25
1.8.4 ¿Por qué aplicar SXP?	26
1.8.4.1 Fases de la metodología SXP	27
1.9 Patrones Arquitectónicos.....	29
1.10 Conclusiones.....	32
CAPÍTULO 2: PLANIFICACIÓN - DEFINICIÓN	33
2.1 Introducción.....	33
2.2 Planificación del proyecto por roles.....	33
2.3 Planificación y Definición del proyecto.	35
2.3.1 Plantilla Concepción del sistema.....	35
2.3.2 Plantilla Modelo de Historias de usuario del negocio.	35
2.3.3 Plantilla Lista de Reserva del Producto (LRP).....	36
2.3.4 Plantilla Historias de usuario	36
Estimación de esfuerzo	37
2.3.5 Plantilla Lista de riesgos.....	38
2.4 Diseño	38
2.4.1 Plantilla Modelo de diseño.....	39
2.5 Planificación de las iteraciones	39
2.5.1 Primera iteración:	40
2.5.2 Segunda iteración:.....	40
2.5.3 Tercera iteración:.....	40
2.5.4 Cuarta iteración:	41
2.5.5 Quinta iteración:	41
2.6 Conclusiones.....	42

CAPÍTULO 3: DESARROLLO	43
3.1 Introducción.....	43
3.2 Diseño de la Solución Propuesta	43
3.2.1 Tarjetas CRC.....	43
3.3 Desarrollo del proyecto.....	44
3.3.1 Plantilla Tarea de ingeniería.....	44
3.3.2 Primera iteración	44
3.3.3 Segunda iteración.....	45
3.3.4 Tercera iteración.....	46
3.3.5 Cuarta iteración	46
3.3.6 Plantilla Cronograma de producción.....	47
3.4 Conclusiones.....	47
CAPITULO 4: PRUEBAS	48
4.1 Introducción.....	48
4.2 Pruebas de software	48
4.2.1 Desarrollo Dirigido por Pruebas.....	48
4.3 Objetivos de las pruebas.....	49
4.4 Plantilla Caso de prueba de aceptación	49
4.5 Conclusiones.....	50
CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD	51
5.1 Introducción.....	51
5.2 Efectos económicos	51
5.2.1 Efectos directos.....	51
5.2.2 Efectos indirectos	52
5.2.3 Intangibles	52
5.3 Beneficios y Costos Intangibles en el proyecto	53
5.4 Ficha de costo	53
5.5 Conclusiones.....	56
CONCLUSIONES GENERALES	57
RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS	59
GLOSARIO DE TÉRMINOS	60
ANEXOS	62
Anexo 1: Plantilla Concepción del Sistema	62
Anexo 2: Plantilla Modelo de HU del Negocio.....	64
Anexo 3: Plantilla Lista de Reserva del Producto.....	64
Anexo 4: Plantilla de Historias de Usuarios.....	66
Anexo 5: Plantilla Lista de Riesgos	70
Anexo 6: Plantilla Modelo de Diseño	71
Anexo 7: Tarjetas CRC	71
Anexo 8: Tareas de Ingeniería	74
Anexo 9: Pruebas de Aceptación	79
Anexo 10: Modelo de la Base de Datos	84

Índice de Tablas.

Tabla_1. Tiempos de las iteraciones. -----	41
Tabla_2. Tarjeta CRC #1 Principal -----	43
Tabla_3. Historias de Usuario primera iteración-----	45
Tabla_4. Tareas de ingeniería # 1 Crear cuentas de Usuarios -----	45
Tabla_5. Historias de Usuario segunda iteración -----	45
Tabla_6. Historias de Usuario tercera iteración-----	46
Tabla_7. Historias de Usuario cuarta iteración-----	46
Tabla_8. Prueba de aceptación #1 para la HU “Gestionar Usuarios” -----	49
Tabla_9. Plantilla Concepción del Sistema -----	62
Tabla_10. Actores del negocio -----	64
Tabla_11. Plantilla Lista de Reserva del Producto -----	64
Tabla_12. Plantilla Historia de usuario 1.-----	66
Tabla_13. Plantilla Historia de usuario 2.-----	66
Tabla_14. Plantilla Historia de usuario 3.-----	67
Tabla_15. Plantilla Historia de usuario 4.-----	67
Tabla_16. Plantilla Historia de usuario 5.-----	67
Tabla_17. Plantilla Historia de usuario 6.-----	68
Tabla_18. Plantilla Historia de usuario 7.-----	68
Tabla_19. Plantilla Historia de usuario 8.-----	69
Tabla_20. Plantilla Historia de usuario 9.-----	69
Tabla_21. Plantilla Historia de usuario 10.-----	69
Tabla_22. Plantilla Lista de Riesgos -----	70
Tabla_23. Plantilla Modelo de Diseño -----	71
Tabla_24. Tarjeta CRC #2 Ges_datos_pri -----	71
Tabla_25. Tarjeta CRC#3 Auten_usuario -----	72
Tabla_26. Tarjeta CRC#4 Gestión_asis -----	72
Tabla_27. Tarjeta CRC#5 Ges_eva_pa_es -----	72
Tabla_28. Tarjeta CRC #6 Ges_eva_pa_areas -----	72
Tabla_29. Tarjeta CRC #7 Ges_eva_gen -----	73
Tabla_30. Tarjeta CRC #8 Informaciones-----	73
Tabla_31. Tarjeta CRC #9 Resultados -----	73
Tabla_32. Tarjeta CRC #10 Gestion_repor -----	74
Tabla_33. Tareas de ingeniería # 2 Modificar cuentas de Usuarios -----	74
Tabla_34. Tareas de ingeniería # 3 Eliminar cuentas de usuarios-----	74
Tabla_35. Tareas de ingeniería # 4 Cargar Datos-----	74
Tabla_36. Tareas de ingeniería # 5 Mostrar datos-----	75
Tabla_37. Tareas de ingeniería # 6 Entrar al sistema -----	75
Tabla_38. Tareas de ingeniería # 7 Cambiar contraseña-----	75
Tabla_39. Tareas de ingeniería # 8 Insertar asistencia -----	75
Tabla_40. Tareas de ingeniería # 9 Mostrar asistencia -----	75
Tabla_41. Tareas de ingeniería # 10 Modificar asistencia -----	76
Tabla_42. Tareas de ingeniería # 11 Insertar evaluación de los estudiantes ---	76
Tabla_43. Tareas de ingeniería # 12 Mostrar evaluación de los estudiantes ---	76
Tabla_44. Tareas de ingeniería # 13 Modificar evaluación de los estudiantes- 	76
Tabla_45. Tareas de ingeniería # 14 Insertar evaluaciones de las áreas -----	77
Tabla_46. Tareas de ingeniería # 15 Mostrar evaluaciones del las áreas -----	77

Tabla_47. Tareas de ingeniería # 16 Modificar evaluaciones de las áreas-----	77
Tabla_48. Tareas de ingeniería # 17 Dar evaluación -----	77
Tabla_49. Tareas de ingeniería # 18 Mostrar evaluación -----	78
Tabla_50. Tareas de ingeniería # 19 Insertar información-----	78
Tabla_51. Tareas de ingeniería # 20 Mostrar información-----	78
Tabla_52. Tareas de ingeniería # 21 Modificar información -----	78
Tabla_53. Tareas de ingeniería # 22 Mostrar gráficas -----	79
Tabla_54. Tareas de ingeniería # 23 Comparar resultados-----	79
Tabla_55. Tareas de ingeniería # 24 Brindar reportes -----	79
Tabla_56. Prueba de aceptación #2 para la HU “Gestionar Datos Primarios” -	79
Tabla_57. Prueba de aceptación #3 para la HU “Autenticar usuarios” -----	80
Tabla_58. Prueba de aceptación #4 para la HU “Gestionar asistencia” -----	80
Tabla_59. Prueba de aceptación #5 para la HU “Gestionar evaluación parcial de los estudiantes” -----	81
Tabla_60. Prueba de aceptación #6 para la HU “Gestionar evaluación parcial de las áreas” -----	81
Tabla_61. Prueba de aceptación #7 para la HU “Mostrar evaluación” -----	82
Tabla_62. Prueba de aceptación #7 para la HU “Gestionar información” -----	82
Tabla_63. Prueba de aceptación #8 para la HU “Gestionar gráficas de los resultados” -----	83
Tabla_64. Prueba de aceptación #9 para la HU “Gestionar reportes” -----	83

Introducción

Desde el Triunfo de la Revolución nuestro Comandante en Jefe Fidel Castro aseguró que el futuro de nuestro país tenía que ser necesariamente un futuro de hombres de ciencia. Para hacer realidad estas aspiraciones se incrementa el número de centros e instituciones a lo largo de todas las provincias. Las universidades cubanas han dado pasos firmes en la preparación profesional de sus estudiantes, cada día se va consolidando su papel activo dentro de la sociedad, se eleva la calidad en su formación y se vinculan activamente a las necesidades más apremiantes del desarrollo económico y social de la nación.

En el mundo de hoy, cada vez más globalizado por las telecomunicaciones, la informática juega un papel decisivo en cualquier esfera o actividad, esta se ha ido desarrollando y sufriendo continuos cambios que inciden en nuestra sociedad de forma positiva.

En la actualidad, todas las entidades en su mayoría utilizan las tecnologías de la Informática ya que éstas aportan soluciones que permiten actuar con rapidez, sacar el máximo rendimiento del personal y tomar decisiones. Las Tecnologías de la Informática, por su desarrollo y sus avances, han posibilitado el incremento rápido del desarrollo de la economía y la sociedad. Este desarrollo viene dado gracias a la ambición del conocimiento del ser humano y son aplicados en gran medida en nuestra sociedad mediante la producción de software.

Nuestro país no está ajeno al desarrollo de software y en los últimos años ha avanzado en busca de la informatización de la sociedad. Las Universidades son las principales artífices en el desarrollo de software, sometidas a procesos de cambios y transformaciones ante los retos que le plantea la actual sociedad del conocimiento. La importancia del vínculo entre las universidades, las entidades de producción y la sociedad en general, forman parte del desarrollo de nuestro país.

En el Instituto Superior Minero Metalúrgico de Moa, el Departamento de Informática se percató de la necesidad de elaborar un software que facilite el control, rapidez y calidad de los Cortes Evaluativos que se les realizan a los estudiantes. Este se hace

con el objetivo de controlar la trayectoria de los estudiantes en las asignaturas impartidas.

La periodicidad de los informes de Corte Evaluativo se fijará en las Indicaciones para la planificación del curso, que se emiten por la Vicerrectoría Docente.

Como vía para la retroalimentación del desarrollo del proceso docente educativo y la regulación del proceso en cada instancia, se requiere disponer de los informes de las facultades, elaborados por los decanos o en quien delegue, a partir de las informaciones de los departamentos de carreras.

Estos Cortes Evaluativos tendrán en cuenta los resultados de las evaluaciones frecuentes, parciales o finales realizadas en el período que se informa.

Los estudiantes hasta el momento del Corte Evaluativo, se evaluarán de forma cualitativa en Bien, Regular o Mal de acuerdo a su desempeño.

Los coordinadores de años serán los máximos responsables ante el jefe del departamento de compilar toda la información del colectivo de año, para que en los departamentos docentes terminales, se hagan las reuniones para los análisis pertinentes y lleguen al decanato unificadamente por carreras.

Los informes se realizarán con el modelo que se adjunta en este sistema con vistas a unificar la información de la carrera.

Como consecuencia de lo planteado anteriormente se tiene la siguiente **situación problemática:**

Los Cortes Evaluativos juegan un papel de gran importancia en las universidades por ser un control académico de las evaluaciones de los estudiantes que ayuda a los profesores a determinar cuántos estudiantes se encuentran mal académicamente y qué medidas tomar para solucionar este problema, esta actividad se realiza de forma digital pero no con la organización y seguridad que debe poseer, lo que provoca serias dificultades durante el proceso de gestión de las informaciones, demora en la entrega de los resultados y falta de calidad en el desarrollo, por tal motivo en el

Instituto Superior Minero Metalúrgico de Moa (ISMMM), el Departamento de Informática se propuso automatizar todas estas tareas para facilitar un mejor funcionamiento y calidad del proceso.

Dada esta situación se define como **problema científico**:

La no existencia de un sistema automatizado que permita la evaluación y el control de los resultados de los estudiantes del ISMMM en los Cortes Evaluativos.

El objeto de estudio que se define es la informatización del sistema de Cortes Evaluativos.

El **campo de acción** que persigue es la informatización del proceso de los Cortes Evaluativos en el ISMMM.

De manera que el objetivo general de la investigación es:

Desarrollar una aplicación Web que permita la gestión, evaluación y el control de las informaciones y resultados de los estudiantes del ISMMM en los Cortes Evaluativos.

Para nuestra investigación la **idea a defender** es la siguiente:

La implementación de un sistema automatizado para la gestión del proceso de Cortes Evaluativos permitirá mayor eficiencia, calidad y rapidez del mismo.

Para lograr un mejor desarrollo de la investigación y darle seguimiento al objetivo trazado se plantearon las siguientes **tareas**:

- Detallar el proceso actual de evaluación de los Cortes Evaluativos.
- Fundamentación y análisis bibliográfico que permita la familiarización con las principales tecnologías y herramientas utilizadas actualmente para el desarrollo de aplicaciones informáticas.
- Desarrollar cada una de las etapas de la ingeniería de software hasta la implementación de un sistema que permita la gestión de los Cortes Evaluativos con la calidad requerida.
- Prueba y Documentación del Sistema.
- Desarrollo del Manual de Usuarios.

Para el cumplimiento de las tareas, los métodos de investigación empleados fueron Métodos Teóricos y Métodos Empíricos.

Los **Métodos Empíricos** empleados son:

- **Entrevistas:** Nos permite obtener la información adecuada para adentrarnos más en el problema, en las necesidades del Instituto y así determinar los principales requerimientos del sistema.
- **Observación:** Se utilizó para analizar el funcionamiento que tienen los Cortes Evaluativos.

Los **Métodos Teóricos** que sustentan la investigación:

- **Análisis-Síntesis:** Para la recopilación y el procesamiento de la información obtenida en los métodos empíricos y arribar a las conclusiones de la investigación.
- **Histórico-lógico:** Para investigar el desarrollo que ha tenido el tema y apoyar los conocimientos que sobre este existen.

El documento consta de 5 capítulos:

Capítulo 1: “FUNDAMENTACIÓN TEÓRICA”: Se exponen conceptos y criterios necesarios para el entendimiento del problema, además se realiza un estudio del proceso de los Cortes Evaluativos y una caracterización de las metodologías de desarrollo a utilizar, el lenguaje de programación, los sistemas gestores de base de datos, las herramientas para el desarrollo del software y las técnicas empleadas durante el proceso de desarrollo.

Capítulo 2: “PLANIFICACIÓN - DEFINICIÓN”: En este capítulo se presenta parte de la propuesta solución de la investigación, haciendo uso de la metodología de desarrollo SXP propuesta por la Universidad de Ciencias Informáticas en el que se aborda la fase de planificación y definición en la que se explica toda la dinámica del proyecto basándose en el expediente del proyecto.

Capítulo 3: “DESARROLLO”: En este capítulo se abordan los elementos pertenecientes a la fase de desarrollo. Se presenta las tarjetas CRC para apreciar el

desarrollo orientado a objetos. También aparecen las tareas de ingeniería para llevar a cabo el desarrollo de las historias de usuario.

Capítulo 4: “PRUEBAS”: Este capítulo está dedicado a las pruebas que se les realizan al software las pruebas de aceptación del cliente. Estas pruebas fueron llevadas a cabo antes de cada entrega que se realizó durante todo el desarrollo del proyecto.

Capítulo 5: “ESTUDIO DE FACTIBILIDAD”: En este capítulo se realiza un estudio de factibilidad del proyecto, se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea la conveniencia de la ejecución del proyecto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza un proceso investigativo para la elaboración de los Cortes Evaluativos, un estudio del proceso de gestión y la estructura que debe tener con sus funciones asociadas; estableciendo la estrategia a llevar a cabo y los beneficios esperados. Se describen principales conceptos asociados al problema que son necesarios para un mejor entendimiento y darle solución al mismo. Se hace una caracterización de cada tipo de herramientas y del lenguaje de programación, así como las metodologías a utilizar para el desarrollo del software.

1.2 ¿Qué son los Cortes Evaluativos?

Los Cortes Evaluativos son evaluaciones realizadas por el profesor, frecuentes, parciales o finales, en el período que se informa, el cual da una evaluación al estudiante de bien, regular o mal.

Los Cortes Evaluativos tienen dentro de sus propósitos ofrecer al profesor la información necesaria que le permita evaluar la calidad del proceso de enseñanza aprendizaje de sus alumnos y de esta forma trazar la estrategia metodológica que le permita solucionar las dificultades detectadas así como el proyecto educativo. Para el éxito de esta actividad, debemos lograr que los estudiantes se preocupen por su futuro y por alcanzar buenos resultados académicos, por la otra, construir nuevos conocimientos en base al que ya poseen y profundizar en el aprendizaje por parte de los profesores, tomando medidas necesarias para el desarrollo profesional. (Vicerrectoría Docente 2009)

1.2 Funcionamiento de los Cortes Evaluativos.

La periodicidad de los informes del Corte Evaluativo se fijará en las indicaciones para la planificación del curso, que se emiten por la Vicerrectoría Docente.

Para desarrollar este propósito con éxito los Cortes Evaluativos tendrán en cuenta los resultados de las evaluaciones, frecuentes, parciales o finales realizadas en el período que se informa.

Los coordinadores de años serán los máximos responsables ante el jefe del departamento de compilar toda la información del colectivo de año, para que en los departamentos docentes terminales, se hagan las reuniones para los análisis pertinentes y lleguen al decanato unificadamente por carreras el cual emite un resumen y lo manda para la Vicerrectoría Docente.

Los informes se realizarán con el modelo que se adjunta en este sistema con vistas a unificar la información del proceso. (Vicerrectoría Docente 2009)

1.2.1 Estructura de los Cortes Evaluativos.

- Profesores.
- Coordinadores de año.
- Responsable en el departamento.
- Responsable en la facultad.
- Responsable en la Vicerrectoría Docente.

Funciones:

- Profesores:

Cada profesor que imparta una asignatura debe hacer un Corte Evaluativo cada seis semanas, insertar la cantidad de horas de ausencia que tiene cada alumno, un promedio de las evaluaciones frecuentes y las evaluaciones parciales.

- Coordinadores de año:

Serán los máximos responsables ante el jefe del departamento de compilar toda la información del colectivo de año. Planificar el trabajo que se desarrollará con los estudiantes, chequear la asistencia de estos y coordinar con los profesores que imparten las asignaturas que darán salida a los resultados por asignatura de los estudiantes.

- Responsable en el departamento:

Será el responsable de hacer un resumen evaluativo, donde se muestre por año cuántos estudiantes están evaluados de bien, regular o mal. Tendrán que

insertar la valoración docente, la asignatura en la que se tiene mayor dificultad y medidas tomadas, situación de los estudiantes extranjeros, bajas y causas, estrategia de permanencia (acciones realizadas), principales afectaciones al proceso docente y medidas tomadas.

➤ Responsable en la facultad:

Será el máximo responsable de realizar un resumen evaluativo para todas las carreras, donde se muestre por carrera y año cuántos estudiantes están bien, regular y mal. Insertar la valoración docente, situación de los estudiantes extranjeros, bajas y causas, marcha del proyecto educativo y de los proyectos sociales, estrategia de permanencia (acciones realizadas) y principales afectaciones al proceso docente y medidas tomadas, asignaturas con mayores dificultades y medidas tomadas, comportamiento del autoservicio (Agricultura), cumplimiento de los objetivos por año, cumplimiento de las estrategias curriculares, situación de los alumnos ayudantes y atención que se les está brindando y situación de los alumnos con arrastres.

➤ Responsable en la Vicerrectoría Docente:

Debe informar cuáles son los años con más altos porcentos de estudiantes evaluados de Mal, una tabla donde aparezca en porciento las evaluaciones de bien, regular y mal, de todos los años, mostrar una gráfica de las evaluaciones de bien y mal, otra con las horas de estudio independiente por carreras, las asignaturas con mayores dificultades por carreras, cantidad de estudiantes que están en proceso de baja, situación de estudiantes con arrastres, mostrar en tablas por facultad y carreras cantidad de evaluados con bien, regular y mal y una tabla con los resultados generales. (Vicerrectoría Docente 2009)

1.3 Estrategia de los Cortes Evaluativos:

Esta estrategia está centrada en la preparación del estudiante, donde se desarrollen rasgos interiores como es el interés y la calidad en el estudio.

La **misión** de esta actividad es lograr que el estudiante encuentre una relación entre lo académico, lo investigativo y profesional, que contribuya a establecer un vínculo estrecho entre el estudiante y los resultados a obtener.

La **visión** consolida un buen desarrollo del proceso enseñanza-aprendizaje que permita alcanzar una formación profesional con la calidad que así lo requiere.

Los **objetivos** de los Cortes Evaluativos alcanzados:

- ✓ Dotar a los estudiantes de los conocimientos necesarios. Alcanzar habilidades para localizar bibliografía e información.
- ✓ Lograr una mejor preparación profesional de los estudiantes.
- ✓ Tener el control por parte de los profesores del rendimiento y desempeño de los estudiantes. (Vicerrectoría Docente 2009)

1.4 Beneficios y resultados esperados:

- Permite la formación integral del futuro profesional.
- Suministra a los docentes una herramienta para que los estudiantes tengan mejor interés a la hora del estudio y del aprendizaje.
- Permite que los estudiantes piensen y actúen de acuerdo a sus aspiraciones.
- Estimula el crecimiento emocional, intelectual y personal en el estudiante.

1.5 Antecedentes

Durante el período investigativo del presente trabajo, no se encontró referencia alguna a un sistema que pueda gestionar el proceso en cuestión, ya sea nacional o internacionalmente. Se conoce que en la Universidad de la Habana se trabaja en la implementación de algo parecido, pero hasta el momento no se tienen resultados.

1.6 Propuesta de Solución

Implementar un sistema Web utilizando la potencialidad de la red que permita mayor eficiencia, rapidez y seguridad de los datos que se gestionan en el proceso de Cortes Evaluativos de los estudiantes en el ISMMM.

1.7 Tendencias y tecnologías actuales:

Las tecnologías informáticas están en constante desarrollo las cuales están determinadas por una combinación de influencias tecnológicas y de nuevos requerimientos por parte de los usuarios finales. Estas influencias actúan sobre todos los componentes de las tecnologías existentes y establecen el comportamiento futuro de cada una de ellas.

1.7.1 Lenguaje de programación:

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

LENGUAJES DEL LADO DEL SERVIDOR:

PHP

Es un lenguaje de programación interpretada, diseñada originalmente para la creación de páginas Web dinámicas, usado principalmente en interpretación del lado del servidor, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando distintas bibliotecas. El Pre-procesador de hipertextos se llamó PHP, siendo publicada bajo licencia de software libre. Sus principales características son: (Marcelo 2006)

- Es un lenguaje multiplataforma.
- Soporte para una gran cantidad de bases de datos.
- Integración con varias bibliotecas externas, permite generar documentos en PDF.
- Ofrece una solución simple y universal para las paginaciones dinámicas Web de fácil programación.
- Producto de código abierto.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.

- Tiene manejo de excepciones. (Marcelo 2006)

ASP (Active Server Pages)

Es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor ASP, se escribe en la misma página Web utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Los tipos de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT, aunque también se puede utilizar en un PC con Windows 98, si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux podemos utilizar las ASP si instalamos un componente denominado Chilisoft, aunque lo ideal es trabajar sobre el servidor Web para el que está pensado. (Marcelo 2006)

Con las ASP podemos realizar muchos tipos de aplicaciones. Nos permite tener acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. También existe la posibilidad de comprar componentes ActiveX fabricados por distintas empresas de desarrollo de software que sirven para realizar múltiples usos, como el envío de correo y generar gráficas dinámicamente.

JSP

Es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es pues, una tecnología orientada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. (Marcelo 2006)

LENGUAJES DEL LADO DEL CLIENTE:

JavaScript

Es un lenguaje de programación interpretado que permite a los desarrolladores crear acciones en sus páginas Web. Es utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos pero mucho más complejos. (HyperText Markup Language). (Marcelo 2006)

HTML (HyperText Markup Language)

El Lenguaje de Marcas de Hipertexto es el lenguaje de marcado predominante para la construcción de páginas Web, usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. PHTML es una extensión para un tipo de páginas Web que llevan código PHP y HTML para ser generadas. Cuando una página está escrita en PHP podemos encontrarla con varios tipos de extensiones como por ejemplo .php, .php4, .php5. (Marcelo 2006)

1.7.2 Justificación del lenguaje de programación a utilizar.

Luego de realizar un estudio de los diferentes lenguajes de programación planteados anteriormente se decide optar **PHP** del lado del servidor y **HTML** del lado del cliente.

¿Por qué utilizar PHP?

Es software libre, lo que implica menor costo y más barato que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. Es muy rápido. Su sintaxis está inspirada en C. Su librería estándar es realmente amplia, lo que permite reducir los llamados 'costes ocultos', uno de los principales defectos de ASP. (Marcelo 2006)

PHP se puede considerar multiplataforma. Funciona en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. El código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra. El acceso a las bases de datos de PHP es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor. PHP es suficientemente versátil y potente como para hacer tantas aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario relleno por el usuario. (Marcelo 2006)

1.7.3 Framework de desarrollo.

Es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones.

Macromedia Dreamweaver 8.

Dreamweaver es un editor (WYSIWYG que es el acrónimo de What You See Is What You Get; en inglés) lo que ves es lo que obtienes. Será usado para el diseño Web de la aplicación por las facilidades que brinda. Esta herramienta brinda soporte para aplicaciones PHP y además facilita el uso de las CSS acrónimo (Cascade Style Sheet; en inglés) hoja de estilo en cascada. Con esa herramienta podremos desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos, como es nuestro caso; además de contar con un soporte para aplicaciones PHP y utilización de bases PostgreSQL. (Marcelo 2006)

1.7.4 Sistemas Gestores de Base de Datos (SGBD)

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes y acrónimos respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System. Actualmente existen muchos sistemas gestores de bases de datos, entre ellos, están: MySQL, PostgreSQL y Microsoft SQL Server. (Osmany 2010)

PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, dentro de sus características destacan las siguientes: (Walberto 2009)

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo cambios. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases de datos, eliminando la necesidad del uso de bloqueos explícitos. (Walberto 2009)

Amplia variedad de tipos nativos

PostgreSQL suministra nativamente soporte para:

- ✓ Números de precisión arbitraria.
- ✓ Texto de largo ilimitado.
- ✓ Figuras geométricas (con una variedad de funciones asociadas)
- ✓ Direcciones IP (IPv4 e IPv6).
- ✓ Arrays. (Walberto 2009)

Como características adicionales tiene:

Claves ajenas también denominadas Llaves ajenas o **Claves Foráneas**

Disparadores: Un disparador se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

Todos los disparadores se definen por seis características:

- El nombre del disparador.
- El momento en que el disparador debe arrancar.
- La tabla donde el disparador se activará.

- La frecuencia de la ejecución.
- La función que podría ser llamada.

PostgreSQL posee integración con un gran número de lenguajes tales como PHP, Java, C, C++. (Walberto 2009)

MySQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Es un software de código abierto, esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos. (Osmany 2010)

Principales Características.

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexibles y seguros.
- Utilidad (Isamchk) para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintas lenguas.

- Todos los comandos tienen -help o - Para las ayudas. (Osmany 2010)

1.7.5 Justificación del SGBD a utilizar.

Luego de realizar un estudio de las diferentes características de los gestores de base de datos expuesto anteriormente se decide optar por **PostgreSQL** por las siguientes razones:

¿Características del PostgreSQL?

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) considerado como el sistema de bases de datos de código abierto más avanzado del mundo. PostgreSQL ofrece muchas ventajas para su compañía o negocio respecto a otros sistemas de bases de datos:

1- Instalación ilimitada.

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a ésta la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto trae consigo ventajas adicionales como:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento. (Walberto 2009)

2- Ahorros considerables en costos de operación

Este software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Además de esto, los programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales. (Walberto 2009)

3- Estabilidad y confiabilidad legendarias.

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. (Walberto 2009)

4- Extensible.

El código fuente está disponible para todos sin costo. Si un equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días. (Walberto 2009)

5- Multiplataforma.

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. (Walberto 2009)

6- Herramientas gráficas de diseño y administración de bases de datos.

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseños de bases de datos (Tora, Data Architect). Además PostgreSQL ofrece una serie de características técnicas, al igual que otros gestores, que permiten un mejor trabajo con las bases de datos; entre estas podemos encontrar:

- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica.
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby.
- Reglas, Vistas, Triggers, Unicote, Secuencias, Herencia, Outer Joins.
- Sub-selects, Una API abierta, Procedimientos almacenados.
- Soporte nativo SSL, Lenguajes procedurales. (Walberto 2009)

1.7.6 Entorno de trabajo.

Servidor Apache.

Apache, es un servidor de protocolo para la transferencia de hipertextos (Hypertext Transfer Protocol, HTTP por sus siglas en inglés) de software libre para plataformas Unix, Windows, y Macintosh, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Las características que lo definen son las siguientes:

- Multiplataforma.
- Es un servidor de Web conforme al protocolo HTTP/1.1.
- Basado en hebras en la versión 2.0.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta. (Osmany 2010)

1.8 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener.

Las técnicas indican cómo debe ser realizada una actividad determinada e identificada en la metodología. Combinan el empleo de modelos o representaciones gráficas junto con el empleo de procedimientos detallados. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software.

1.8.1 Metodologías ágiles.

Las Metodologías Ágiles o “Ligeras” constituyen un nuevo enfoque en el desarrollo de software, las cuales pueden explicarse a través de los siguientes 4 principios fundamentales:

1. Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Dado que el proceso de desarrollo es creativo, no es posible pensar que las personas trabajen respondiendo a órdenes o procesos rígidos.
2. Desarrollar software que funcione es más importante que conseguir una buena documentación. Puesto que si el software no funciona la documentación no vale de nada. A nivel interno puede haber documentación, pero solo la necesaria y a nivel externo lo que el cliente requiera.
3. .La colaboración con el cliente más que la negociación de un contrato. Supone que la satisfacción del cliente con el producto será mayor, mientras exista una conversación y realimentación continua entre éste y la empresa.
4. .Responder a los cambios más que seguir estrictamente un plan. Puesto que si un proyecto de software no es capaz de adaptarse a los cambios fracasará, especialmente en productos de gran envergadura. La estrategia de planificación se basa en: planes detallados para las próximas semanas, planes aproximados para los próximos meses y muy generales para plazos mayores.

A nivel mundial existen varias metodologías para el desarrollo de software que pueden ser catalogadas de ágiles. Cada una con características específicas, por lo que es importante realizar un análisis, antes de aplicar cualquiera de ellas. Dentro de las más conocidas podemos citar: (Metodologías ágiles 2010)

-SXP.

-XP (Extreme Programming – Kent Beck).

-SCRUM (Jeff Sutherland, Ken Schwaber).

De todas las metodologías ágiles citadas, se analizarán, XP y Scrum, las cuales tienen un gran peso en el mundo digital. A continuación se caracterizarán cada una de ellas, teniendo en cuenta los siguientes parámetros:

-Características principales.

-Roles.

-Ciclo de desarrollo o proceso.

-Prácticas. (Metodologías ágiles 2010)

1.8.2 XP (Extreme Programming)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Es una metodología de desarrollo ágil, basada en una serie de valores y de prácticas que persiguen el objetivo de aumentar la productividad a la hora de desarrollar programas. Tiene como filosofía fundamental la satisfacción de las necesidades del cliente, por lo que se integra como una parte más del equipo de desarrollo. Además permite la simplicidad en el trabajo del equipo, la comunicación entre clientes y desarrolladores, así como la retroalimentación o reutilización del código implementado. Esta metodología está diseñada para aplicaciones que requieren un grupo de programadores pequeños y que poseen requisitos cambiantes, debido a su rápida adaptabilidad a los cambios. (Internet 2010)

1.8.2.1 Valores de la metodología XP.

Simplicidad

XP propone el principio de hacer la cosa más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana. (Internet 2010)

El programador tiene que estar en constante comunicación con el cliente para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos. Muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes. (Internet 2010)

Comunicación

El Extreme Programming se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación: la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos (o más) personas se comunican directamente pueden no solo consumir

las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas, etc. que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva. Muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes. (Metodologías ágiles 2010)

Coraje

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave. El coraje existe en el contexto de los otros tres valores.

Cada uno de ellos se apoya en:

- Se requiere valor para comunicarse con los demás cuando eso podría exponer la propia ignorancia.
- Se requiere valor para mantener el sistema simple, dejando para mañana las decisiones de mañana.
- Se requiere coraje para confiar en que la retroalimentación durante el camino es mejor que tratar de adivinar todo con anticipación.
- Y, sin un sistema simple, comunicación constante y retroalimentación, es difícil mantenerse valiente. (Metodologías ágiles 2010)

1.8.2.2 Prácticas de la metodología XP

Planificación incremental

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de feedback que permitan conocer con precisión dónde estamos. Como es lógico, la planificación es iterativa: un representante del negocio decide al comienzo de cada

iteración y qué características concretas se van a implementar. El objetivo de la XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se les denomina releases. Una release cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio. Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente. Es fundamental en toda esta planificación la presencia de un representante del cliente, que forma parte del equipo y que decide cuáles son las historias más valiosas. Estas historias son las que se desarrollarán en la iteración actual. (Metodologías ágiles 2010)

Como se puede ver, y como siempre ocurre con la Programación Extrema, el enfoque utilizado para llevar a cabo la planificación es eminentemente pragmático. Gran parte de la eficacia de este modelo de planificación deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades:

- ✓ Decidir qué se implementa en cada iteración.
 - ✓ Fijar las fechas de fin de la release, recortando unas características o añadiendo otras.
 - ✓ Priorizar el orden de implementación, en función del valor de negocio.
- (Metodologías ágiles 2010)

Diseño simple

Otra práctica fundamental de la Programación Extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". Se evita diseñar características extra porque a la hora de la verdad la experiencia indica que raramente se puede anticipar qué necesidades se convertirán en reales y cuáles no.

La XP nos pide que no vivamos bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras: lo que parece necesario cambia con frecuencia. Es obvio que, si no vamos a anticipar futuras necesidades, debemos

poder modificar el diseño si alguna de estas se materializa. La XP soporta estas modificaciones gracias a los tests automatizados. Éstos permiten hacer cambios importantes gracias a la red de protección que proporcionan. La refactorización, que hace que el código existente sea claro y sencillo, también ayuda a hacer factibles las modificaciones.

La XP define un "diseño tan simple como sea posible" como aquel que:

- ✓ Pasa todos los tests.
- ✓ No contiene código duplicado.
- ✓ Deja clara la intención de los programadores (enfatisa el qué, no el cómo) en cada línea de código.
- ✓ Contiene el menor número posible de clases y métodos. (Metodologías ágiles 2010)

Entregas pequeñas

Siguiendo la política de la XP de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Éstas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente. Una entrega no debería tardar más de tres meses. (Internet 2010)

Estándares de programación

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible, facilitando los cambios. (Internet 2010)

1.8.2.3 Fases de la metodología XP

Hay diversas prácticas inherentes a la Programación Extrema, en cada uno de los ciclos de desarrollo del proyecto.

Planificación

Historias de usuario: El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente.

Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de tres o cuatro líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados. Release planning: Tras definir las historias de usuario es necesario crear un plan de publicaciones, donde se indiquen las historias de usuario que se implementarán para cada versión de la aplicación y las fechas en las que se publicarán dichas versiones.

Iteraciones. Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. (Metodologías ágiles 2010)

Diseño

Diseños simples: La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácil de entender e implementar que a la larga costará menos tiempo y esfuerzo desarrollar.

Glosarios de términos: Usar una correcta especificación de los nombres de clases, métodos y propiedades ayudará a comprender el diseño y facilitará futuras ampliaciones y la reutilización del código.

Tarjetas CRC. El uso de las tarjetas CRC (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. (Metodologías ágiles 2010)

Codificación

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes

del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse atendiendo a estándares y patrones de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y la escalabilidad. Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. (Metodologías ágiles 2010)

Pruebas

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa. El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.

Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido. (Metodologías ágiles 2010)

1.8.3 Scrum

Scrum es una metodología que enfatiza valores y prácticas de gestión sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas. Es totalmente un proceso de administración y control que implementa técnicas de control de procesos. En fin esta metodología define todo un marco para la gestión de proyectos, con un rápido cambio de requisitos. Tiene como características fundamentales, el desarrollo de software mediante iteraciones, denominadas Sprint, con una duración de 30 días y las reuniones a lo largo del proyecto, entre ellas se

destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Scrum 2010)

Características de Scrum

El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados. Entonces, se determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint. (Scrum 2010)

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar. (Scrum 2010)

1.8.4 ¿Por qué aplicar SXP?

Si estás pensando en cambiar tu tecnología debes tener claro qué problemas tienes y saber si una metodología ágil, te va a ayudar a resolver ese problema; ya que cambiar una tecnología, porque otra es la del momento puede traer muchas dificultades. (Metodologías ágiles 2010)

Para el desarrollo de este trabajo se propone el uso de la metodología ágil SXP la cual describe la arquitectura del sistema, estrategias de integración con otros sistemas, así como la definición de herramientas y tecnologías de desarrollo a

emplear por el equipo de solución, tiene cuatro fases, planeación-definición, prueba, entrega y mantenimiento, metodología propuesta por la Universidad de las Ciencias Informáticas y que es requerida por nuestra institución.

1.8.4.1 Fases de la metodología SXP

- **Planificación-Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto. Se generan los documentos que se encuentran relacionados con la concepción inicial del sistema, concepción de sistema, usuarios del negocio, lista de reserva del producto, historias de usuarios, lista de riesgos y modelo de diseño.
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que esté listo para ser entregado y se realizan las pruebas.
- **Entrega:** puesta en marcha.
- **Mantenimiento:** donde se realiza el soporte para el cliente. (Peñalver y Garcías 2010)

De cada una de ellas se despliegan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, implantación, prueba, entrega de la documentación, soporte e investigación, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto. (Peñalver y Garcías 2010)

Entre los resultados obtenidos podemos decir que:

La metodología establece el uso de sistemas automatizados para la generación de algunos artefactos. Y además los recomienda de manera explícita. Contar con un expediente le permitió sobrevivir, llegar a usarse. Establecerse en el entorno con apoyo legal. Muy usada en proyectos que no utilizan paradigma orientado a objetos. Fundamentalmente proyectos de Software Libre (SWL) que trabajan con ficheros Bash o C.

La descripción de una historia de usuario suele ser más sencilla que la de un caso de uso.

Positiva para el uso de estudiantes que aprenden ingeniería. En la práctica los estudiantes de años avanzados en la carrera son los únicos que pueden desempeñar el rol de analistas o diseñadores de un software, lo que hace muy improductiva a la fuerza joven, que en ocasiones tiene buenas ideas y energía para desarrollarlas, e incluso muy buena preparación en uno o dos lenguajes de programación.

SXP cuenta con artefactos muy bien definidos para períodos de investigación. En el desarrollo de software libre, para no reinventar la rueda, se hace necesario investigar los proyectos que han intentado hacer lo mismo. En un banco de software disponible de unos 200 000 proyectos, los períodos de investigación suelen ser de unos 3 ó 4 meses. Posterior a ello el equipo comienza siempre a desarrollar partiendo de código ya desarrollado y que fue encontrado durante la investigación. Documentar estos 4 meses de investigación - en pocos artefactos - suele ser productivo para que otros equipos de desarrollo tomen decisiones en menos tiempo. También permite a desarrolladores en entornos académicos obtener publicaciones.

SXP ha sido excelente para equipos pequeños, siempre ha sido recomendado para menos de 20 personas.

La documentación siempre tiene un día o una semana de retraso con respecto al código.

Siempre hay más código que documentación. En el entorno UCI con metodología RUP suele haber mucho papel y poco código. Lamentablemente el proceso de revisión y auditorías no tiene un acápite para revisar el código, o sea si usted ha dicho que usa una metodología ágil lo más conveniente sería revisar las Historias de Usuario (HU) que debe haber implementado según el cronograma y hacer al menos pruebas de caja negra a la aplicación para evaluar la veracidad de la documentación.

Las pruebas en SXP han sido por regla general satisfactorias.

Estableciendo un sistema de trabajo colaborativo SXP tiene una plantilla de Arquitectura que ha pasado por múltiples manos. SXP también tiene asociado el concepto de visión de Arquitectura: Si tu equipo no sabe cómo hacer este sistema ¡píntalo! Luego programación chatarra e integra todo eso cada vez que sientas que estás en un escalón superior. (Peñalver y Garcías 2010)

1.9 Patrones Arquitectónicos

Patrón de arquitectura MCV (Modelo vista controlador) - Para el diseño de aplicaciones con sofisticadas interfaces se emplea el patrón de diseño MCV.

La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. (Basulto 2010)

Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error. (Basulto 2010)

Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad.

Elementos del patrón:

- Modelo: los datos con lo que trabaja el sistema (Base de Datos).
- Vista: muestra la información del modelo a los usuarios.
- Controlador: gestiona las entradas de los usuarios.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información en una base de datos, que puede presentarse de diversas formas: diagrama de pastel, de barras, tabular, etc.

Analizan cada componente:

El modelo es responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema).
- Llevar un registro de las vistas y controladores del sistema.
- Si se está en presencia de un modelo activo, el mismo notificará a las vistas los cambios que en los datos pueda producir un agente externo.

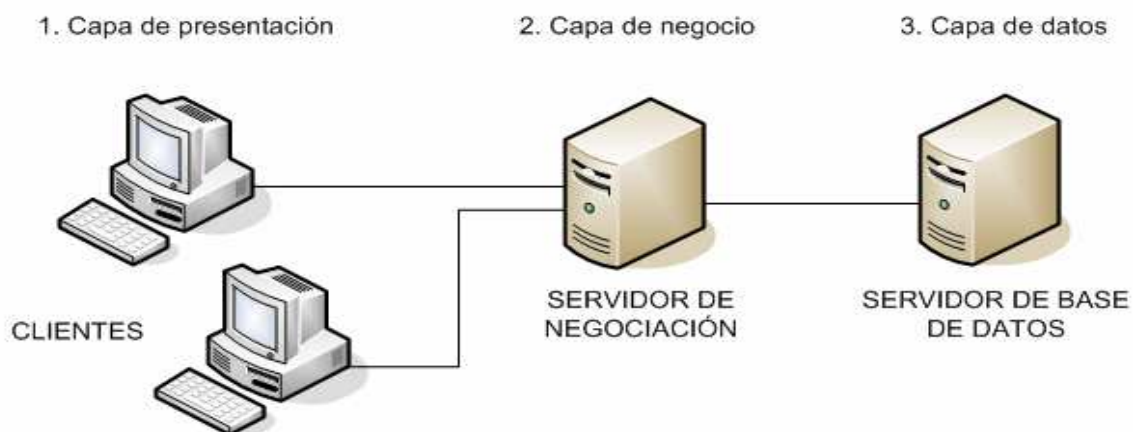
El controlador es responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo “si el evento z, entonces acción w”. estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar.

Las vistas son responsables de:

- Recibir datos del modelo y mostrarlo al usuario.
- Pueden suministrar el servicio de actualizar, para que sea invocado por el controlador o por el modelo, (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El diseño de sistemas informáticos suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). (Basulto 2010)



Capas o niveles

Capa de presentación o interface: es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso, (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la del negocio. (Basulto 2010)

Capa de lógica o de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio e incluso lógica del negocio, pues es aquí donde se establecen las reglas que deben cumplirse. Esta capa se comunica con la de presentación para recibir las solicitudes y presentar los resultados, y con la capa de

datos para solicitar al gestor de bases de datos para almacenar o recuperar datos de él. (Basulto 2010)

Capa de datos: es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben solicitudes de almacenamiento o de recuperación de información desde la lógica del negocio. (Basulto 2010)

Todas estas capas pueden residir en un único ordenador, esto no sería lo normal, lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor).

Las capas de negocio y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, pueden separarse en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos. En una arquitectura de tres niveles, los términos “capas” y “niveles” no significan lo mismo ni son similares. El término capa hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: interface/lógica del negocio/datos. (Basulto 2010)

En cambio, el término nivel, corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo: Una solución de tres capas (interface, lógica, datos), que residen en un solo ordenador (interface +lógica +datos), se dice que la arquitectura de la solución es de tres capas y un nivel.

Una solución de tres capas que residen en dos ordenadores (interface +lógica, lógica +datos), se dice que la arquitectura de la solución es de tres capas y dos niveles. Una solución de tres capas que residen en tres ordenadores, la arquitectura que la define es: una solución de tres capas y tres niveles. (Basulto 2010)

El que se utiliza en el software es **Arquitectura en 3 capas y dos niveles**: ya que el sistema la base de datos que va a utilizar estará en otro ordenador.

1.10 Conclusiones

En este capítulo se hace un estudio sobre los Cortes Evaluativos y se exponen los conceptos necesarios para el entendimiento del software, se toman decisiones importantes, luego de un estudio de las distintas herramientas para el desarrollo de software tales como la elección de los lenguajes de programación y metodología utilizada.

CAPÍTULO 2: PLANIFICACIÓN - DEFINICIÓN

2.1 Introducción

En este capítulo se presenta parte de la propuesta de solución de la investigación, haciendo uso adecuado de las metodologías de desarrollo SXP en la que se aborda la fase de planificación y definición en el cual se explicará toda la dinámica del proyecto. Esta metodología es la unión de XP y Scrum para el logro de un buen desarrollo de software.

La metodología está dividida en cuatro fases las cuales son:

- Planificación-Definición
- Desarrollo
- Entrega
- Mantenimiento

Cada una de estas fases está compuesta por una serie de actividades que son las que generan los artefactos que quedan incluidos en el expediente de proyecto.

Procedimientos

Explica la estructura del proyecto plantando una serie de planillas que son responsabilidad de los diferentes roles que deben existir a la hora del desarrollo del software.

2.2 Planificación del proyecto por roles.

La creación de software con metodologías ágiles exige de la creación de pequeños grupos de trabajo. Uno de los principales aspectos antes de comenzar el proceso de documentación es distribuir las tareas por cada uno de los roles existentes, lo que garantiza un trabajo organizado.

➤ Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona realimentación al equipo en el proceso. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso

de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

➤ **Cliente (Customer):**

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

➤ **Programador (Programmer):**

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

➤ **Analista (Analyst):**

Escribe la concepción del sistema y las historias de usuario. Crea el Modelo de historia de usuario del negocio y la LRP. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio, todo esto lo realiza junto con el cliente.

➤ **Diseñador (Designer):**

Son los encargados del diseño del sistema, así como el de los prototipos de interfases, máximos responsables de la realización del diseño de las metáforas y supervisan el proceso de construcción.

➤ **Encargado de Pruebas (Tester):**

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales, ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

2.3 Planificación y Definición del proyecto.

En la fase de planificación se incluyen aquellos documentos que están relacionados con la estimación inicial de esfuerzos, y la valoración de los riesgos. Cada una de las plantillas que han sido incluidas en esta fase se genera de una actividad en específico, y tienen su importancia en el proceso de documentación de software.

2.3.1 Plantilla Concepción del sistema

La Plantilla Concepción del sistema, es el primer documento generado en la fase de Planificación-Definición, este queda elaborado luego de realizarse la actividad de entrevista con el cliente, momento en el que se define la concepción inicial del sistema. Este documento además de reflejar la visión general del producto a implementar, también recoge los diferentes roles que intervendrán en el desarrollo del software, así como las responsabilidades que tendrán en dicho proceso. Se recoge además cuáles herramientas serán utilizadas para el desarrollo de la aplicación, el alcance que va a tener, una descripción de los involucrados en el negocio, cuáles son los motivos de la necesidad del desarrollo del software y la propuesta de solución. Esta es la plantilla principal dentro de la documentación por ser la guía para los demás documentos que se generan durante el ciclo de desarrollo de software.

Roles: Analista y Cliente.

La plantilla está en el: [Anexo 1](#)

2.3.2 Plantilla Modelo de Historias de usuario del negocio.

La plantilla del Modelo Historias de usuario del negocio, es un artefacto que se genera del Juego de la planificación, luego de estar definida la concepción del sistema, se hace mucho más fácil comprender el negocio.

Se definen las características específicas del negocio, así como la forma en que interactúa el sistema con el cliente y viceversa. El Modelo de negocio cuando se trabaja con metodologías ágiles, es diferente al ya conocido en el proceso unificado, ya que en este caso se trabaja con historias de usuarios, en vez de con casos de

uso. Pero independientemente de los cambios técnicos que puedan existir, el negocio se modela igual en cualquier metodología.

Roles: Analista

La plantilla está en el: [Anexo 2](#)

2.3.3 Plantilla Lista de Reserva del Producto (LRP)

La plantilla de Lista de Reserva del Producto, es el primer artefacto generado en la etapa de captura de requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración.

Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. Esta lista puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas.

Roles: Analista y Cliente

La plantilla está en el: [Anexo 3](#)

2.3.4 Plantilla Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. El tratamiento de las historias de usuario es muy dinámico y flexible y tienen el mismo propósito que los casos de uso, las escriben los propios clientes, tal y como ven ellos las necesidades del sistema así como las tareas que el sistema debe hacer y su

construcción depende principalmente de la habilidad que tenga el cliente para definir las.

Las historias de usuario guían la construcción de las pruebas de aceptación, elemento clave en XP (deben generarse una serie de pruebas para verificar que la historia ha sido implementada adecuadamente) y son utilizadas para estimar tiempos de desarrollo. En este sentido, sólo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. En el momento de implementar una historia de usuario, se debe detallar a través de la comunicación con el cliente. En esta plantilla los campos de puntos estimados y puntos reales son llenados luego del desarrollo de la actividad de estimación de esfuerzo, en el cual se decide el tiempo que se le dedicará a cada historia de usuario.

Las historias de usuario proporcionan ventajas, tales como:

- Están escritas en lenguaje del cliente, por lo que es muy fácil su comprensión.
- Especifican cada uno de los requisitos del sistema, sin necesidad de documentaciones extensas.
- Reflejan todas las características del sistema.
- Si se definen correctamente, guían el proceso de implementación.

Roles: Analista y Cliente

La plantilla está en el: [Anexo 4](#)

Estimación de esfuerzo

Se mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. De

esta forma se puede trazar el plan de entregas en función de estos dos parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente. Las iteraciones individuales son planificadas en detalle justo antes de que comience cada iteración.

2.3.5 Plantilla Lista de riesgos

La plantilla de Lista de riesgos, es el documento que se genera de la actividad de valoración de riesgos. En ella quedan definidos los posibles riesgos que actuarán sobre el proceso de desarrollo de software, así como la estrategia trazada para mitigarlos, además de un plan de contingencia que describe qué curso seguirán las acciones si el riesgo se materializa.

Esta plantilla posee una gran importancia, pues a pesar de que es imposible definir desde un inicio todos los riesgos que pueda atravesar un proyecto, sí se tendrán algunos en cuenta, fundamentalmente si se trata de un equipo de desarrollo con experiencia. Esta plantilla propicia algunas ventajas, tales como:

- Se definen los posibles riesgos, así como la forma de mitigarlos, lo que disminuye el efecto de los mismos, si ocurrieran.
- Se lleva un control de todos los problemas que han azotado al proyecto, así como de la manera que fueron enfrentados y el impacto que tuvieron en el proceso de desarrollo.

Roles: Gerente

La plantilla está en el: [Anexo 5](#)

2.4 Diseño

El diseño debe implementar todos los requisitos explícitos contenidos en la Lista de Reserva. Debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software. El diseño debe proporcionar una completa idea de lo que es el software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la Implementación.

En la fase de Diseño se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un Sistema, con suficientes detalles como para permitir su interpretación y realización física. Éste transforma elementos estructurales de la arquitectura del programa. Dentro del diseño es donde se fomenta la calidad del proyecto.

2.4.1 Plantilla Modelo de diseño

La plantilla del Modelo de diseño, es el documento que se genera del Diseño con las metáforas, donde se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora.

Teniendo en cuenta las características anteriores, se define en esta plantilla, un esbozo inicial del diseño del sistema, sin entrar en especificaciones ni detalles, solo lo que el diseñador necesita para hacer un primer entregable del sistema.

Esta plantilla proporciona ventajas, tales como:

- Permite confeccionar un diseño inicial y sencillo del sistema.
- Es la base para la definición de una futura arquitectura.

Roles: Diseñador

La plantilla está en el: [Anexo 6](#)

2.5 Planificación de las iteraciones

Con respecto a las Historias de Usuarios ya presentadas realizamos una planificación de iteraciones basándonos en el tiempo y procurando obtener la funcionalidad relacionada en la misma iteración.

2.5.1 Primera iteración:

- Gestionar Usuarios
- Gestionar Datos Primarios
- Autenticar Usuarios

En esta primera iteración se le da cumplimiento a las Historias de Usuario que se consideraron importantes en el estado inicial para el desarrollo del sistema.

Se trata del trabajo que se va a hacer con las cuentas de los usuarios y los datos iniciales que debe tener el sistema.

2.5.2 Segunda iteración:

- Gestionar Asistencia
- Gestionar Evaluaciones parciales de los Estudiantes
- Gestionar Evaluaciones parciales de las Áreas

En esta iteración se le dará cumplimiento a la Historia de Usuario, gestionar Asistencia, gestionar evaluaciones parciales de los estudiantes y gestionar evaluaciones parciales de las áreas, las cuales tienen como funciones que el usuario pueda insertar, eliminar o modificar estos datos.

Esta iteración unida a la anterior da resultado a la segunda versión de prueba del mismo, y será entregada al cliente a ver si cumple con los requisitos que se acordaron.

2.5.3 Tercera iteración:

- Mostrar Evaluación
- Gestionar Información

En esta iteración se les dará cumplimiento a la Historia de Usuario Gestionar Información que le permite al usuario insertar informaciones necesarias, modificarlas y mostrarla y la otra Historia de Usuario es Mostrar Evaluación que le permite al usuario dar evaluaciones de las áreas, modificarlas y mostrarla al usuario.

El resultado de esta iteración integrada a la iteración anterior dará como resultado la tercera versión del sistema, además de la tercera versión de prueba del mismo. Esta

versión será entregada al cliente para verificar si cumple con los requisitos acordados.

2.5.4 Cuarta iteración:

- Gestionar Gráficas
- Gestionar Reportes

En esta iteración se les dará cumplimiento a la Historia de Usuario Gestionar Gráficas de los resultados en la cual se grafican los resultados y se comparan y se les dará cumplimiento a la Historia de Usuario Gestionar Reportes la cual brinda reportes de los cortes evaluativos y le permite al usuario imprimir dichos reportes.

El resultado de esta iteración integrada a la iteración anterior dará como resultado la cuarta versión del sistema, además de la cuarta versión de prueba del mismo. Esta versión será entregada al cliente para verificar si cumple con los requisitos acordados.

2.5.5 Quinta iteración:

- Plan de duración de las iteraciones.

De acuerdo a las Historias de Usuarios ya Planteadas se realizará una planificación en cuatro iteraciones basándose en el tiempo y procurando agrupar la funcionalidad relacionada en la misma iteración.

Tabla_1. Tiempos de las iteraciones.

Iteración	Orden de implementación por Historias de Usuario	Duración total de la iteración en semanas
1	<ul style="list-style-type: none"> • Gestionar Usuarios • Gestionar Datos Primarios • Autenticar Usuarios 	4
2	<ul style="list-style-type: none"> • Gestionar Asistencia. • Gestionar Evaluaciones parciales de los estudiantes • Gestionar Evaluaciones 	3.5

	parciales de las áreas	
3	<ul style="list-style-type: none">• Mostrar Evaluación• Gestionar Información	4
4	<ul style="list-style-type: none">• Gestionar Gráficas• Gestionar Reportes	4

2.6 Conclusiones

En este capítulo hemos abordado la fase de planificación y definición en el cual se explicará parte del proyecto basándose en el expediente del proyecto, se planificaron las historias de usuario que se deben tener en cuenta para la construcción del sistema y las iteraciones basándose en la estimación de esfuerzo de cada historia de usuario. Se concluyó que este capítulo está listo para arribar a la siguiente etapa de desarrollo.

CAPÍTULO 3: DESARROLLO

3.1 Introducción

En este capítulo se aborda lo perteneciente a la fase de desarrollo de acuerdo a la metodología de desarrollo, se presentan las tarjetas CRC que le permiten al programador centrarse en desarrollo orientado a objetos. También aparecen las tareas de ingeniería para llevar a cabo el desarrollo de las historias de usuario.

3.2 Diseño de la Solución Propuesta

La metodología de desarrollo XP establece prácticas especializadas, que inciden directamente en la realización y elaboración del diseño de un software, sin embargo no requiere que la representación del sistema sea mediante diagramas de clases basados en UML, sino que pueden emplearse indistintamente sencillos esquemas descritos en pizarras u otras técnicas como las tarjetas CRC. No obstante el empleo de los diagramas UML pueden ser utilizados siempre y cuando contribuyan al mejoramiento de la comunicación del equipo de desarrollo, que no sean muy extensos y no requieran de mucho tiempo para su creación.

3.2.1 Tarjetas CRC

El uso de las tarjetas CRC (Class, Responsibilities and Collaboration) permiten al programador centrarse en el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas CRC representan objetos; la clase a la que pertenece el objeto se escribe en la parte de arriba de la tarjeta, en una columna a la izquierda se escriben las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. Las tarjetas CRC fueron propuestas por Ward Cunningham and Kent Beck.

Tabla_2. Tarjeta CRC #1 Principal

Nombre de la clase: Principal
Tipo de la clase: Lógica del negocio

Responsabilidades:	Colaboradores:
Gestionar usuarios	Usuarios

Para consultar el resto de las tarjetas C.R.C ir al: [Anexo 7](#)

3.3 Desarrollo del proyecto

En XP generalmente las historias de usuarios se dividen en tareas de ingenierías (TI). Estas se crean para lograr una mejor planificación de las historias; con ellas se pretende cumplir con las funcionalidades generales de cada historia. Las tareas no tienen que obligatoriamente ser entendidas por el cliente, pueden ser escritas en lenguajes técnicos, puesto que estas son usadas únicamente por los programadores. A continuación se muestran algunas de las tareas de ingeniería en cada iteración del sistema.

3.3.1 Plantilla Tarea de ingeniería

La plantilla Tarea de ingeniería, tiene gran importancia, pues permite definir cada una de las actividades asociadas a las Historias de Usuario. Así se estimará el tiempo que llevará cada historia de usuario en implementarse de acuerdo a su complejidad. Esta plantilla proporciona ventajas tales como: Permite organizar el proceso de implementación, pues las tareas se van implementando de acuerdo a su prioridad, posibilita conocer el grado de complejidad de cada historia de usuario, teniendo en cuenta la cantidad de tareas asociadas.

Roles: Programador

3.3.2 Primera iteración

En esta iteración se le dio cumplimiento a la implementación de la Historia de Usuario número 1, 2 y 3, dichas HU brindan funcionalidades como la gestión de usuarios, cargar y visualizar los datos primarios y la entrada al sistema de los usuarios correctamente.

Tabla_3.Historias de Usuario primera iteración

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar Usuarios	2	1.5
Gestionar Datos Primarios	1	1
Autenticar Usuarios	1	0.7

Tareas de ingeniería vinculadas a esta iteración
Tabla_4.Tareas de ingeniería # 1 Crear cuentas de Usuarios

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Crear cuentas de usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 10 de abril 2011	Fecha fin: 12 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir crear cuentas de usuarios con el nivel adecuado.	

Para consultar el resto de las Tareas de Ingeniería ir al: [Tabla 33 – Tabla 38](#)

3.3.3 Segunda iteración

En esta iteración del sistema se le dará cumplimiento a la Historia de Usuario número 4, 5 y 6, dichas HU brindan funcionalidades como, permitir la gestión de asistencia, las evaluaciones de las áreas y las evaluaciones parciales de los estudiantes.

Tabla_5.Historias de Usuario segunda iteración

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar Asistencia	2	1.5
Gestionar Evaluaciones parciales de los estudiantes	2	1.5
Gestionar Evaluaciones parciales de las áreas	1.5	1.5

Tareas de ingeniería vinculadas a esta iteración

Para consultar las Tareas de Ingeniería de esta iteración ir al: [Tabla 39 – Tabla 47](#)

3.3.4 Tercera iteración

En esta iteración del sistema se le dará cumplimiento a la Historia de Usuario número 7 y 8, dichas HU brindan funcionalidades como, mostrar evaluaciones generales de los estudiantes y la gestión de las informaciones.

Tabla_6.Historias de Usuario tercera iteración

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Mostrar Evaluación	2	2
Gestionar Información	2	1.5

Tareas de ingeniería vinculadas a esta iteración

Para consultar las Tareas de Ingeniería de esta iteración ir al: [Tabla 48 – Tabla 53](#)

3.3.5 Cuarta iteración

En esta iteración del sistema se le dará cumplimiento a la Historia de Usuario número 9 y 10, dichas HU brindan funcionalidades como, permitir la gestión de las gráficas de los resultados, brindando una comparación y los reportes permitiéndolo pasar de formato digital a impreso.

Tabla_7.Historias de Usuario cuarta iteración

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Gestionar Gráficas de los resultados de las evaluaciones	2	2
Gestionar Reportes	2	1

Tareas de ingeniería vinculadas a esta iteración

Para consultar las Tareas de Ingeniería de esta iteración ir al: [Tabla 54 – Tabla 56](#)

3.3.6 Plantilla Cronograma de producción

El cronograma de producción no es más que la plantilla que recoge las actividades realizadas a desarrollo durante la iteración. Se realiza un cronograma para cada iteración planificada durante el proceso.

3.4 Conclusiones

Con la realización de este capítulo hemos abordado la fase de desarrollo planteada por la metodología SXP en el cual se hace referencia a las tareas de ingeniería correspondientes a cada historia de usuario, se elaboraron las tarjetas CRC para apreciar el desarrollo orientado a objetos y se realizó el diseño de los datos.

CAPITULO 4: PRUEBAS

4.1 Introducción

En este capítulo se abordan las pruebas de aceptación que fueron realizadas por el cliente para demostrar que el sistema funcione adecuadamente. Estas pruebas fueron llevadas a cabo antes de cada entrega que se realizó durante todo el desarrollo del proyecto.

4.2 Pruebas de software

Las pruebas de software son elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

Una vez generado el código fuente, el software debe ser probado para descubrir y corregir el máximo de errores posibles antes de su entrega al cliente. El objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores. Aquí es donde se aplican las técnicas de pruebas del software.

En la metodología XP es importante el desarrollo de las pruebas. Cada vez que se quiere implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. XP posee dos tipos de pruebas; las unitarias o TDD (desarrollo dirigido por pruebas, del inglés Test Driven Development), desarrolladas por los programadores verificando su código de forma automática y las pruebas de aceptación que son las que se utilizarán en este capítulo, dichas pruebas serán evaluadas luego de culminar una iteración verificando así si se cumplió la funcionalidad requerida.

4.2.1 Desarrollo Dirigido por Pruebas

El desarrollo dirigido por pruebas (TDD), se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso y obtener un resultado, aunque aún no con lógica para el negocio, pero sí funcional. Algunas personas confunden este término con las nombradas "pruebas de caja blanca", las cuales son pruebas que se realizan a los métodos u operaciones para medir la funcionalidad del mismo

desde la perspectiva de la validez para el cliente. Sin embargo el TDD se aplica antes de comenzar a implementar cada paso de la tarea en desarrollo asumiendo que la prueba es insatisfactoria desde un inicio. Solo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida. Luego se realiza un proceso conocido informalmente como "refactorización" de código, el cual consiste en limpiarlo, organizarlo y adaptarlo a los patrones.

4.3 Objetivos de las pruebas

El objetivo de las pruebas, es encontrar el mayor número posible de errores con una cantidad razonable de esfuerzo, aplicado sobre una cantidad de tiempo realista. Se debe ejecutar antes de que llegue al cliente, con el propósito de descubrir todos los errores, de manera que el cliente obtenga el producto de calidad.

Otro de sus objetivos son que, un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces, una prueba tiene éxito si descubre un error no detectado hasta entonces, reducir costos de mantenimiento, obtener información concreta acerca de fallas, que pueda usarse como apoyo en la mejora de procesos, y en la de los desarrolladores.

En fin nuestro objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo. (Pruebas 2008)

4.4 Plantilla Caso de prueba de aceptación

La plantilla de Caso de prueba de aceptación, se genera de la etapa de pruebas. El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Roles: Cliente y Tester.

Tabla_8.Prueba de aceptación #1 para la HU "Gestionar Usuarios"

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_HU1_P1	Nombre Historia de Usuario: Gestionar Usuarios
Nombre de la persona que realiza la prueba: Joandris Rivera legrá	
Descripción de la Prueba: Se realiza una prueba para probar si se logra una buena gestión de usuarios (eliminar, modificar y crear cuentas de usuarios).	
Condiciones de Ejecución: Se debe probar que el sistema este seguro y a la hora de insertar un usuario, si ya existe un usuario con el nivel que se quiere ingresar, de un mensaje de error.	
Entrada / Pasos de ejecución: Se solicita la gestión de usuarios y después se verifica si funciona correctamente.	
Resultado Esperado: El sistema no presenta errores.	
Evaluación de la Prueba: Satisfactoria	

Para consultar las otras Prueba ir al: [Anexo 12](#)

4.5 Conclusiones

Con la elaboración de este capítulo hemos abordado la fase de pruebas planteada por la metodología SXP, con la realización de las pruebas de aceptación en las que el cliente se asegura de que las funciones implementadas cumplan su objetivo satisfactoriamente, probando individualmente cada HU y asignándole la evaluación correspondiente. Todas las pruebas que se realizaron fueron efectivas y el cliente estuvo satisfecho.

CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD

5.1 Introducción

Para estudiar la factibilidad de este proyecto se utilizará la **Metodología Costo Efectividad (Beneficio)**, la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados. (GARCÍA)
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). (GARCÍA)

5.2 Efectos económicos

- ✓ Efectos directos.
- ✓ Efectos indirectos
- ✓ Efectos externos
- ✓ Intangibles

5.2.1 Efectos directos

- Positivos
 - ✓ Se gestiona la información necesaria a la que los usuarios con acceso al sistema podrán acceder.
 - ✓ Mayor integración usuario- artefactos, ya que por medio de este el usuario siente necesidades de interactuar con el sistema, debido a que este gestiona, facilita y le brinda lo referente a la realización de los Cortes Evaluativos.

- ✓ Se cuenta con una herramienta capaz de agilizar el proceso de los Cortes Evaluativos.
 - ✓ Permite al usuario obtener resumen de las evaluaciones de los estudiantes, carrera, facultad y general.
 - ✓ Permitirá llevar este resumen a formato que se pueda imprimir.
- **Negativos**
- ✓ Para usar la aplicación es imprescindible el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.

5.2.2 Efectos indirectos

- Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

5.2.3 Intangibles

- En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible. (GARCÍA)

A fin de medir con precisión los efectos, deberán considerarse dos situaciones:

✓ **Situación sin Proyectos**

La información de los Cortes Evaluativos es recogida mediante documentos Excel, Word y de forma manual lo que provoca confusión y pérdida de información al manejar todo el contenido.

✓ **Situación con Proyectos**

Los usuarios podrán ver el contenido solo con entrar al sistema. Para la gestión de los datos se debe autenticar, si está autorizado por un administrador u otro usuario con privilegio siguiendo los pasos correspondientes:

- Luego de autenticado el sistema permitirá insertar, evaluar, eliminar y mostrar la información deseada de acuerdo al nivel de acceso que posea.
- Trabajar con los usuarios, ya sea insertar modificar o eliminar de acuerdo al nivel de acceso que posea.

5.3 Beneficios y Costos Intangibles en el proyecto

COSTOS:

- Resistencia al cambio.

BENEFICIOS:

- ✓ Mayor comodidad para los usuarios.
- ✓ Mejora en la calidad de la información.
- ✓ Menor tiempo empleado en la gestión de los datos.
- ✓ Facilidad a la hora de buscar la información.

5.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar Una Ficha De Costo de un Producto Informático [Dra. Ana María Gracia Pérez, UCLV]. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional. (GARCÍA)

Costos en Moneda Libremente Convertible:

- ✓ **Costos Directos.**
 1. Compra de equipos de cómputo: No procede.
 2. Alquiler de equipos de cómputo: No procede.
 3. Compra de licencia de Software: No procede.
 4. Depreciación de equipos: \$ 25.00
 5. Materiales directos: No procede.

Total: \$ 25.00

✓ **Costos Indirectos.**

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00

✓ **Gastos de distribución y venta.**

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00

Costos en Moneda Nacional:

✓ **Costos Directos.**

1. Salario del personal que laborará en el proyecto: \$100.00
2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede.
4. Gasto por consumo de energía eléctrica: \$ 152.83
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

✓ **Costos Indirectos.**

1. Know How: No procede

Total: \$ 252.83

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tómesese como costo el tiempo empleado para desarrollar el proceso de gestión de las Prácticas Laborales y la variable sería la cantidad de pasos a realizar, para llevar a cabo el proceso, para lo cual tenemos 2 valores. (GARCÍA)

Valores de la variable (Solución manual)

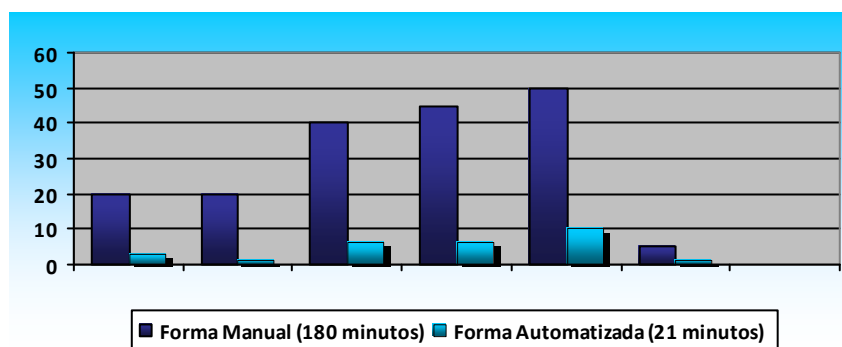
1. Realizar la evaluación de los estudiantes por parte de los profesores durante el transcurso de los Cortes Evaluativos. 1 variables (20 min.).
2. Realizar la evaluación de los estudiantes por parte de los profesores guías durante el transcurso de los Cortes Evaluativos. 1 variables (20 min.).
3. Realizar la evaluación del resumen de la carrera durante el transcurso de los Cortes Evaluativos. 1 variables (40 min.).
4. Realizar la evaluación del resumen de la facultad durante el transcurso de los Cortes Evaluativos. 1 variables (45 min.).
5. Realizar la evaluación del resumen de la vicerrectoría docente durante el transcurso de los Cortes Evaluativos. 1 variables (50 min.).
6. Imprimir resumen con las evaluaciones realizadas. 3 variables (5 min.).

Valores de la variable (Solución con el programa)

1. Cargar formulario con las evaluaciones de los estudiantes por parte de los profesores durante el transcurso de los Cortes Evaluativos, 1 variables. (3 min).
2. Cargar formulario con las evaluaciones generales de los estudiantes por parte de los profesores guías durante el transcurso de los Cortes Evaluativos, 1 variables. (1min).

3. Cargar formulario con la evaluación del resumen de la carrera, 1 variables. (6 min).
4. Cargar formulario con la evaluación del resumen de la facultad, 1 variables. (6 min).
5. Cargar formulario con la evaluación del resumen de la vicerrectoría docente, 1 variables. (10 min).
6. Exportar resumen con las evaluaciones realizadas. 3 variables (1 min.).

Gráfica de la solución manual y solución automatizada



5.5 Conclusiones

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$ 25.00 CUC y \$ 252.83 MN demostrándose la factibilidad del proyecto.

CONCLUSIONES GENERALES

- Se dió cumplimiento a las tareas concebidas para el desarrollo de la aplicación dando así desempeño al objetivo general del trabajo.
- Se hizo un análisis detallado de las diferentes metodologías ágiles para la elaboración del sistema, teniendo como resultado el uso de las metodologías SXP en las que se les dio cumplimiento a las Historias de Usuarios del sistema,
- Se realizaron las tarjetas CRC, las tareas de ingenierías para cada Historia de Usuario y las pruebas de aceptación realizadas por el cliente.
- Luego del análisis de factibilidad podemos concluir que el trabajo responde como propuesta de solución al problema planteado. Además el sistema se encuentra disponible para ser usado.

RECOMENDACIONES

- Utilizar el sistema de una forma óptima en el Instituto Superior Minero Metalúrgico de Moa.
- Agregar nuevas funcionalidades acorde a nuevos requisitos que pudieran existir durante el funcionamiento del sistema o para aumentar su rendimiento.

REFERENCIAS BIBLIOGRÁFICAS

BASULTO JORGE, Mario. *Sistema de Gestión integral de la empresa Empleadora del NIQUEL "EMPLEN". MODULO GESTION DE CONTRATOS DE COMPRAS.* Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", 2010.

DÍAZ COBAS, Walberto, *Implementación del módulo Caja del Sistema Integral de Gestión CEDRUX*, Trabajo de Diploma, UCI, 2009.

GARCÍA, S.; MENESES, A.; PEÑALVER, G. *SXP, Metodología Ágil para el desarrollo de software.* Ciudad de La Habana: Universidad de las Ciencias Informáticas, 2010.

[GARCÍA PÉREZ, Ana María "Procedimientos para la elaboración de la ficha de costo de un producto informático. Facultad MFC UCLV. Villa Clara.

HERNÁN RUIZ, Marcelo. *Programación Web Avanzada.* Trabajo de Diploma. La Habana, Editorial Félix Varela, 2006

Internet, 2010. [Consultado: 2011-01-15].
http://www.metodologiaxpvsmetodologiarup.blogspot.com/2008/04/caracteristicas-de-la-metodologia-xp_25.html.

Metodologías ágiles, 2010. [Consultado: 2011-03-25].
<http://es.wikipedia.org/w/metodologias-agiles>.

MySQL, 2007. [Consultado: 2011-03-25]. Disponible en:
<http://es.wikipedia.org/wiki/MySQL>.

Pruebas, 2008. [Consultado: 2011-03-10]. Disponible en:

http://lsi.ugr.es/~arroyo/inndoc/do+c/pruebas/pruebas_d.php .

SANTANA DÍAZ, Osmany. *Sistema de Gestión y Control de las Prácticas Laborales*, Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", 2010

Scrum, 2010. [Consultado: 2011-06-25].
<http://es.wikipedia.org/w/index.php?title=Scrum>

Vicerrectoría Docente, *Procedimiento para el informe del Corte Evaluativo a la Vicerrectoría Docente*, Moa 2009.

GLOSARIO DE TÉRMINOS

CCS: hojas de estilo en cascada (Cascading Style Sheets,) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML.

UCI: Universidad de las Ciencias Informáticas

Estándares: Es una especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

Framework: Conjunto de APIs y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera general

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web.

XHTML: Reformulación del lenguaje HTML utilizado para crear páginas web.

Iteraciones: En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio. Múltiples iteraciones contribuyen a crear un producto completamente integrado.

Interfaz de Usuario: Es la parte de una aplicación que se encarga de interactuar con el usuario.

Metodologías de Desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Release: Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Scrum: Define un marco para la gestión de proyectos. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones y la segunda característica importante son las reuniones a lo largo proyecto.

Testing: Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, se usa para detallar los artefactos en el sistema, para documentar y construir.

XP(Programación Extrema): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XML: Sigla en inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas, una manera de definir lenguajes para diferentes necesidades.

ANEXOS

Anexo 1: Plantilla Concepción del Sistema

Tabla_9.Plantilla Concepción del Sistema

1. Clasificación del proyecto.

Desarrollo Web

2. Tipo de proyecto.

Nacional

3. Resumen:

En este documento se reflejan aspectos acerca de los principales conceptos que debemos tener en cuenta para el inicio de proyecto Sistema de Gestión de los Corte Evaluativo v1.0, también se tienen las principales aspecto de porqué es necesario que se lleve a cabo el desarrollo del proyecto y cuáles son las dificultades que quedarán atrás, una vez que se ponga en funcionamiento, además de los roles que intervendrán en el desarrollo del trabajo por parte de los clientes y desarrolladores. Palabras claves: Corte Evaluativo v 1.0, producto, proyecto, clientes, desarrolladores.

4. Surgimiento

El proyecto surge a partir de que los Cortes Evaluativos se realizan de forma digital pero no con la organización y seguridad que debe poseer lo que provoca serias dificultades durante el proceso de gestión de las informaciones, demora en la entrega de los resultados y falta de calidad en el desarrollo.

5. ¿Qué es?

Un Trabajo de corte evaluativo, como se le conoce es una afirmación cuya veracidad ha sido argumentada, demostrada, o justificada de alguna manera. Estos se realizan de manera periódica lo cual se fijará para la planificación del curso que emite la vicerrectora docente.

Tendrán en cuenta los resultados de las evaluaciones frecuentes, parciales o finales realizadas en el período que se informa.

Para llevar a cabo un corte evaluativo se tienen en cuenta algunos criterios como:

- Ver el porcentaje de asistencia del alumno.
- Las evaluaciones que se realizan.
- Los resultados obtenidos en los trabajos de controles.

Se debe de informar a los estudiantes la cantidad de horas clases que tiene la asignatura, y explicar cuáles son de conferencia, cuáles de clase práctica y cuáles de seminarios. El alumno debe saber que no se debe de pasar del 20 % de asistencia.

Los estudiantes hasta el momento del corte evaluativo se evaluarán cualitativamente en Bien, Regular o Mal

de acuerdo a su desempeño.

Los coordinadores de año serán los máximos responsables ante el jefe de departamento de compilar toda la información del colectivo de año, para que en los departamentos docentes terminales, se hagan las reuniones para los análisis pertinentes y lleguen al decanato unificadamente. Los informes se realizarán con un modelo con vista a unificar la información de la facultad.

6. Metodología a utilizar

SXP, metodología propuesta por el proyecto Ágiles Unicornios de la facultad 10 (UCI), aprobada por la dirección de calidad de software y con un expediente de proyecto que cumple con CMMI.

7. Roles

Rol	Nombre	Localización
Gerente	Joandris Rivera legrá	Edificio 4 cuarto 424
Cliente	ISMM	Colorada nueva
Analista	Joandris Rivera legrá	Edificio 4 cuarto 424
Diseñador	Joandris Rivera legrá	Edificio 4 cuarto 424
Encargado de Pruebas (Tester)	Joandris Rivera legrá	Edificio 4 cuarto 424
Arquitecto de Información	Joandris Rivera legrá	Edificio 4 cuarto 424
Desarrollador	Joandris Rivera legrá	Edificio 4 cuarto 424
Consultor	Dabiel González Ramos	Moa

8. Misión

El portal web que se desea crear tiene como misión, construir la plataforma de apoyo para organizar los cortes evaluativos en el Instituto Superior Minero Metalúrgico de Moa. El Mismo deberá fomentar una propuesta inicial de la estructura de los cortes evaluativos así como permitir organizar nuevamente los cortes en caso de inconformidad. Lograr que el sistema se convierta en el espacio que brinde el soporte que hace falta para llevar a cabo una tarea que manualmente se dificulta mucho.

9. Visión

Se desea contar con un sitio Web que sea robusto e interactivo, que sirva de desarrollo para alcanzar un conocimiento en el contexto científico técnico, que responderá a las necesidades de los consumidores, se podrá adquirir conocimiento, tributando positivamente a la hora de toma de decisiones en este contexto.

10. Herramientas utilizadas

- Dreamweaver.
- PostgreSQL v8.4 como servidor de Base de Datos.
- PHP 5 como lenguaje de programación Web.
- Metodología SXP.

Anexo 2: Plantilla Modelo de HU del Negocio

Tabla_10. Actores del negocio

Actor	Descripción
Administrador	Web máster del portal encargado de gestionar la configuración del sitio, administrar los usuarios y la información.
Profesor	Encargado de insertar evaluación y asistencia de los estudiantes.
Coordinador de año	Recoger todas las informaciones de los estudiantes.
Responsable en las carrera	Recoger todas las informaciones de todos los grupos de su carrera.
Responsable en la facultad	Responsable de realizar un resumen evaluativo pero para todas las carreras de esa facultad.
Responsable en la Vicerrectoría Docente	Responsable de realizar un resumen evaluativo de todas las carreras, graficar, mostrar la asignatura con mayor problema.

Anexo 3: Plantilla Lista de Reserva del Producto

Tabla_11. Plantilla Lista de Reserva del Producto

Asignado a	Ítem	Descripción	Estimación	Estimado por
Prioridad			Alta	
Joandris Rivera Legrá (Analista)	1	Gestionar usuarios	2	Desarrollador
Joandris Rivera Legrá (Analista)	2	Gestionar Datos Primarios	1	Desarrollador

Joandris Rivera Legrá (Analista)	3	Autenticar Usuarios	1	AI, Diseñador
Prioridad			Media	
Joandris Rivera legrá (Analista)	4	Gestionar Asistencia	2	Desarrollador
Joandris Rivera legrá (Analista)	5	Gestionar evaluaciones parciales del estudiante	2	AI, Desarrollador
Joandris Rivera legrá (Analista)	6	Gestionar evaluaciones parciales de las áreas	1.5	AI, Desarrollador
Joandris Rivera Legrá (Analista)	7	Mostrar evaluación	2	AI, Desarrollador
Joandris Rivera Legrá (Analista)	8	Gestionar Información	2	AI, Desarrollador
Joandris Rivera Legrá (Analista)	9	Gestionar las gráficas de los resultados de las evaluaciones.	2	Desarrollador
Joandris Rivera Legrá (Analista)	10	Gestionar reportes.	2	Desarrollador
Prioridad			Baja	
RNF (Requisitos No Funcionales)				
Joandris Rivera Legrá, Diseñador	11	Apariencia o Interfaz Externa, Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema. Su interfaz agradable logra captar la confianza de los usuarios así como su identificación con los colores y formatos en que realizan sus acciones. Se tiene en cuenta algunos elementos de diseño como gráficos de encabezamiento, estilos y formatos de texto, paletas de color de los gráficos y colores del fondo.	1 Semana	A, AI
Joandris Rivera Legrá Analista	12	Usabilidad, El proyecto garantizará un acceso fácil y rápido a los usuarios, facilita el trabajo manual realizado hasta el momento. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos acerca de la computadora o de una ambiente Web.	Tiempo total del proyecto	A, AI
Joandris Rivera Legrá, Desarrollador	13	Rendimiento, Los tiempos de carga y procesamiento de la información debe ser lo más rápido posible. El acceso a la base de datos tendrá la rapidez suficiente. El tiempo de respuesta debe ser corto pues se deben generar	Tiempo total del proyecto	A, AI

		pantallas dinámicas, debe tener un grado alto de eficiencia.		
Joandris Rivera Legrá Desarrollador	14	Soporte, La base de datos que utiliza el sistema como medio de almacenamiento de la información está soportada sobre un gestor de bases de datos PostgreSQL y su plataforma es Apache.	1 Semana	A, AI, D

Anexo 4: Plantilla de Historias de Usuarios

Tabla_12.Plantilla Historia de usuario 1.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar Usuarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo	Puntos Reales: 1.5
Descripción: Crear cuentas de usuarios, eliminar, modificar y crear privilegios.	
Observaciones: Se hace referencia a la funcionalidad 1 enumerada en el Listado de Reservas del Producto.	

Tabla_13.Plantilla Historia de usuario 2.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar Datos Primarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Cargar datos de los estudiantes, carrera y facultad, y mostrarlos.	
Observaciones: Se hace referencia a la funcionalidad 2 enumerada en el Listado de	

Reservas del Producto.

Tabla_14.Plantilla Historia de usuario 3.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Autenticar Usuarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 0.7
Descripción: Esta HU tiene la responsabilidad de que el usuario entre al sistema con el nivel de acceso que posee y le permite cambiar su contraseña.	
Observaciones: Se hace referencia a la funcionalidad 3 enumerada en el Listado de Reservas del Producto.	

Tabla_15.Plantilla Historia de usuario 4.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Gestionar Asistencia
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 2
Prioridad en Negocio: Medio	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1.5
Descripción: Esta permite insertar asistencia de los estudiantes, mostrarla y modificarla.	
Observaciones: Se hace referencia a la funcionalidad 4 enumerada en el Listado de Reservas del Producto.	

Tabla_16.Plantilla Historia de usuario 5.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar Evaluaciones parciales de los Estudiantes

Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1.5
Descripción: Esta permite insertar evaluaciones parciales del estudiante, modificarlas y mostrarla.	
Observaciones:	

Tabla_17.Plantilla Historia de usuario 6.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar Evaluaciones parciales de las Áreas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1.5
Riesgo en Desarrollo: Medio	Puntos Reales: 1.5
Descripción: Esta permite insertar evaluaciones parciales de las áreas, modificarlas y Mostrarla.	
Observaciones:	

Tabla_18.Plantilla Historia de usuario 7.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Mostrar Evaluación
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
Descripción: Permite dar evaluación general del estudiante	
Observaciones: Se hace referencia a la funcionalidad 7 enumerada en el Listado de Reservas del Producto.	

Tabla_19.Plantilla Historia de usuario 8.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Gestionar Información
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1.5
Descripción: Permitir que el usuario responsable, pueda insertar, modificar y mostrar informaciones necesarias.	
Observaciones: Se hace referencia a las funcionalidades 8 enumeradas en el Listado de Reservas del Producto.	

Tabla_20.Plantilla Historia de usuario 9.

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Gestionar Gráficas de los resultados
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 2
Descripción: Permitir mostrar las gráficas de los resultados de las evaluaciones y comparar los resultados.	
Observaciones: Se hace referencia a las funcionalidades 8 enumeradas en el Listado de Reservas del Producto.	

Tabla_21.Plantilla Historia de usuario 10.

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Gestionar reportes
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Joandris Rivera Legrá	Iteración Asignada: 4

Prioridad en Negocio: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Brinda reportes de los cortes evaluativos y le permite al usuario imprimir estos reportes.	
Observaciones: Se hace referencia a las funcionalidades 9 enumeradas en el Listado de Reservas del Producto.	

Anexo 5: Plantilla Lista de Riesgos

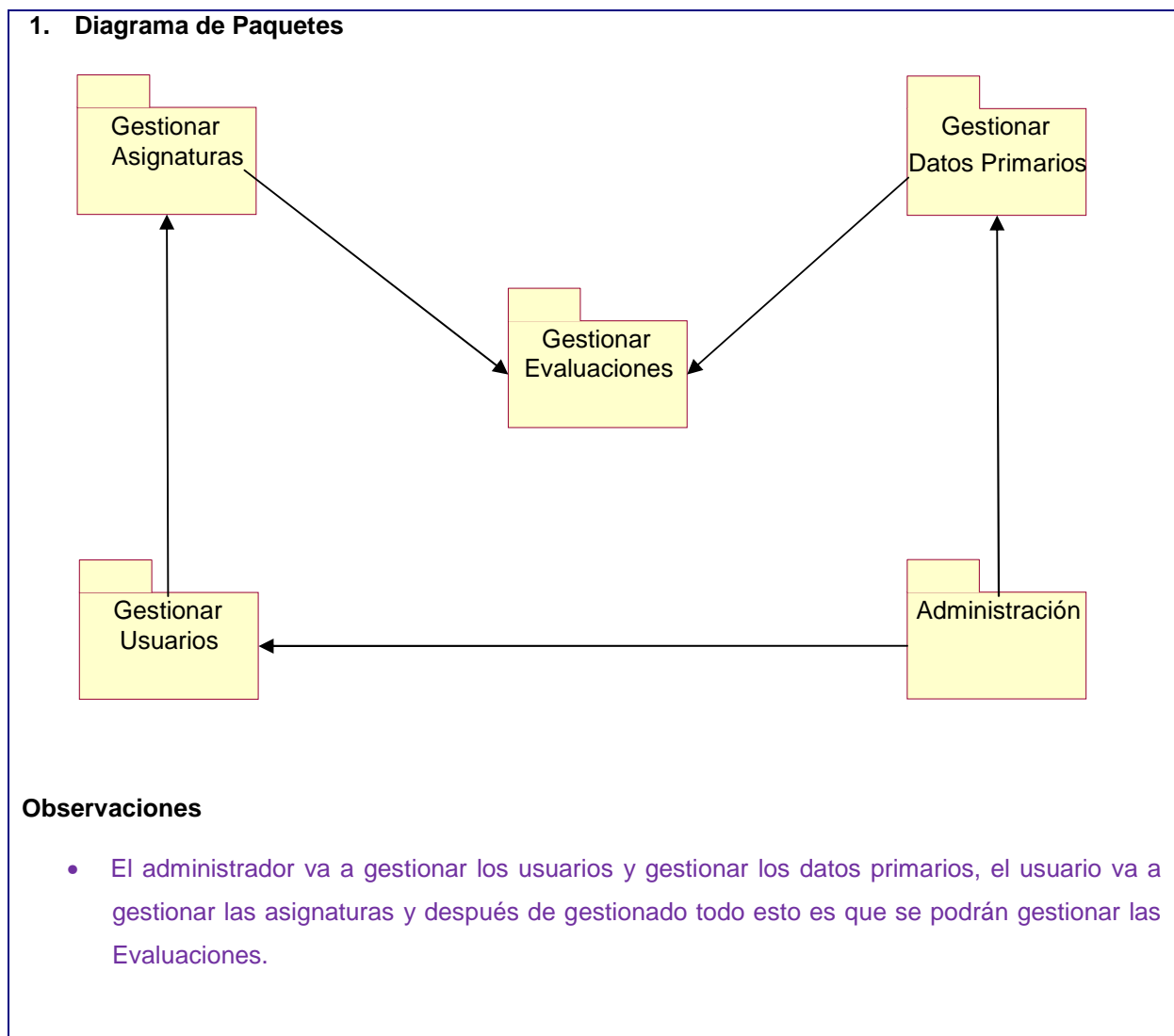
Tabla_22.Plantilla Lista de Riesgos

Riesgo	Tipos de riesgos	Impacto	Descripción	Probabilidad	Efectos	Mitigación del riesgo
El conocimiento del equipo de trabajo es insuficiente para llevar a cabo la tarea a desarrollar.	Personal			Alta	Serio	Impartir cursos de Capacitación.
Los clientes no perciben el alcance de los cambios en los requerimientos	Requerimientos			Media	Tolerable	Es necesario tener un encuentro con el cliente y explicarle en qué consisten los cambios
El tiempo para desarrollar el proyecto esta subestimado.	Estimación			Baja	Serio	Reunirse una vez más con el cliente y discutir el tema del tiempo de desarrollo.
Cambios de requerimientos que precisan modificaciones en el diseño	Requerimientos		Existencia de más cambios de requerimientos de los previstos inicialmente	Media	Serias	Rastrear la información para valorar el impacto de los requerimientos, maximizar la información oculta en ellos
El cliente no pueda participar en revisiones y en reuniones	Comunicación		El cliente no se encuentra disponible cuando se realizan las reuniones.	Media	Serias	Explicarle a cliente la importancia de su presencia cuando se van a discutir temas sobre el proyecto.
Herramientas defectuosas en los momentos críticos de desarrollo	Herramientas	Despreciables Crítico Catastrófico Indeseable		Alta	Serio	
Personal clave enfermo o no disponible en momentos críticos	Personal			Media	Serias	Reorganizar el equipo de tal forma que se solapen el trabajo y los miembros comprendan el trabajo de los demás

Retrasos en la Especificación.	Estimación		Retrasos en las especificaciones de interfaces esenciales	Media	Insignificante	Alertar al cliente de las dificultades potenciales y las posibilidades de retraso
Rotación de personal.	Personal	Despreciable Marginal Crítico Catastrófico	Equipo Corto de Personal para el desarrollo del Proyecto	Alta	Insignificante	Alertar al cliente de las dificultades, y las posibilidades de retraso

Anexo 6: Plantilla Modelo de Diseño

Tabla_23.Plantilla Modelo de Diseño



Anexo 7: Tarjetas CRC

Tabla_24. Tarjeta CRC #2 Ges_datos_pri

Nombre de la clase: Ges_datos_pri	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar datos primarios	Datos_primarios

Tabla_25.Tarjeta CRC#3 Auten_usuario

Nombre de la clase: Auten_usuario	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Autenticar usuarios	Contraseña

Tabla_26.Tarjeta CRC#4 Gestión_asis

Nombre de la clase: Gestión_asis	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar asistencia.	Asistencia

Tabla_27.Tarjeta CRC#5 Ges_eva_pa_es

Nombre de la clase: Ges_eva_pa_es	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar evaluaciones parciales de los estudiantes	Evaluaciones

Tabla_28.Tarjeta CRC #6 Ges_eva_pa_areas

Nombre de la clase: Ges_eva_pa_areas	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar evaluaciones parciales de las áreas	Evaluar_are

Tabla_29.Tarjeta CRC #7 Ges_eva_gen

Nombre de la clase: Ges_eva_gen	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar evaluación	Evaluar_ge

Tabla_30.Tarjeta CRC #8 Informaciones

Nombre de la clase: Informaciones	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar Información	Informacion

Tabla_31. Tarjeta CRC #9 Resultados

Nombre de la clase: Resultados	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar gráficas	Gestionar_graficas

Tabla_32. Tarjeta CRC #10 Gestion_repor

Nombre de la clase: Gestion_repor	
Tipo de la clase: Lógica del negocio	
Responsabilidades:	Colaboradores:
Gestionar reportes	Gestionar_reportes

Anexo 8: Tareas de Ingeniería

Tabla_33. Tareas de ingeniería # 2 Modificar cuentas de Usuarios

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Modificar cuentas de usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 13 de abril 2011	Fecha fin: 14 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir modificar cuentas de usuarios existentes.	

Tabla_34. Tareas de ingeniería # 3 Eliminar cuentas de usuarios

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Eliminar cuentas de usuarios	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 15 de abril 2011	Fecha fin: 16 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir eliminar las cuentas de usuarios existentes.	

Tabla_35. Tareas de ingeniería # 4 Cargar Datos

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 2
Nombre Tarea: Cargar Datos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 17 de abril 2011	Fecha fin: 18 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir cargar los datos de los estudiantes, grupo, carrera y facultad y en ese momento eliminar los anteriores.	

Tabla_36. Tareas de ingeniería # 5 Mostrar datos

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 2
Nombre Tarea: Mostrar datos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 19 de abril 2011	Fecha fin: 20 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir mostrar los datos que se cargaron.	

Tabla_37. Tareas de ingeniería # 6 Entrar al sistema

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 3
Nombre Tarea: Entrar al sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Fecha inicio: 21 de abril 2011	Fecha fin: 22 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir que el usuario entre con el nivel de acceso que posee.	

Tabla_38. Tareas de ingeniería # 7 Cambiar contraseña

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 3
Nombre Tarea: Cambiar contraseña	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 23 de abril 2011	Fecha fin: 24 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario cambiar su contraseña.	

Tabla_39. Tareas de ingeniería # 8 Insertar asistencia

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 4
Nombre Tarea: Insertar asistencia	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 25 de abril 2011	Fecha fin: 26 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario que sea profesor insertar asistencia de sus estudiantes.	

Tabla_40. Tareas de ingeniería # 9 Mostrar asistencia

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 4
Nombre Tarea: Mostrar asistencia	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 27 de abril 2011	Fecha fin: 28 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario ver la asistencia de los estudiantes ya insertadas.	

Tabla_41. Tareas de ingeniería # 10 Modificar asistencia

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 4
Nombre Tarea: Modificar asistencia	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 29 de abril 2011	Fecha fin: 30 de abril 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle modificar las asistencias de los estudiantes ya insertadas.	

Tabla_42. Tareas de ingeniería # 11 Insertar evaluación de los estudiantes

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 5
Nombre Tarea: Insertar evaluación de los estudiantes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 1 de mayo 2011	Fecha fin: 2 de Mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario que sea profesor insertar la evaluación de cada uno de sus estudiantes.	

Tabla_43. Tareas de ingeniería # 12 Mostrar evaluación de los estudiantes

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 5
Nombre Tarea: Mostrar evaluación de los estudiantes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 3 de mayo 2011	Fecha fin: 4 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario ver las evaluaciones de los estudiantes ya insertadas.	

Tabla_44. Tareas de ingeniería # 13 Modificar evaluación de los estudiantes

Tarea de Ingeniería	
---------------------	--

Número Tarea: 13	Número Historia de Usuario: 5
Nombre Tarea: Modificar evaluación de los estudiantes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 5 de mayo 2011	Fecha fin: 6 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle modificar las evaluaciones de los estudiantes ya insertadas.	

Tabla_45. Tareas de ingeniería # 14 Insertar evaluaciones de las áreas

Tarea de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: 6
Nombre Tarea: Insertar evaluaciones de las áreas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 7 de mayo 2011	Fecha fin: 8 de Mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitir insertar las evaluaciones de las áreas	

Tabla_46. Tareas de ingeniería # 15 Mostrar evaluaciones del las áreas

Tarea de Ingeniería	
Número Tarea: 15	Número Historia de Usuario: 6
Nombre Tarea: Mostrar evaluaciones de las áreas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 9 de mayo 2011	Fecha fin: 10 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle ver las evaluaciones de las áreas	

Tabla_47. Tareas de ingeniería # 16 Modificar evaluaciones de las áreas

Tarea de Ingeniería	
Número Tarea: 16	Número Historia de Usuario: 6
Nombre Tarea: Modificar evaluaciones de las áreas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 11 de mayo 2011	Fecha fin: 12 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle modificar las evaluaciones de las áreas	

Tabla_48. Tareas de ingeniería # 17 Dar evaluación

Tarea de Ingeniería	
Número Tarea: 17	Número Historia de Usuario: 7
Nombre Tarea: Dar evaluación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.0

Fecha inicio: 13 de mayo 2011	Fecha fin: 14 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe dar evaluaciones generales a los estudiantes.	

Tabla_49. Tareas de ingeniería # 18 Mostrar evaluación

Tarea de Ingeniería	
Número Tarea: 18	Número Historia de Usuario: 7
Nombre Tarea: Mostrar evaluación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.0
Fecha inicio: 15 de mayo 2011	Fecha fin: 16 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe mostrar las evaluaciones generales de los estudiantes.	

Tabla_50. Tareas de ingeniería # 19 Insertar información

Tarea de Ingeniería	
Número Tarea: 19	Número Historia de Usuario: 8
Nombre Tarea: Insertar informaciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 17 de mayo 2011	Fecha fin: 18 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario responsable insertar informaciones necesarias.	

Tabla_51. Tareas de ingeniería # 20 Mostrar información

Tarea de Ingeniería	
Número Tarea: 20	Número Historia de Usuario: 8
Nombre Tarea: Mostrar informaciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 19 de mayo 2011	Fecha fin: 20 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario ver las informaciones ya insertadas.	

Tabla_52. Tareas de ingeniería # 21 Modificar información

Tarea de Ingeniería	
Número Tarea: 21	Número Historia de Usuario: 8
Nombre Tarea: Modificar informaciones	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 21 de mayo 2011	Fecha fin: 22 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe permitirle al usuario modificar las informaciones ya	

insertadas.

Tabla_53. Tareas de ingeniería # 22 Mostrar gráficas

Tarea de Ingeniería	
Número Tarea: 22	Número Historia de Usuario: 9
Nombre Tarea: Mostrar gráficas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.0
Fecha inicio: 23 de mayo 2011	Fecha fin: 24 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema debe mostrar las gráficas de los resultados de los cortes evaluativos.	

Tabla_54. Tareas de ingeniería # 23 Comparar resultados

Tarea de Ingeniería	
Número Tarea: 23	Número Historia de Usuario: 9
Nombre Tarea: Comparar resultados	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.0
Fecha inicio: 25 de mayo 2011	Fecha fin: 26 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema nos brinda una breve comparación de los resultados.	

Tabla_55. Tareas de ingeniería # 24 Brindar reportes

Tarea de Ingeniería	
Número Tarea: 24	Número Historia de Usuario: 10
Nombre Tarea: Brindar reportes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.0
Fecha inicio: 27 de mayo 2011	Fecha fin: 28 de mayo 2011
Programador responsable: Joandris Rivera legrá	
Descripción: El sistema nos brinda los reportes en formato PDF de los cortes evaluativos.	

Anexo 9: Pruebas de Aceptación

Tabla_56. Prueba de aceptación #2 para la HU “Gestionar Datos Primarios”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_HU2_P2	Nombre Historia de Usuario: Gestionar Datos Primarios

Nombre de la persona que realiza la prueba: Joandris Rivera Legrá
Descripción de la Prueba: Se realiza una prueba para comprobar que se carguen correctamente todos los datos necesarios (estudiantes, grupo, carrera y facultad), y se eliminen los anteriores, en caso de que se estén trabajando todavía con esos datos el sistema debe dar un mensaje de error.
Condiciones de Ejecución: Los clientes deben probar que se muestren todos los datos correctamente.
Entrada / Pasos de ejecución: Se solicita cargar datos y luego se verifica si cargaron correctamente, si funcionan como se desea. y si elimina correctamente cuando se solicite eliminar datos.
Resultado Esperado: El sistema no exhibe errores.
Evaluación de la Prueba: Satisfactoria

Tabla_57. Prueba de aceptación #3 para la HU “Autenticar usuarios”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU3_P3	Nombre Historia de Usuario: Autenticar usuarios
Nombre de la persona que realiza la prueba: Joandris Rivera Legrá	
Descripción de la Prueba: Se realiza una prueba para verificar que el usuario entre al sistema con el nivel correcto, en caso de entrada mal de los datos, el sistema debe dar un mensaje de error y verificar que el cambio de contraseña se haga correctamente.	
Condiciones de Ejecución: Los clientes debe probar que esta correcta la entrada al sistema y que no hay problema en el cambio de contraseña.	
Entrada / Pasos de ejecución: Los usuarios entran al sistema con el acceso correcto y se verifica si al cambiar la contraseña no hay problema.	
Resultado Esperado: El sistema no exhibe errores.	
Evaluación de la Prueba: Satisfactoria	

Tabla_58. Prueba de aceptación #4 para la HU “Gestionar asistencia”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU4_P4	Nombre Historia de Usuario: Gestionar asistencia

Nombre de la persona que realiza la prueba: Joandris Rivera legrá
Descripción de la Prueba: Se realiza una prueba para gestionar la asistencia de los estudiantes, en caso de modificar si la asistencia ya ha sido usada el sistema debe dar un mensaje de error.
Condiciones de Ejecución: Los clientes deben probar que se inserten, se modifique y se muestren correctamente las asistencias de los estudiantes.
Entrada / Pasos de ejecución: Se solicita insertar evaluación de los estudiantes y luego se verifica que la asistencia se halla insertado bien y se muestre correctamente, después se solicita modificar para probar que se modifique correctamente.
Resultado Esperado: El sistema no exhibe errores.
Evaluación de la Prueba: Satisfactoria

Tabla_59. Prueba de aceptación #5 para la HU “Gestionar evaluación parcial de los estudiantes”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_HU5_P5	Nombre Historia de Usuario: Gestionar evaluación parcial de los estudiantes
Nombre de la persona que realiza la prueba: Joandris Rivera legrá	
Descripción de la Prueba: Se realiza una prueba para gestionar las evaluaciones de los estudiantes, en caso de modificar si la evaluación ya ha sido usada el sistema debe dar un mensaje de error.	
Condiciones de Ejecución: Los clientes deben probar que se inserten, se modifique y se muestren correctamente las evaluaciones de los estudiantes.	
Entrada / Pasos de ejecución: Se solicita insertar evaluación de los estudiantes y luego se verifica que la evaluación se halla insertado bien y se muestre correctamente, después se solicita modificar para probar que se modifique correctamente.	
Resultado Esperado: El sistema no exhibe errores.	
Evaluación de la Prueba: Satisfactoria	

Tabla_60. Prueba de aceptación #6 para la HU “Gestionar evaluación parcial de las áreas”

Caso de Prueba de Aceptación

Código Caso de Prueba: PL_ HU5_P6	Nombre Historia de Usuario: Gestionar evaluación parcial de las áreas
Nombre de la persona que realiza la prueba: Joandris Rivera legrá	
Descripción de la Prueba: Se realiza una prueba para gestionar las evaluaciones de las áreas, en caso de modificar si la evaluación ya ha sido usada el sistema debe dar un mensaje de error.	
Condiciones de Ejecución: Los clientes deben probar que se inserten, se modifique y se muestren correctamente las evaluaciones de las áreas.	
Entrada / Pasos de ejecución: Se solicita insertar evaluación y luego se verifica que la evaluación se haya insertado bien y se muestre correctamente, después se solicita modificar para probar que se modifique correctamente.	
Resultado Esperado: El sistema no exhibe errores.	
Evaluación de la Prueba: Satisfactoria	

Tabla_61. Prueba de aceptación #7 para la HU “Mostrar evaluación”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU7_P7	Nombre Historia de Usuario: Mostrar evaluación
Nombre de la persona que realiza la prueba: Joandris Rivera Legrá	
Descripción de la Prueba: Se realiza una prueba para verificar que el sistema brinde correctamente las evaluaciones, en caso de que falte alguna evaluación de la que había que insertarse anteriormente el sistema dará una nota con el nombre de la evaluación que falta.	
Condiciones de Ejecución: Los clientes deben verificar que el sistema brinde correctamente las evaluaciones y que las notas que brinda sean correctas.	
Entrada / Pasos de ejecución: Los clientes deben verificar en los reportes de los cortes evaluativos las evaluaciones y si están correctamente.	
Resultado Esperado: El sistema no exhibe errores.	
Evaluación de la Prueba: Satisfactoria	

Tabla_62. Prueba de aceptación #7 para la HU “Gestionar información”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU6_P8	Nombre Historia de Usuario: Gestionar información

Nombre de la persona que realiza la prueba: Joandris rivera Legrá
Descripción de la Prueba: Se realiza una prueba para gestionar la información de los estudiantes, en caso de modificar si la información ya ha sido usada el sistema debe dar un mensaje de error.
Condiciones de Ejecución: El cliente debe probar que todos los aspectos relacionados con la gestión de las informaciones cumplan con las perspectivas esperadas.
Entrada / Pasos de ejecución: Se solicita insertar evaluación de los estudiantes y luego se verifica que la información se haya insertado bien y se muestre correctamente, después se solicita modificar para probar que se modifique correctamente.
Resultado Esperado: El sistema no exhibe errores.
Evaluación de la Prueba: Satisfactoria

Tabla_63. Prueba de aceptación #8 para la HU “Gestionar gráficas de los resultados”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU8_P9	Nombre Historia de Usuario: Gestionar gráficas de los resultados
Nombre de la persona que realiza la prueba: Joandris Rivera legrá	
Descripción de la Prueba: Se realiza una prueba para verificar la Gestión de las gráficas de los resultados de los cortes evaluativos y las comparaciones de los resultados.	
Condiciones de Ejecución: El cliente debe probar que todos los aspectos relacionados con Gestionar gráficas de los resultados cumplan con las perspectivas esperadas.	
Entrada / Pasos de ejecución: Se solicita Mostrar Corte Evaluativo por parte del responsable en la Vicerrectoría Docente y se prueba si están correctamente y si la comparación es la adecuada.	
Resultado Esperado: El sistema no exhibe errores.	
Evaluación de la Prueba: Satisfactoria	

Tabla_64. Prueba de aceptación #9 para la HU “Gestionar reportes”

Caso de Prueba de Aceptación	
Código Caso de Prueba: PL_ HU9_P10	Nombre Historia de Usuario: Gestionar reportes

Nombre de la persona que realiza la prueba: Joandris Rivera legrá
Descripción de la Prueba: Se realiza una prueba para verificar la Gestión de reportes.
Condiciones de Ejecución: Los clientes deben probar que los reportes están correctos y que se impriman correctamente.
Entrada / Pasos de ejecución: Se solicita la Gestión de los reportes y luego se procede a la impresión.
Resultado Esperado: El sistema no exhibe errores.
Evaluación de la Prueba: Satisfactoria

Anexo 10: Modelo de la Base de Datos

