**DOCUMENTACIÓN OFICIAL**

**EMPRESA:**

*Colegio Parroquial Privado Subvencionado “Monseñor Juan Wiesen”*

**AUTORES:**

*Pedro Damián González Villalba*

*Mirian Magdalena Saucedo Gómez*

**AÑO: 2012**

***Descripción de la Empresa***

La empresa en cuestión, es una institución educativa denominada legalmente como “Colegio Parroquial Privado Subvencionado Monseñor Juan Wiesen” ubicado en la localidad de Fram, departamento de Itapúa, a 50km de Encarnación; dedicado a la formación intelectual y moral de jóvenes. Cuenta con la Educación Escolar Básica (desde el séptimo hasta el noveno grado, con dos secciones cada grado), además cuenta con dos Bachilleratos: Bachillerato Científico con Énfasis en Ciencias Básicas y Bachillerato Técnico en Administración de Negocios. Posee en total aproximadamente 250 alumnos y 25 docentes.

Actualmente no cuenta con un sistema académico informático.

***Primera Entrevista***

**Fecha:** 20 de Setiembre de 2012.-

**Entrevistado:** Lic. Carlos Daniel Tkachik (Director)

**Objetivo:** Conocer la estructura del colegio y su sistema académico.

1. *¿Hace cuántos años se inició el colegio?*

El colegio se inició en el año 1970, exactamente hace 42 años.

1. *¿La institución es un colegio nacional o privado?*

El colegio “Monseñor Juan Wiesen” actualmente es un Colegio Parroquial Privado Subvencionado. En donde el Bachillerato Técnico, el segundo y tercer año es privado.

1. *¿Con qué niveles cuenta la institución, cuenta con el nivel primario y secundario?*

La institución cuenta con el nivel primario pero solamente el Tercer Ciclo que abarca desde el séptimo hasta el noveno grado, y el nivel secundario desde el primer y tercer año, donde se posee dos énfasis de bachilleratos: el Bachillerato Técnico en Administración de Negocios y el Bachillerato con énfasis en Ciencias Básicas.

1. *Actualmente, ¿con cuántos alumnos cuenta el colegio?*

La institución cuenta aproximadamente con más de 250 alumnos.

1. *¿Con cuántos docentes cuenta aproximadamente?*

El plantel de docentes está compuesto aproximadamente por 25 docentes.

1. *¿En el plantel administrativo, con cuántos personales cuenta?*

En el plantel administrativo se cuenta con 3 personales, incluyendo la dirección serían exactamente 4 personales.

1. *¿La institución cuenta con algún sistema informático?*

No, la institución no cuenta con ningún tipo de sistema informático.

1. *¿Qué área le gustaría informatizar?*

Me gustaría informatizar la parte académica de la institución, el control de asistencia de los alumnos, docentes y todo el plantel del colegio.

1. *¿Cuál es el objetivo que desea lograr al informatizar dicha área?*

El objetivo principal es de contar con un mejor control de asistencia especialmente de los alumnos, así también que los docentes logren lo mismo. Contar con una mejor planificación de las actividades escolares y brindar a los alumnos y porque no los padres un mejor control sobre sus hijos estando al tanto sobre la situación académica de éstos. Así también contar con un sistema informático sería un gran crecimiento para la institución ya que estaríamos actualizados con lo que es el mundo de la tecnología.

1. *¿Quiénes serían los principales usuarios del programa que desea?*

Los principales usuarios serían la secretaría, los docentes y la dirección. Así también los alumnos para que puedan ver su situación académica.

1. *Nos podría contar a grandes rasgos, ¿cuáles serían las funciones que desea que cumpla el programa a ser desarrollado?*

Las funciones principales que desearía que realice son poder realizar un informe mensual de la asistencia, que los docentes puedan realizar su planificación de tareas y exámenes, que los alumnos tengan acceso a dicha información así también a ver su situación académica, esto incluye el proceso de clase, los puntajes acumulados, etc.

***Descripción de la Metodología Utilizada***

**Metodología RUP**

La metodología a ser usada para la documentación del sistema será la metodología RUP, llamada así por sus siglas en inglés Rational Unified Process. RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Sus características principales son:

➢ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).

➢ Pretende implementar las mejores prácticas en Ingeniería de Software.

➢ Desarrollo iterativo e incremental.

➢ Administración de requisitos.

➢ Uso de arquitectura basada en componentes.

➢ Control de cambios.

➢ Modelado visual del software.

➢ Verificación de la calidad del software.

➢ Diseñado para ser flexible y extensible.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo.

En RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura para que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura.

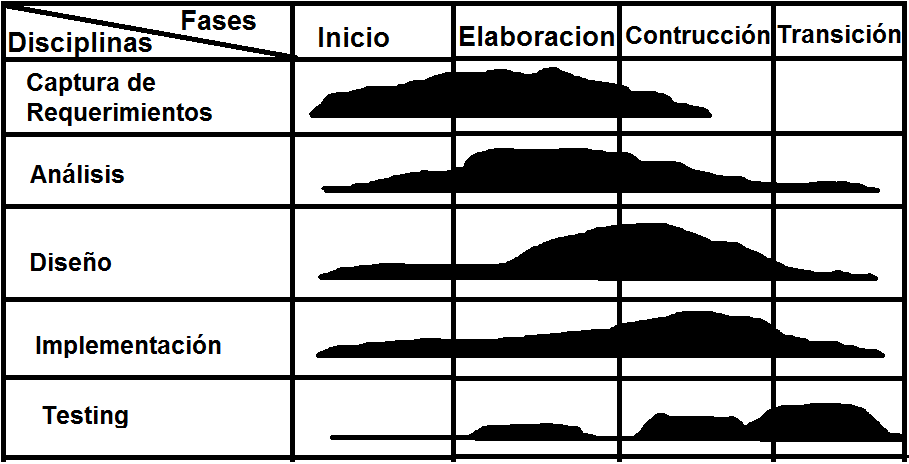
Divide en 4 fases el desarrollo del software, como se puede observar en la figura 1:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.

- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.

- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

- Transición: El objetivo es llegar a obtener el reléase del proyecto.



Figura

Es importante aclarar que en cada etapa, las iteraciones se basan en el modelo de ciclo de vida en cascada.

RUP se basa en el desarrollo iterativo e incremental. Y este proceso complicado lleva una adecuada documentación. Para hacer más fácil el trabajo, dividen a las actividades en disciplinas y dentro de éstas proponen el uso de modelos. Cada disciplina atacará cierta actividad o tarea desde un punto de vista.

Es aquí donde entra UML. Como UML ofrece un amplio conjunto de diagramas para representar las ideas desde diferentes, y complementarios, punto de vista, RUP aprovecha esta ventaja y adoptan a UML como una herramienta más para realizar, diseñar y documentar el desarrollo de sistemas.

En síntesis, RUP propone usar UML para llevar la documentación del sistema, facilitar la etapa del diseño y posterior construcción o desarrollo, transmitir ideas y ayudar al equipo a comunicarlas. Ahora bien, UML tiene mayor sentido cuando se está hablando de un análisis, diseño y programación bajo el paradigma OO (Orientado a Objetos), aunque uno puede, si así lo desea, extrapolar el concepto de un diagrama para transmitir una idea fuera del paradigma OO. Como por ej., el diagrama de actividad que en ocasiones se lo emplea para representar el flujo de información y los procesos de un área o departamento de una empresa.

Las disciplinas de una metodología RUP conllevan los flujos de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina, estas disciplinas se dividen en dos grupos: las primarias y las de apoyo. Las primarias son las necesarias para la realización de un proyecto de software, aunque para proyectos no muy grandes se pueden omitir algunas; entre ellas se tienen: Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue. Las de apoyo son las que como su nombre lo indica sirven de apoyo a las primarias y especifican otras características en la realización de un proyecto de software; entre estas se tienen: Entorno, Gestión del Proyecto, Gestión de Configuración y Cambios.

Existen algunos elementos con los cuales se puede trabajar en el RUP, ahora en los elementos, entre estos se tienen: Flujos de Trabajo, Detalle de los Flujos de Trabajo, Actores, Actividades y Artefactos. Los actores son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP. Los artefactos son el resultado parcial o final que es producido y usado por los actores durante el proyecto. Son las entradas y salidas de las actividades, realizadas por los actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo.

La elaboración de distintos diagramas y artefactos siguiendo la metodología RUP proveen una fácil ejecución del proceso de elaboración de un Sistema de Software, ya que describen cómo está estructurado el sistema desde diferentes perspectivas orientadas a los diferentes involucrados en un proyecto.

Se puede reducir el tiempo de desarrollo de un Sistema de Software, aplicando la metodología RUP y UML ya que permite lograr de una manera fiable y rápida el desarrollo del Sistema deseado.

El tener todo el procedimiento de desarrollo de un Sistema de Software, es una herramienta necesaria y efectiva para administrarlo; y así contar con una visión unificada de todo el proceso, con lo que se facilita la implementación del mismo.

***Descripción de los Actores***

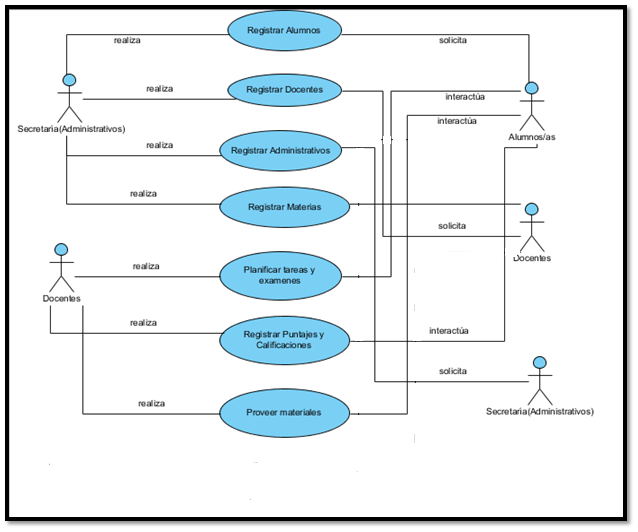
* **Dirección:** Persona encargada de supervisar y controlar la actividad académica y general de la institución.
* **Secretaría:** Persona encargada de la parte administrativa de la institución.
* **Docentes:** Persona encargada de enseñar en la institución.
* **Alumnos/as:** Personas que adquieren los servicios (enseñanza) de la institución.

***Modelo de Negocios***

***Casos de Uso:***

1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Materias
4. Registrar Administrativos
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio

***Diagrama de Modelo de Negocios***



* **REGISTRO DE ALUMNOS**

**CUMN 1**

|  |  |
| --- | --- |
| **Nombre** | Registro de alumnos |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Alumnos |
| **Descripción** | El alumno solicita su inscripción a la institución, por ende la secretaría registra los datos del interesado. |
| **Precondiciones** | La institución precisa de alumnos |
| **Flujo Normal** | 1. El alumno rellena el formulario de inscripción 2. La secretaría recepciona el formulario y procesa los datos. |
| **Pos condiciones** | La institución cuenta con alumnos inscriptos y registrados. |
| **Flujo Alternativo** | El alumno no cumple con los requisitos para ser inscripto. |

* **REGISTRO DE DOCENTES**

**CUMN 2**

|  |  |
| --- | --- |
| **Nombre** | Registro de Docentes |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Docentes |
| **Descripción** | La institución requiere contar los datos de todos sus docentes, por ende la secretaría realiza el registro de los mismos. |
| **Precondiciones** | La institución precisa de los datos precisos de sus docentes. |
| **Flujo Normal** | 1. La institución contrata docentes. 2. Los docentes presentan los documentos requeridos. 3. La secretaría procesa los datos de los docentes. |
| **Pos condiciones** | La secretaría realiza el registro de todos los docentes. |
| **Flujo Alternativo** | El docente no cumple con los requisitos y no es contratado por la institución. |

* **REGISTRAR ADMINISTRATIVOS**

**CUMN 3**

|  |  |
| --- | --- |
| **Nombre** | Registrar Administrativos |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Administrativos |
| **Descripción** | La institución requiere contar los datos de todos sus personales administrativos, por ende la secretaría realiza el registro de los mismos. |
| **Precondiciones** | La institución precisa de los datos precisos de sus personales administrativos. |
| **Flujo Normal** | 1. La institución contrata personales para el sector administrativo. 2. Los personales presentan los documentos requeridos. 3. La secretaría procesa los datos de los administrativos. |
| **Pos condiciones** | La secretaría registra los datos de los administrativos. |
| **Flujo Alternativo** | El personal administrativo no cumple con los requisitos y no es contratado por la institución. |

* **REGISTRAR MATERIAS**

**CUMN 4**

|  |  |
| --- | --- |
| **Nombre** | Registrar Materias |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Docentes |
| **Descripción** | La secretaría registra las materias para cada grado y curso. |
| **Precondiciones** | El sistema académico necesita registrar las materias, porque de éste depende el funcionamiento principal del sistema. |
| **Flujo Normal** | 1. La secretaría revisa la malla curricular de los respectivos grados y cursos de diferente énfasis. 2. La secretaría registra las materias a cada grado y curso. |
| **Pos condiciones** | Cada curso cuenta con sus materias registradas. |
| **Flujo Alternativo** | La secretaría no registra correctamente las materias. |

* **PLANIFICACIÓN DE TAREAS Y EXAMENES**

**CUMN 5**

|  |  |
| --- | --- |
| **Nombre** | Planificación de tareas y exámenes |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Docentes – Alumnos |
| **Descripción** | El docente debe realizar la calendarización y/o planificación de sus actividades escolares junto con sus alumnos. |
| **Precondiciones** | 1. La institución precisa que los docentes planifiquen sus actividades para una mejor organización. 2. Los alumnos requieren contar con un calendario de actividades para tener un seguimiento correcto de sus responsabilidades y deberes. |
| **Flujo Normal** | 1. El docente planifica fecha de exámenes, trabajos, proyectos, etc., junto con sus alumnos. 2. Se realizan los exámenes y todas las actividades de evaluación. 3. El docente evalúa el desarrollo de los mismos. |
| **Pos condiciones** | Los alumnos y docentes cuentan con la planificación de sus actividades escolares, todo esto por etapa escolar. |
| **Flujo Alternativo** | No se realizó correctamente la planificación o no fue entregado. |

* **REGISTRO DE PUNTAJES Y CALIFICACIONES**

**CUMN 6**

|  |  |
| --- | --- |
| **Nombre** | Registro de Puntajes y Calificaciones |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Docentes – Alumnos |
| **Descripción** | Los alumnos precisan tener un seguimiento de sus puntajes acumulados en cada materia, por ende el docente registra todos los puntajes obtenidos. |
| **Precondiciones** | Los alumnos precisan conocer sus puntajes acumulados por etapa. |
| **Flujo Normal** | 1. El docente evalúa todos los prácticos, proyectos y exámenes desarrollados. 2. El docente registra los puntajes acumulados y la calificación final por etapa. |
| **Pos condiciones** | Los alumnos pueden tener acceso a sus puntajes acumulados y así ver y analizar su situación escolar. |
| **Flujo Alternativo** | El docente no registra los puntajes y calificación del alumno. |

* **PROVEER MATERIALES DE ESTUDIO**

**CUMN 7**

|  |  |
| --- | --- |
| **Nombre** | Proveer materiales de estudio |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Docentes – Alumnos |
| **Descripción** | Los docentes proveen a los alumnos de los materiales de estudio acerca de los temas desarrollados en clase, a través del sistema académico informático. |
| **Precondiciones** | Los alumnos necesitan poseer los materiales de estudio de cada clase desarrolla respectivamente de cada materia. |
| **Flujo Normal** | 1. El docente desarrolla su clase con materiales didácticos. 2. Los alumnos/as requieren de los materiales de estudio para contar con los mismos para que estos puedan estudiar. 3. El docente sube a la base de datos del sistema académico el material. |
| **Pos condiciones** | Los alumnos cuentan con el material didáctico desarrollado en clase. |
| **Flujo Alternativo** | El docente no cuenta con un material didáctico o no sube correctamente a la base de datos del sistema. |

***Modelo de Requisitos***

***Modelo de Requisitos***

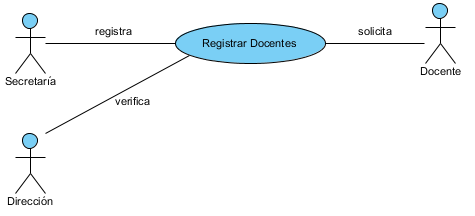
1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Administrativos
4. Registrar Materias
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio

***Casos de Uso de Modelo de Requisito***

1. ***Registrar Alumnos***

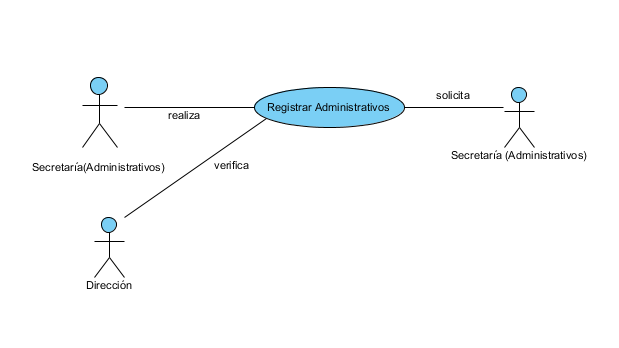
|  |  |
| --- | --- |
| **Nombre** | Registrar Alumnos |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría - Dirección – Alumnos |
| **Descripción** | Este caso se inicia cuando el alumno llega a la institución para inscribirse a un grado o curso. |
| **Precondiciones** | 1. Un alumno se inscribe a la institución previamente haber rellenado el formulario de inscripción. 2. La institución acepta al alumno/a. |
| **Flujo Normal** | 1. La secretaría recibe el formulario rellenado. 2. Verifica si los datos son correctos conforme a su documento presentado. 3. No se verifica ninguna irregularidad 4. La secretaría registra al alumno en el sistema académico. |
| **Pos condiciones** | Se registró al alumno en el sistema académico de la institución. |
| **Flujo Alternativo** | 1. No se encontraron concordancias entre el formulario presentado por el alumno y sus documentos respectivos. 2. La secretaría comunica a la dirección de la irregularidad. 3. La dirección comunica al alumno de la irregularidad en su formulario. 4. El alumno regulariza su situación para poder aceptado por la institución. |

1. ***Registrar Docentes***

******

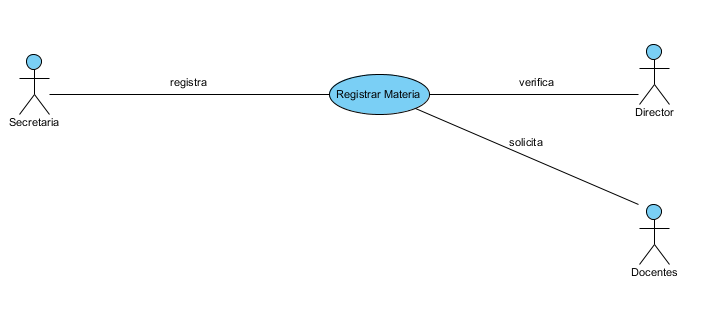
|  |  |
| --- | --- |
| **Nombre** | Registrar Docentes |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Dirección - Docentes |
| **Descripción** | Este caso se inicia cuando la institución contrata un nuevo docente, o en todo caso un docente que ya forma parte del colegio es contratado para llevar otra materia más. |
| **Precondiciones** | La institución contrató un docente. |
| **Flujo Normal** | 1. La dirección realiza el contrato con el docente. 2. La secretaría recibe dicho contrato. 3. El docente es registrado en el sistema por la secretaría. |
| **Pos condiciones** | El docente cuenta con una cuenta en el sistema. |
| **Flujo Alternativo** | 1. No se llegó a un acuerdo entre la dirección y el docente sobre el contrato. 2. La secretaría encuentra algunas irregularidades dentro del contrato. 3. Informa a la dirección sobre dicha irregularidad. 4. La dirección se vuelve a reunir con el docente para verificar la irregularidad del contrato y así poder llegar a un acuerdo. |

1. ***Registrar Administrativos***

******

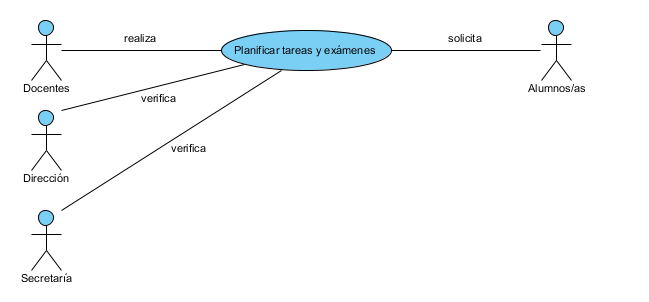
|  |  |
| --- | --- |
| **Nombre** | Registrar Administrativos |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Dirección - Administrativos |
| **Descripción** | Este caso se inicia cuando la institución contrata un nuevo personal administrativo. |
| **Precondiciones** | La institución contrató un personal administrativo. |
| **Flujo Normal** | 1. La dirección realiza el contrato con el personal. 2. La secretaría recibe dicho contrato. 3. El personal administrativo es registrado en el sistema por la secretaría. |
| **Pos condiciones** | El personal administrativo cuenta con una cuenta en el sistema. |
| **Flujo Alternativo** | 1. No se llegó a un acuerdo entre la dirección y el personal administrativo sobre el contrato. 2. La secretaría encuentra algunas irregularidades dentro del contrato. 3. Informa a la dirección sobre dicha irregularidad. 4. La dirección se vuelve a reunir con el personal para verificar la irregularidad del contrato y así poder llegar a un acuerdo. |

1. ***Registrar Materias***

******

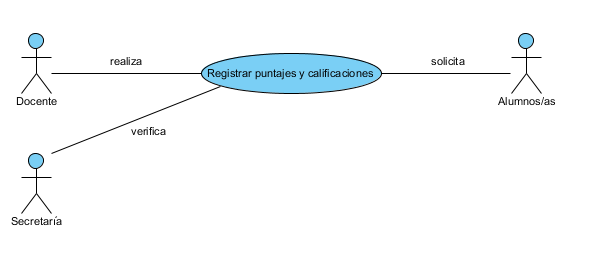
|  |  |
| --- | --- |
| **Nombre** | Registrar Materias |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Docentes-Dirección |
| **Descripción** | Este caso de uso comienza cuando cambia la malla curricular de algún curso, entonces nuevas materias son asignadas para el grado. |
| **Precondiciones** | 1. La institución debe agregar una nueva materia a un curso. 2. Una materia de la malla curricular ya no se desarrollara en la institución por lo que hay que eliminarla del sistema. |
| **Flujo Normal** | 1. Por algún motivo se decide el cambio de la malla curricular por lo que el director solicita a la secretaria actualizar las materias de cada curso. 2. La secretaria registra las nuevas materias, asignándolos a un curso y a un docente, y elimina las antiguas. |
| **Pos condiciones** | El sistema académico informático se encuentra actualizado. |
| **Flujo Alternativo** | La secretaría no registra correctamente las materias. |

1. ***Planificar tareas y exámenes***

******

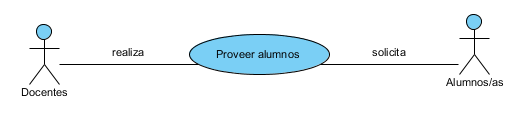
|  |  |
| --- | --- |
| **Nombre** | Planificar tareas y exámenes |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Secretaría – Docentes-Dirección - Alumnos |
| **Descripción** | Este caso se inicia apenas arranque cada etapa de la actividad escolar, en donde los alumnos planifican las fechas de exámenes y los trabajos prácticos a realizarse durante la etapa. Esta actividad es supervisada por la dirección y la secretaría. |
| **Precondiciones** | 1. Se inicia una nueva etapa de cada curso o grado en el año. 2. La secretaría y la dirección solicita al docente una planificación por etapa de la actividad escolar. 3. El docente planifica junto con sus alumnos las fechas de exámenes, entrega de trabajos prácticos, etc. |
| **Flujo Normal** | 1. El docente planifica las tareas con los alumnos. 2. Registra las actividades a realizarse en el sistema académico. 3. Los alumnos tienen acceso al registro de dicha actividad. |
| **Pos condiciones** | Se registraron las tareas y exámenes para la etapa escolar, mediante el cual los alumnos tienen acceso a la planificación por etapa realizado conjuntamente con el docente. |
| **Flujo Alternativo** | El docente realizó la planificación pero no subió correctamente al sistema. |

1. ***Registrar puntajes y calificaciones.***

******

|  |  |
| --- | --- |
| **Nombre** | Registrar puntajes y calificaciones |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Docentes - Secretaría - Alumnos |
| **Descripción** | Este caso sucede cuando el docente verifica todas las medios de evaluación utilizadas para la evaluar el desempeño del alumno. |
| **Precondiciones** | Se realiza un examen o la entrega de trabajos prácticos, deberes, proyectos, etc. |
| **Flujo Normal** | 1. El docente realiza la corrección de trabajos, deberes y/o exámenes. 2. El docente registra en el sistema el puntaje obtenido por el alumno. 3. La nota y el puntaje es verificado por la secretaría y así también los alumnos tienen acceso a él. |
| **Pos condiciones** | El docente registró los puntajes acumulados y esto puede ser accedido por los mismos, por la secretaría y por los alumnos. |
| **Flujo Alternativo** | 1. El docente no realiza la corrección de trabajos y/o exámenes. 2. La secretaría no encuentra concordancia entre la planilla entregado por el docente y lo registrado en el sistema. 3. La secretaría avisa al docente de dicha irregularidad. 4. El docente vuelve a verificar y modifica el puntaje en caso de que sea necesario. |

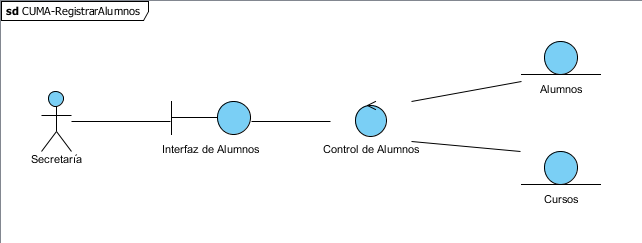
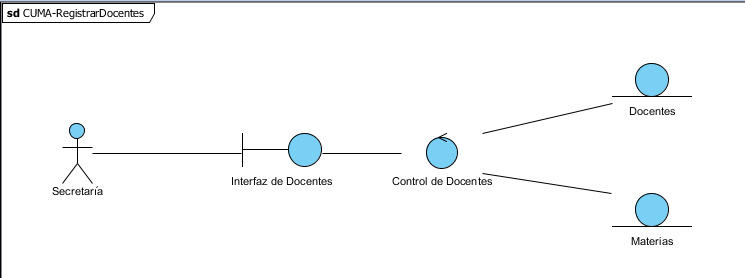
***7- Proveer Materiales de Estudio***

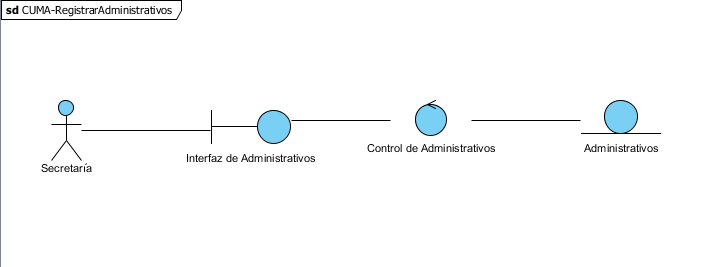
******

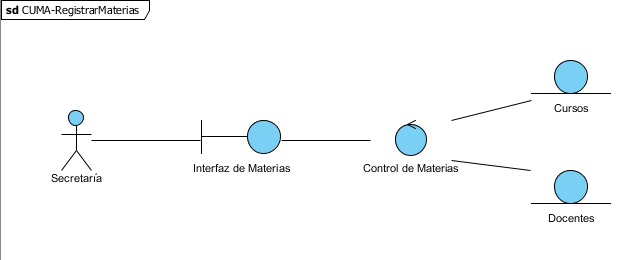
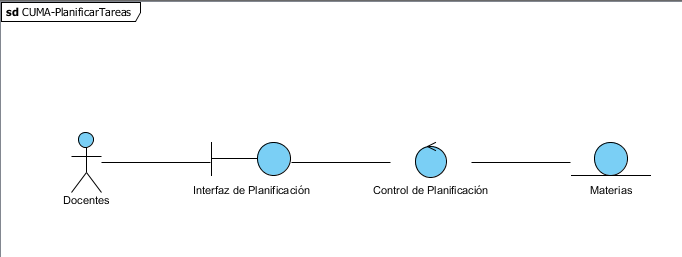
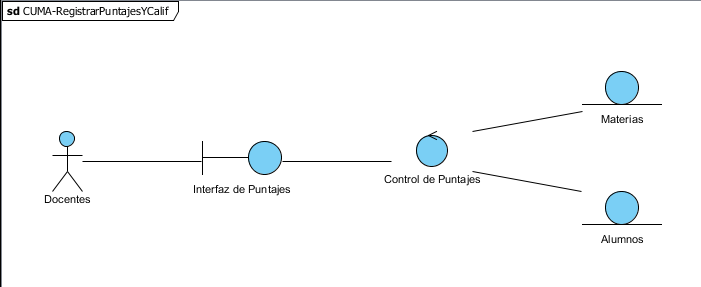
|  |  |
| --- | --- |
| **Nombre** | Proveer Materiales de Estudio |
| **Autor** | IISS 2012 |
| **Fecha** | Setiembre 2012 |
| **Actores** | Docente – Alumnos |
| **Descripción** | Este caso se inicia siempre que se termina de desarrollar una clase o un tema específico de una materia donde el profesor haya utilizado algún material didáctico y que los alumnos han de precisar con dicho material. |
| **Precondiciones** | 1. El docente utilizó un material didáctico para el desarrollo de su clase. 2. Los alumnos necesitan de dicho material didáctico para poder estudiar. |
| **Flujo Normal** | 1. El docente termina de desarrollar su clase 2. Sube al sistema el material utilizado por el mismo |
| **Pos condiciones** | El alumno tiene acceso a dicho material y lo descarga para poder utilizarlo. |
| **Flujo Alternativo** | El docente no subió el material utilizado en clase al sistema informático. |

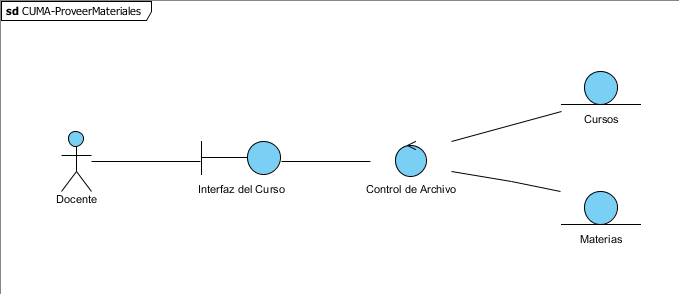
***Casos de Usos de Modelo de Análisis***

***Casos de Uso de Modelo de Análisis (CUMA)***

1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Administrativos
4. Registrar Materias
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio
8. ***Registrar Alumnos***
9. ***Registrar Docentes***
10. ***Registrar Administrativos***

******

1. ***Registrar Materias***
2. ***Planificar tareas y exámenes***
3. ***Registrar Puntajes y Calificaciones***
4. ***Proveer Materiales de Estudio***

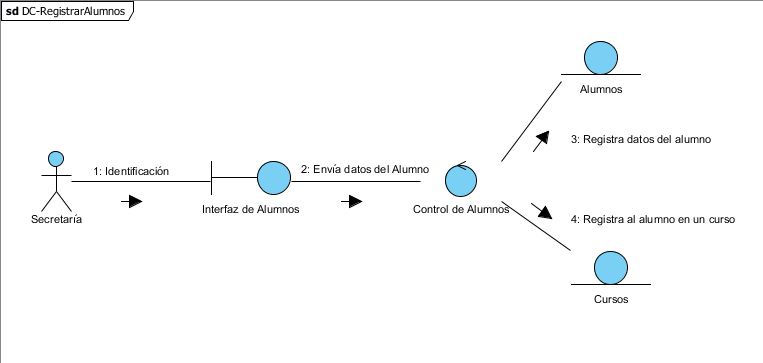
******

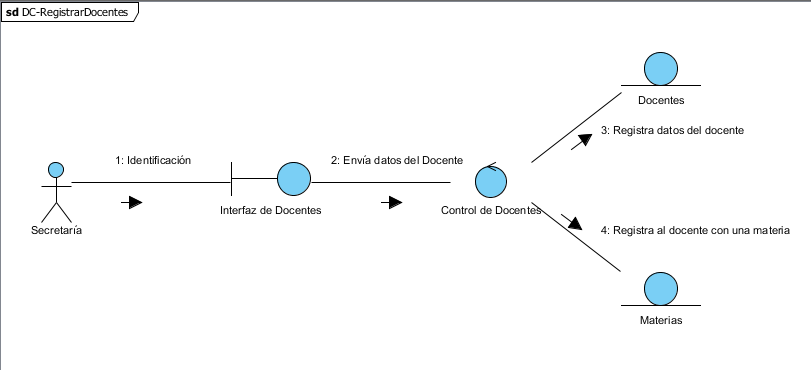
***Diagrama de Colaboración***

***Diagramas de Colaboración***

1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Administrativos
4. Registrar Materias
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio
8. ***Registrar Alumnos***

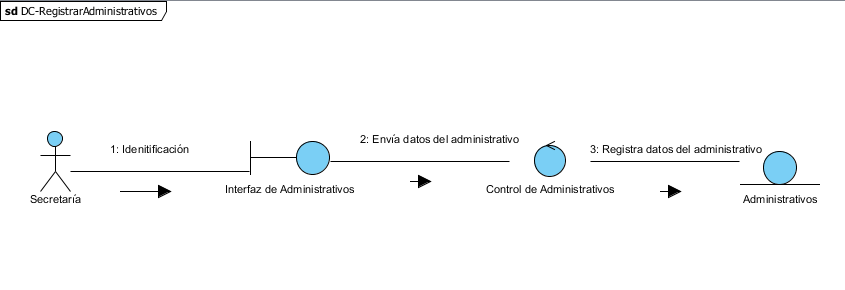
|  |
| --- |
| 1. Se identifica el usuario que realizará y carga el registro de los alumnos del colegio. |
| 1. Se envía los datos del registro de alumnos al control de alumnos. |
| 1. Se registra el ingreso de datos de alumno. |
| 1. Se registra el ingreso de un alumno a un curso. |

******

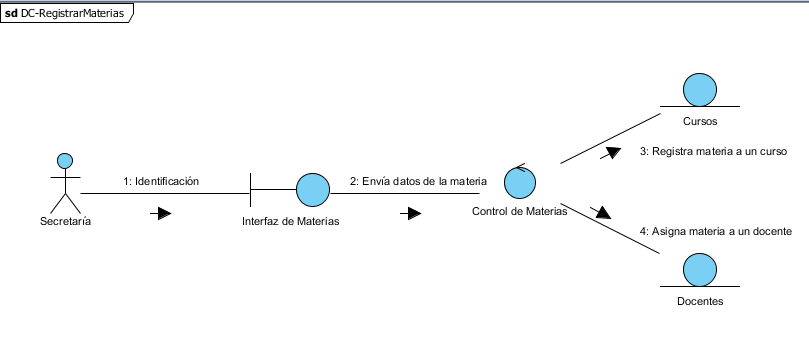
1. ***Registrar Docentes***

|  |
| --- |
| 1. Se identifica el usuario que realizará y carga el registro de los docentes del colegio. |
| 1. Se envía los datos del registro de docentes al control de docentes. |
| 1. Se registra el ingreso de datos del docente. |
| 1. Se registra el ingreso de un docente con una o más materias. |

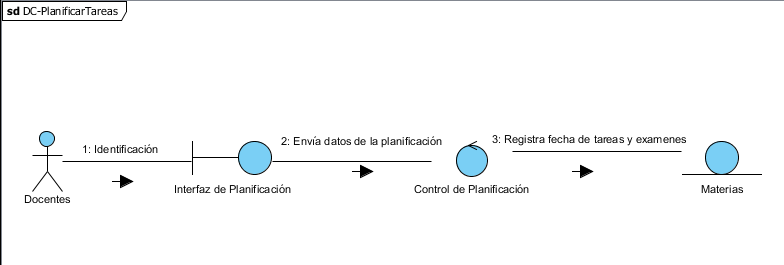
1. ***Registrar Administrativos***

******

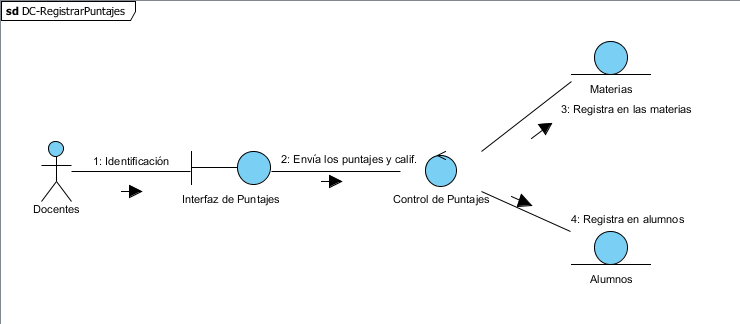
|  |
| --- |
| 1. Se identifica el usuario que realizará y carga el registro de los administrativos del colegio. |
| 1. Se envía los datos del registro de administrativos al control de docentes. |
| 1. Se registra el ingreso de datos del administrativo. |

1. ***Registrar Materias***

|  |
| --- |
| 1. Se identifica el usuario que realizará y carga el registro de todas las materias. |
| 1. Se envía los datos del registro de materias al control de materias. |
| 1. Se registra el ingreso de una materia con el curso correspondiente. |
| 1. Se registra el ingreso de una materia con un docente. |

1. ***Planificar tareas y exámenes***

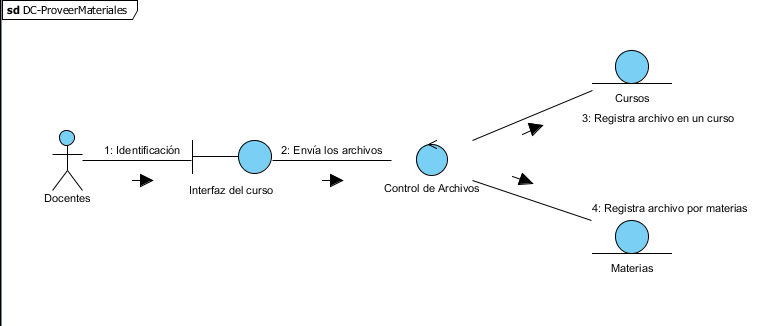
|  |
| --- |
| 1. Se identifica el usuario que realizará y la planificación de tareas y exámenes con las fechas correspondientes a llevarse a cabo cada uno. |
| 1. Se envía los datos de la planificación al control de planificación. |
| 1. Se registra el ingreso de fecha de las tareas y exámenes en las materias correspondientes |

1. ***Registrar Puntajes y Calificaciones***

|  |
| --- |
| 1. Se identifica el usuario que realizará y carga los puntajes acumulados y la calificación obtenida del alumno. |
| 1. Se envía los datos del registro de puntajes y calificaciones al control de puntajes. |
| 1. Se registra el ingreso de los puntajes y/o calificaciones en la materia correspondiente. |
| 1. Se registra el ingreso de los puntajes y/o calificaciones del alumno correspondiente. |

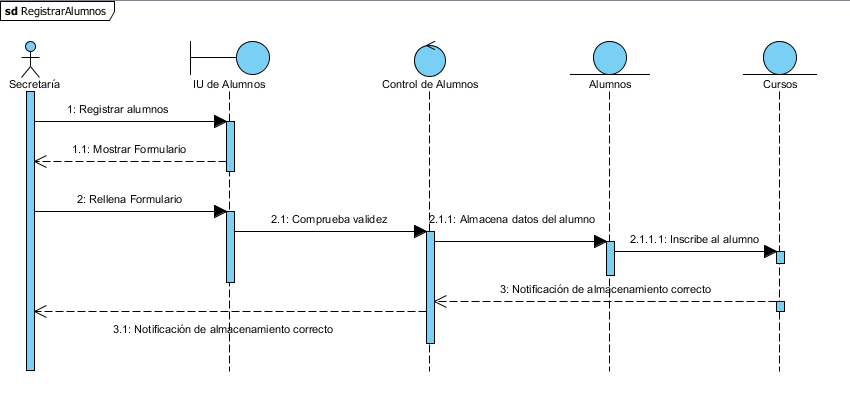
1. ***Proveer materiales de estudio***

|  |
| --- |
| 1. Se identifica el usuario que realizará la carga de materiales en el sistema y realiza dicha acción. |
| 1. Se envía los datos del envío de archivos al control de archivos. |
| 1. Se registra el ingreso de un archivo a un curso correspondiente. |
| 1. Se registra el ingreso de un archivo por la materia correspondiente. |

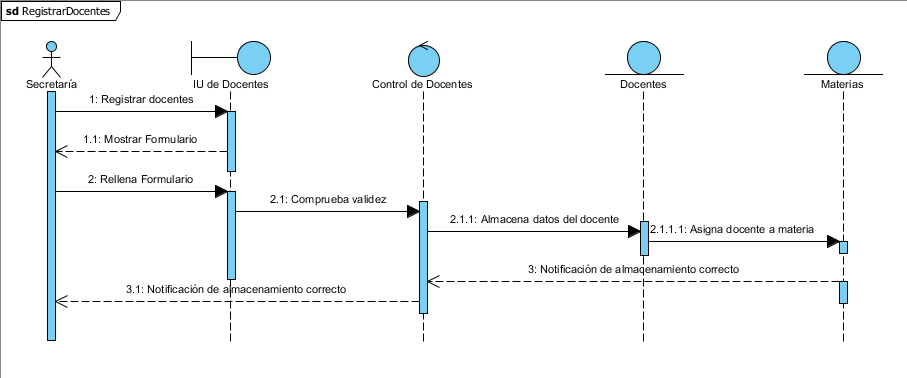
******

***Diagrama de Secuencias***

***Diagramas de Secuencias***

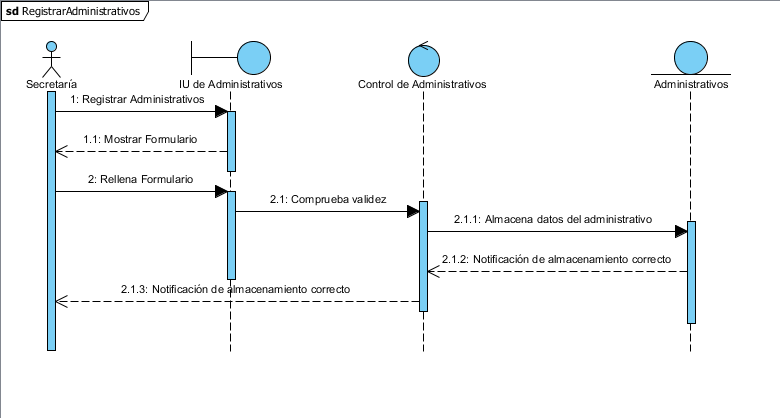
1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Administrativos
4. Registrar Materias
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio
8. ***Registrar Alumnos***

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. La secretaria va pulsar sobre el botón *Registrar Alumno* 2. El sistema muestra un formulario que contiene los campos para ingresar los datos del alumno. 3. La secretaria rellena el formulario con los datos del alumno. 4. El sistema comprueba la validez de los datos ingresados por la secretaria y los almacena en la base de datos. 5. El sistema muestra a la secretaria una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite a la secretaria corregir dichos errores. |

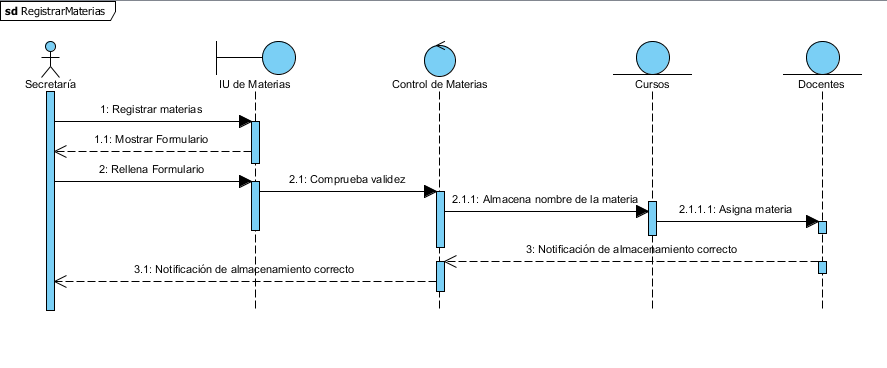
1. ***Registrar Docentes***

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. La secretaria va pulsar sobre el botón *Registrar Docente* 2. El sistema muestra un formulario que contiene los campos para ingresar los datos del docente. 3. La secretaria rellena el formulario con los datos del docente. 4. El sistema comprueba la validez de los datos ingresados por la secretaria y los almacena en la base de datos. 5. El sistema muestra a la secretaria una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite a la secretaria corregir dichos errores. |

1. ***Registrar Administrativos***

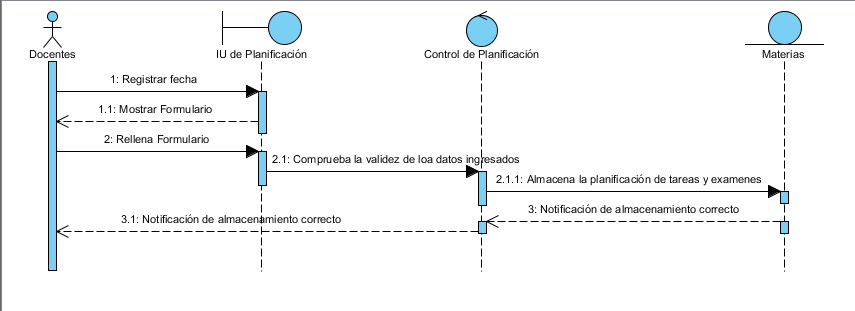
******

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. La secretaria va pulsar sobre el botón *Registrar Administrativo* 2. El sistema muestra un formulario que contiene los campos para ingresar los datos del administrativo. 3. La secretaria rellena el formulario con los datos del personal contratado. 4. El sistema comprueba la validez de los datos ingresados por la secretaria y los almacena en la base de datos. 5. El sistema muestra a la secretaria una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite a la secretaria corregir dichos errores. |

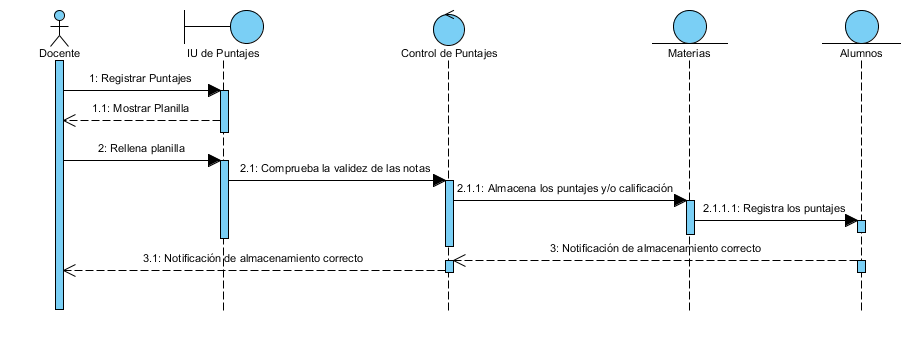
1. ***Registrar Materias***

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. La secretaria va pulsar sobre el botón *Registrar Materia* 2. El sistema muestra un formulario que contiene los campos para ingresar el nombre de la materia y otros datos del mismo. 3. La secretaria rellena dichos campos con los datos correspondientes. 4. El sistema comprueba la validez de los datos ingresados por la secretaria y los almacena en la base de datos. 5. El sistema muestra a la secretaria una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite a la secretaria corregir dichos errores. |

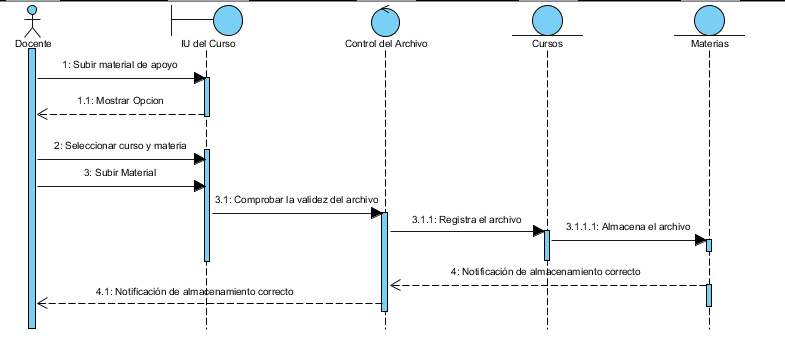
1. ***Planificar tareas y exámenes***

******

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. El docente va pulsar sobre el botón *Registrar Fecha* 2. El sistema muestra un formulario que contiene los campos para ingresar la fecha y los detalles de la planificación, ya sea entrega de trabajos o un examen. 3. El docente rellena dichos campos con los datos correspondientes. 4. El sistema comprueba la validez de los datos ingresados por el docente y los almacena en la base de datos. 5. El sistema muestra al docente una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite al docente corregir dichos errores. |

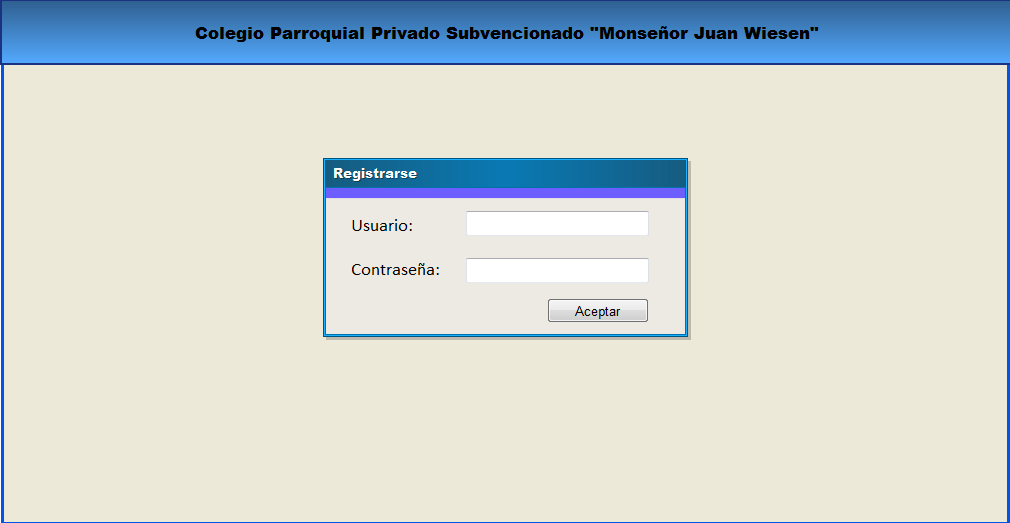
1. ***Registrar Puntajes y Calificaciones***

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. El docente va pulsar sobre el botón *Registrar Puntaje* 2. El sistema muestra una planilla de los alumnos que llevan la materia que el docente desarrolla. 3. El docente rellena la planilla con los datos correspondientes. 4. El sistema comprueba la validez de los datos ingresados por el docente y los almacena en la base de datos. 5. El sistema muestra al docente una notificación de que los datos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 4A- El sistema comprueba la validez de los datos ingresados, y marca en la planilla los errores que este contiene mostrando un mensaje de error y permite al docente corregir dichos errores. |

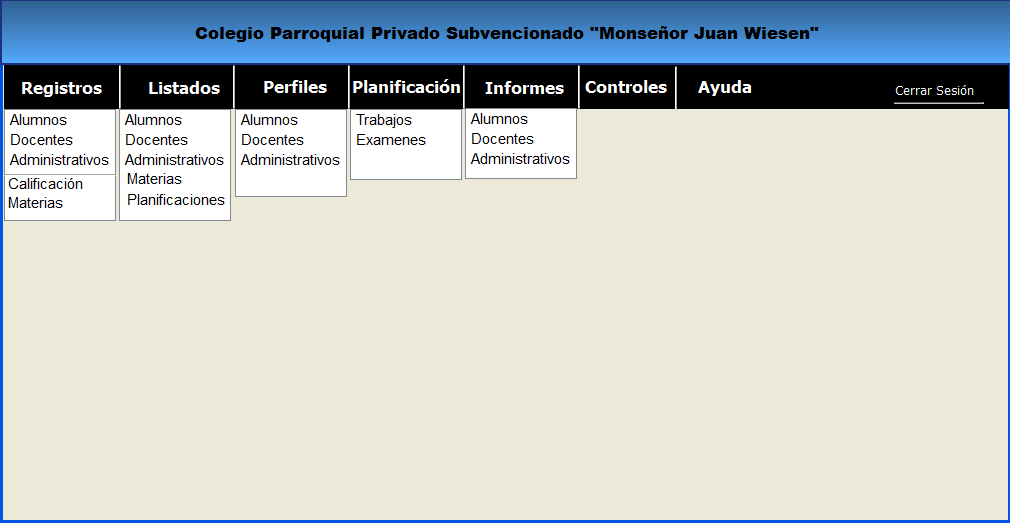
1. ***Proveer materiales de estudio***

|  |  |
| --- | --- |
| ***Flujo Normal*** | 1. El docente va pulsar sobre el botón *Subir Material*. 2. El sistema muestra un formulario con las opciones para subir un material al curso correspondiente. 3. El docente selecciona un curso y la materia correspondiente. 4. El docente sube el material. 5. El sistema comprueba la validez del archivo seleccionado y los almacena. 6. El sistema muestra al docente una notificación de que los archivos fueron almacenados correctamente. |
| ***Flujo Alternativo*** | 5A- El sistema comprueba la validez de los archivos seleccionados, y marca en el formulario los errores que este contiene mostrando un mensaje de error y permite al docente corregir dichos errores. |

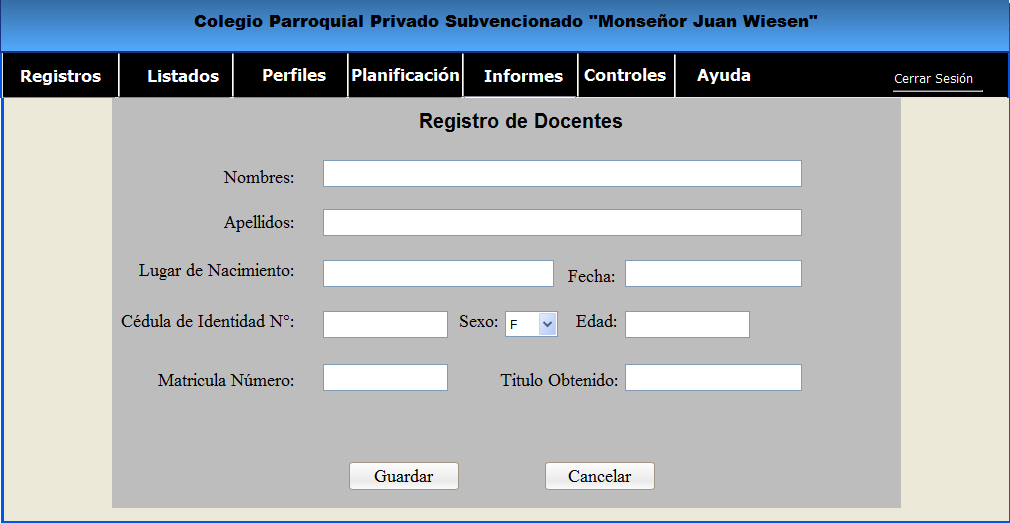
***Diseño de Interfaces de Usuario***

* **Interfaz de Loguin**

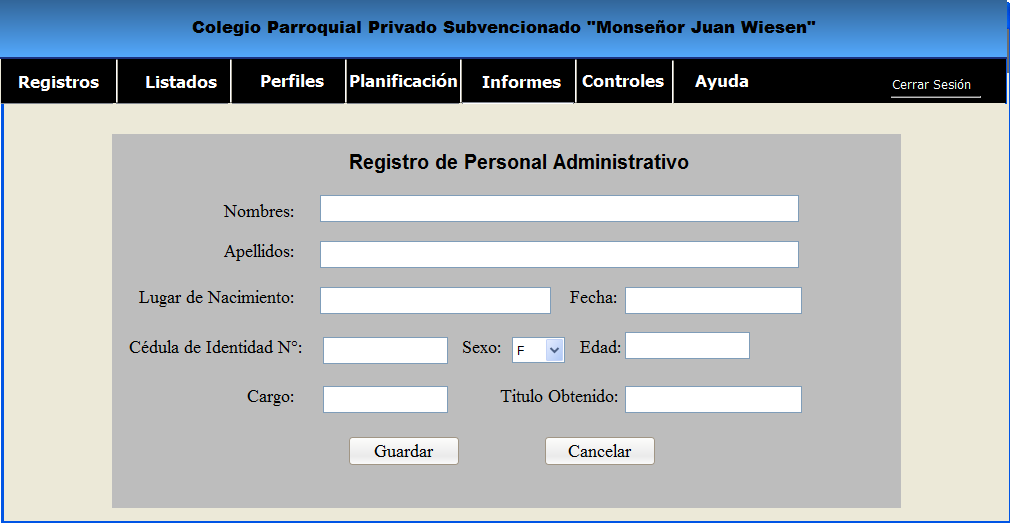
***Descripción:*** Interfaz Principal para loguearse

* **Interfaz Principal**
* **Interfaz de Registro de Alumnos**

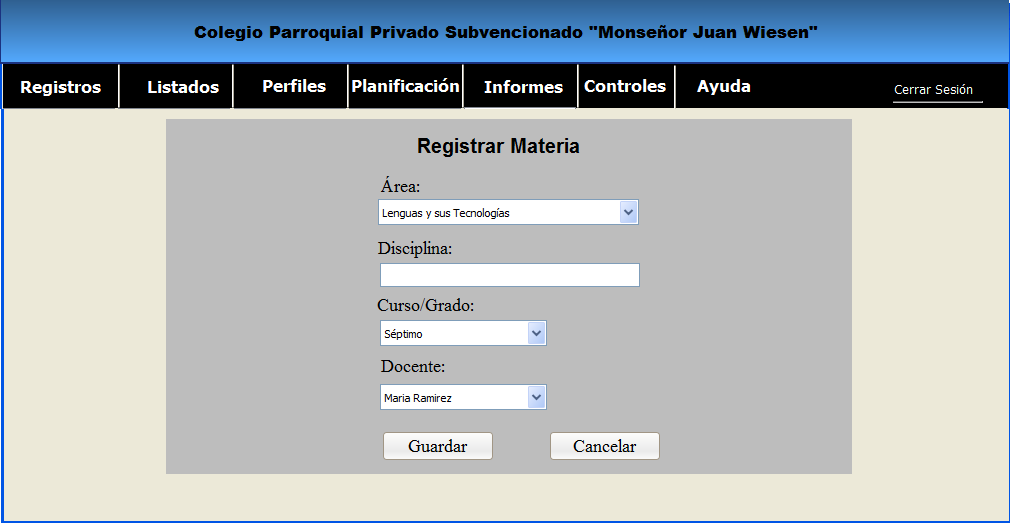
***Descripción:*** Interfaz de registro de alumnos donde tiene acceso la secretaría para registrar los datos del alumno inscripto previamente haber rellenado el formulario de inscripción en papel.

* **Interfaz de Registro de Docentes**

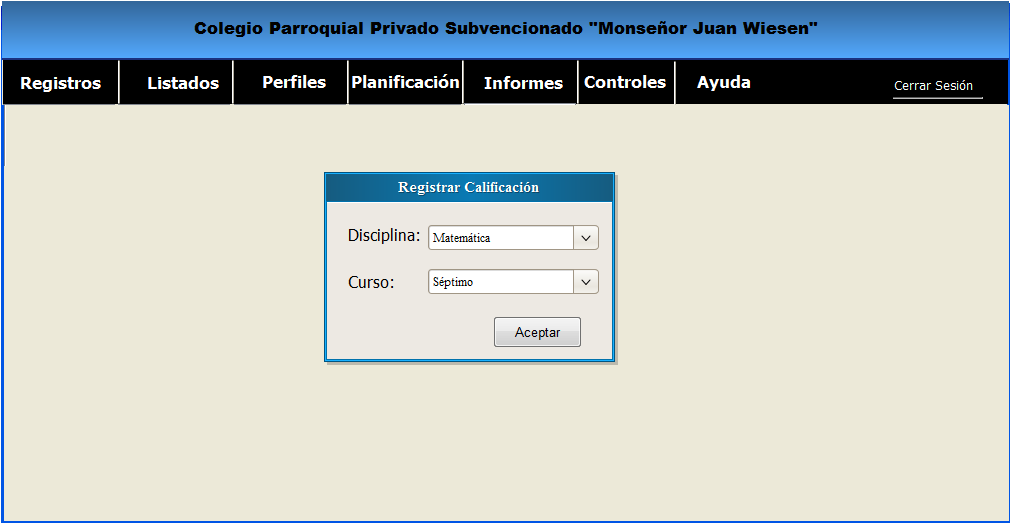
***Descripción:*** Interfaz de registro de docentes donde tiene acceso la secretaría para registrar los datos del docente contratado por la institución previamente haber rellenado el formulario de datos en papel.

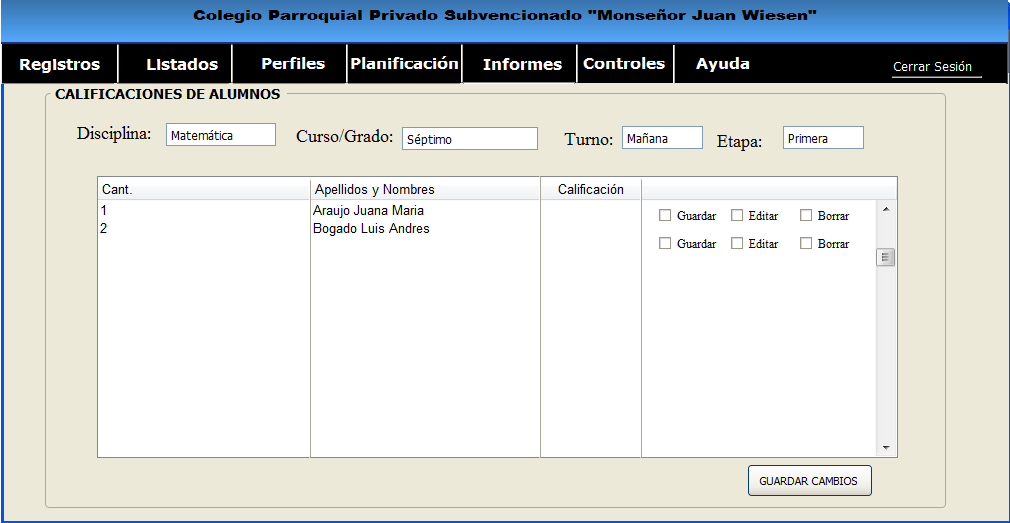
* **Interfaz de Registro de Personal Administrativo**

***Descripción:*** Interfaz de registro de personal administrativo donde tiene acceso la secretaría para registrar los datos del personal nuevo contratado por la institución previamente haber rellenado el formulario de datos en papel.

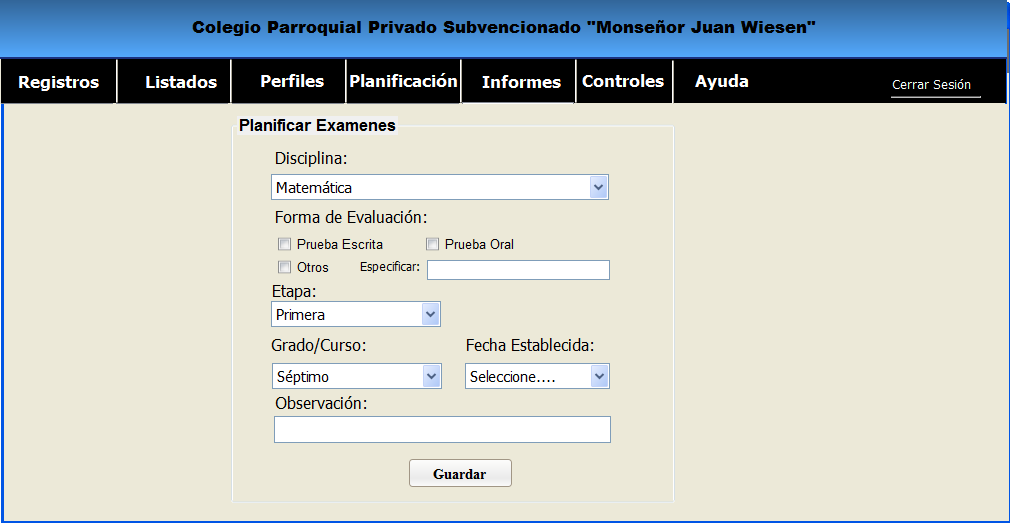
* **Interfaz de Registro de Materias**

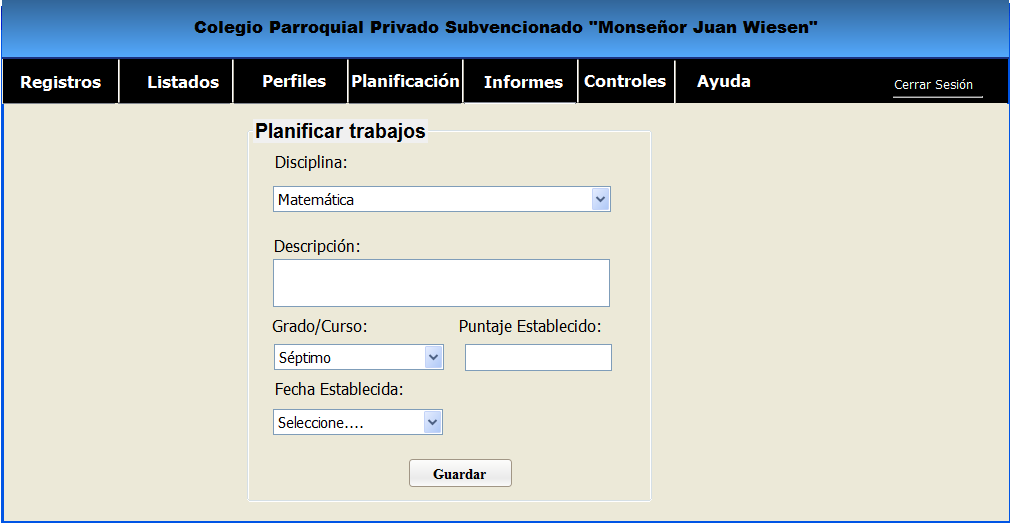
***Descripción:*** Interfaz de registro de materias donde tiene acceso la secretaría para registrar las materias por grado y/o curso y son las respectivas áreas y el docente encargado de la disciplina.

* **Interfaz de Registro de Puntajes y Calificaciones**

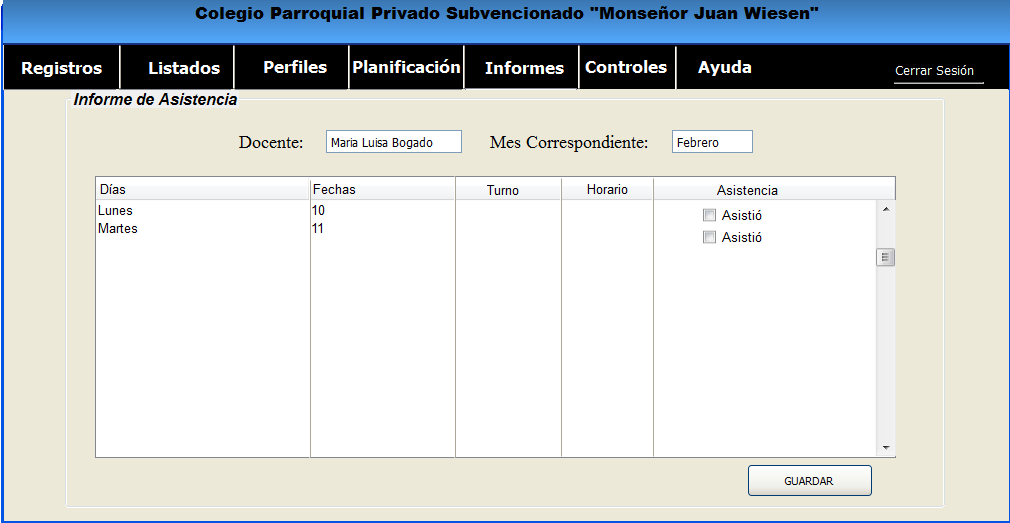
***Descripción:*** El docente al registrarse y seleccionar del menú Registros el menú ítem Calificación, deberá seleccionar en la ventana la disciplina y el curso del que desea cargar las calificaciones de los alumnos:

***Descripción:*** El docente obtiene la lista de alumnos del curso y de la materia seleccionada para el registro de las calificaciones obtenidas por éstos.

* **Interfaz de Planificación de Tareas o exámenes.**

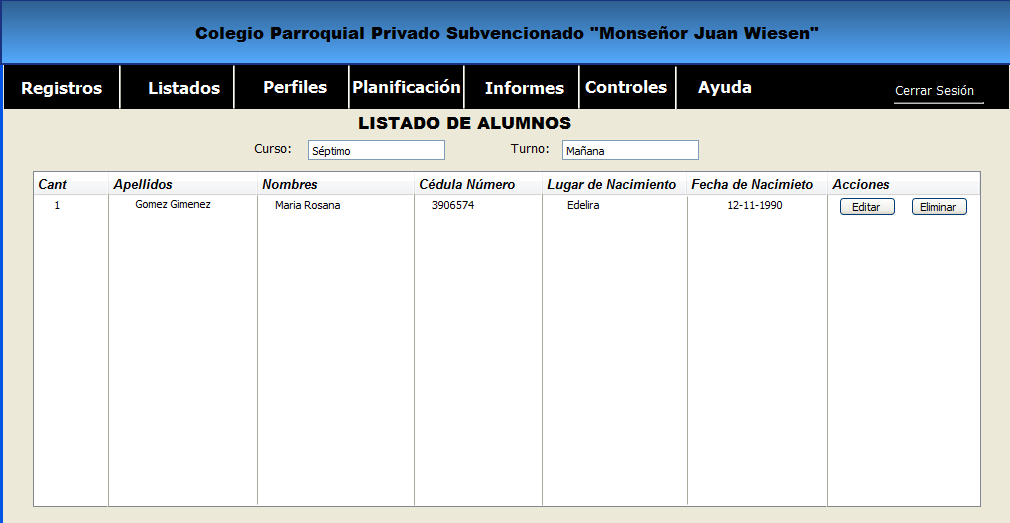
******

***Descripción:*** El docente planifica las tareas y/o exámenes a llevarse a cabo en cierta disciplina y curso.

* **Interfaz de Proveer Materiales**

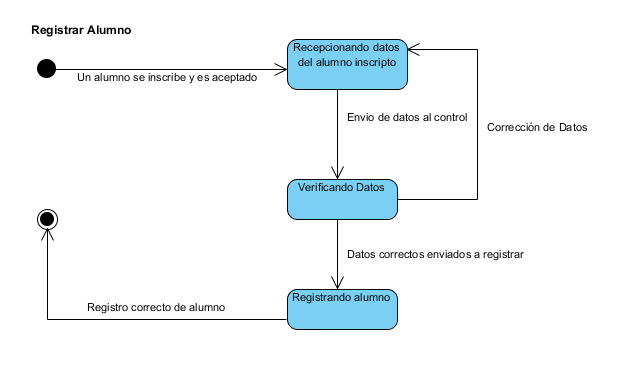
***Descripción:*** El docente selecciona la disciplina y el curso al que desea proveer un material de estudio utilizado durante el desarrollo de las clases.

* **Interfaz de Listado**

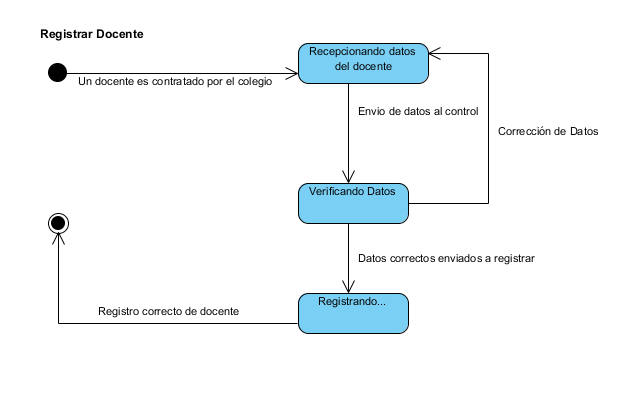


***Diagrama de Máquina de Estado***

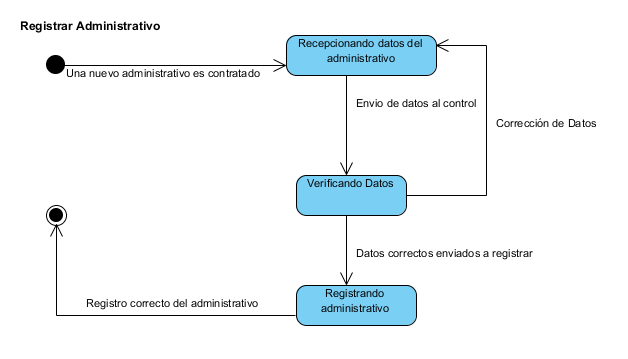
1. Registrar Alumnos
2. Registrar Docentes
3. Registrar Administrativos
4. Registrar Materias
5. Planificar tareas y exámenes
6. Registrar puntajes y calificaciones
7. Proveer materiales de estudio
8. **DME1 - Registrar Alumnos**



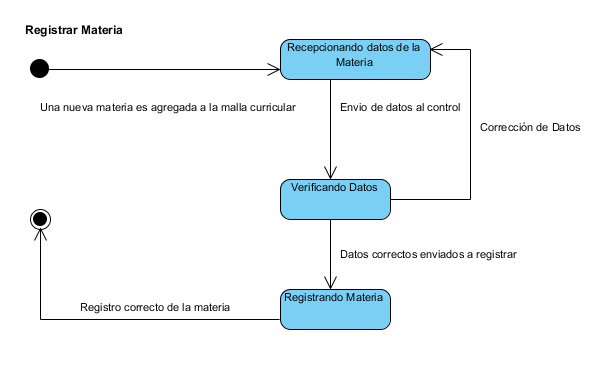
1. **DME2 - Registrar Docentes**

****

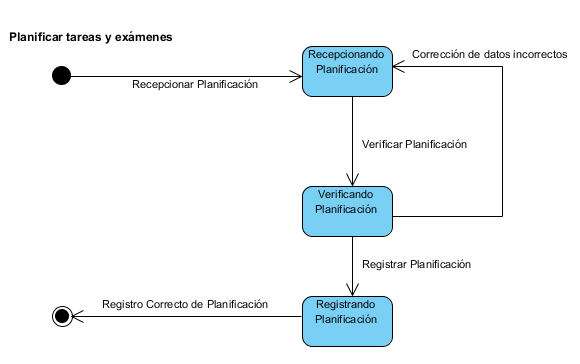
1. **DME2 – Registrar Administrativo**

****

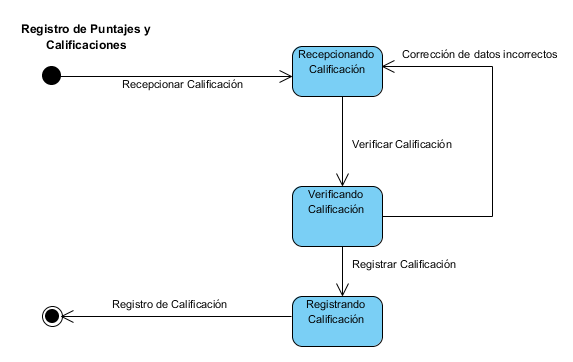
1. **DME3 - Registrar Materias**

****

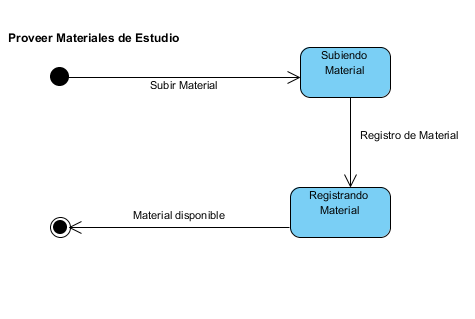
1. **DME4 – Planificar tareas y exámenes**

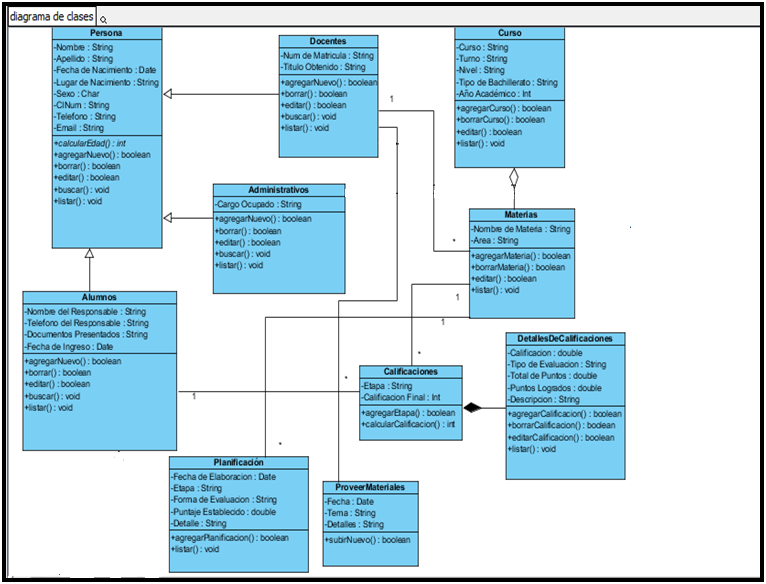
****

1. **DME5 – Registrar Puntajes y Calificaciones**

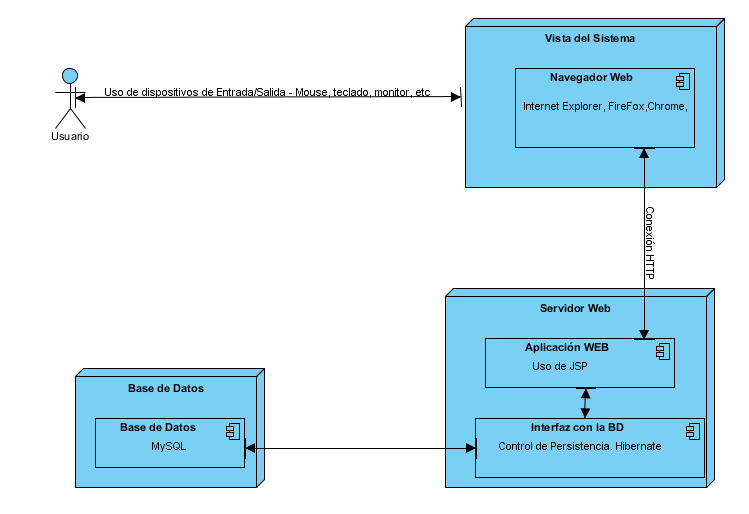


1. **DME7 – Proveer Materiales de Estudio**

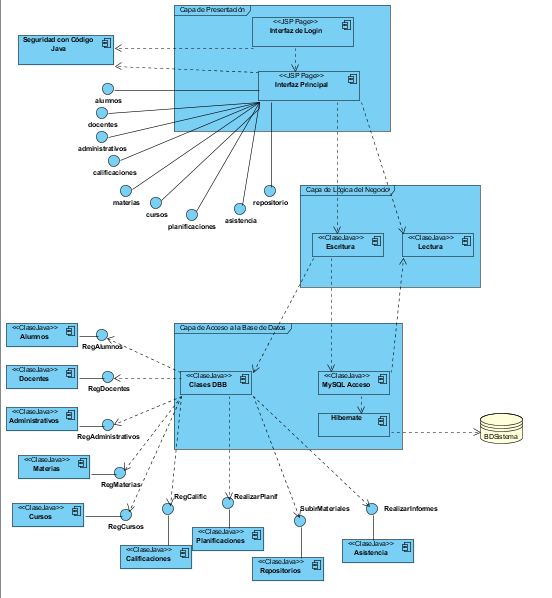
****

***Diagrama de Clase de Diseño***

***Diagrama de Despliegue de Arquitectura***

******

***Diagrama de Componentes***

******

***ESTIMACIONES***

* **Estimación de Costo y Tiempo: COCOMO II**

Para conocer y tener una mejor comprensión de este documento, es necesario especificar los componentes del módulo y la complejidad del mismo.

**Componentes de un Módulo**

1. **Datos:**

En esta categoría entran los archivos lógicos internos (ILF, por sus siglas en inglés) y los archivos de interfaz externa (EIF). El término *archivo* es empleado aquí para referirse a un conjunto de datos lógicamente relacionados e identificados así por los usuarios. Ejemplos: *Usuarios, Alumnos, Docentes, etc.*

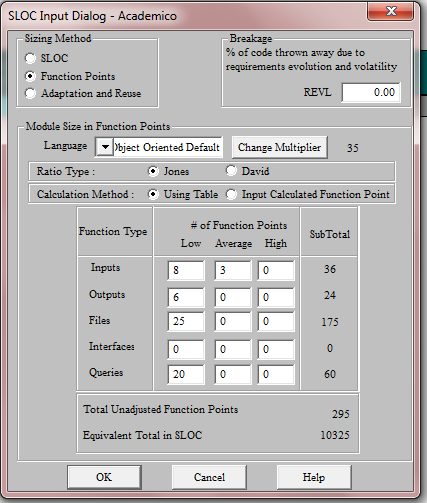
* ***Archivos Lógicos Internos:*** Un ILF es un grupo de datos identificado por el usuario y mantenido dentro de los límites de la aplicación. El propósito fundamental de un archivo lógico interno es almacenar datos que son mantenidos a través de uno o más *procesos elementales* pertenecientes al sistema en cuestión.
* ***Archivos de Interfaz Externa:*** Un EIF es un grupo de datos identificado por el usuario y referenciado por la aplicación, pero mantenido dentro de los límites de otro sistema. El propósito fundamental de un archivo de interfaz externa es almacenar datos mantenidos por otro sistema y referenciados por uno o más *procesos elementales* pertenecientes a la aplicación bajo estudio.

1. **Transacciones**

En esta categoría entra toda la funcionalidad de procesamiento de datos provista por el sistema, vista desde el ángulo de los usuarios. No debe confundirse el término con las *transacciones* provistas por el ambiente de implementación (transacciones sobre la base de datos, transacciones CICS, etc.).

Las transacciones pueden pertenecer a alguno de los siguientes tres tipos de procesos elementales: *entradas externas (EI), salidas externas (EO)* y *consultas externas (EQ)*.

* ***Entradas Externas:*** Una entrada externa es un proceso elemental que toma datos provenientes del medio ambiente del sistema. Su propósito fundamental es mantener uno o más archivos lógicos internos y/o alterar el comportamiento del sistema. Ejemplos: Alta de un Usuario como ser alumno, docentes o administrativos, etc.
* ***Salidas Externas:*** Una salida externa es un proceso elemental que envía datos fuera de los límites de la aplicación. Su propósito fundamental es presentar información al usuario a través de la recuperación y el procesamiento de datos. La lógica de procesamiento debe contener al menos un cálculo o fórmula matemática, o debe crear datos derivados. Una salida externa también puede mantener uno o más archivos lógicos internos. Ejemplos: Listado de nómina de alumnos, listado de calificaciones, listado de docentes, listado de materias por curso, etc.
* ***Consultas Externas:*** Una consulta externa es un proceso elemental que envía datos fuera de los límites de la aplicación. Su propósito fundamental es presentar información al usuario a través de la recuperación de datos. La lógica de procesamiento no contiene cálculos ni fórmulas matemáticas, ni tampoco crea datos derivados. No mantiene archivos lógicos internos.

**Tamaño en SLOC del Módulo del Sistema de acuerdo a la complejidad del mismo**

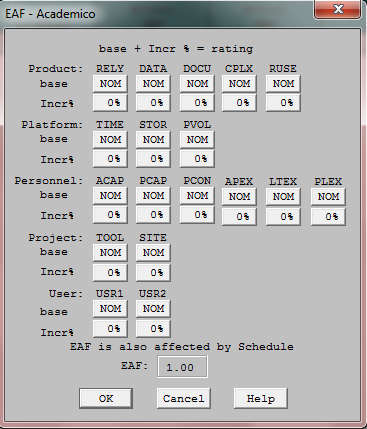
Observación: El lenguaje de programación seleccionado es Ruby on Rails

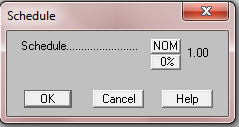
**Factores de Ajuste de Esfuerzo**

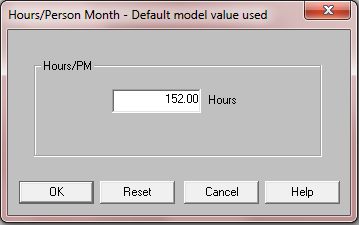
COCOMO II sugiere que muchos de los 17 multiplicadores de esfuerzo no pueden estimarse en el diseño temprano, por lo que los reduce a 7 multiplicadores:

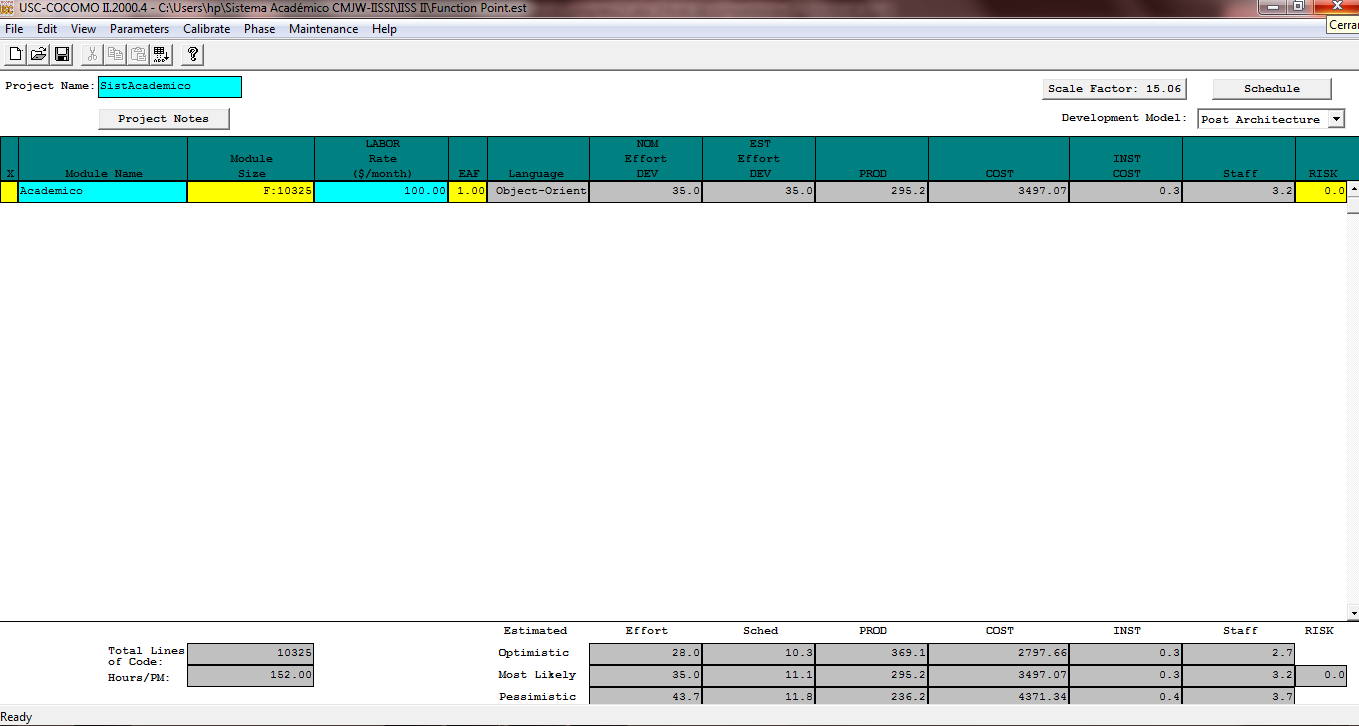
* **RCPX:** Confiabilidad y complejidad del producto. En cierto modo corresponde a una aproximación a lo que más tarde se desglosaría en RELY, DATA, CPLX, DOCU.
* **RUSE:** Nivel de reutilizabilidad del desarrollo. Corresponde exactamente a su homólogo en el modelo de diseño post arquitectura.
* **PDIF:** Dificultad de uso de la plataforma. Prefigura TIME, STOR y PVOL.
* **PERS:** Capacidad del personal de desarrollo. Prefigura ACAP, PCAP, PCON.
* **PREX:** Experiencia del personal de desarrollo. Prefigura APEX, PLEX, LTEX.
* **FCIL:** Facilidades de desarrollo. Prefigura TOOL y SITE.
* **SCED:** exigencias sobre el calendario. Corresponde exactamente a su homónimo en el modelo post-arquitectura.

A continuación exponemos los factores de ajuste correspondientes al módulo de nuestro sistema:



**Schedule**

**Horas – PM**

****

***ANÁLISIS DE RIESGO***

* **INVENTARIO COMPLETO**

Inventar los bienes que se han de proteger.

* **PLAN DE ACCION**

***Análisis y Gestión de Riesgos***

1. **Introducción**

Para comprender correctamente en qué consiste el análisis y gestión de riesgos de un proyecto, debemos conocer y entender claramente que es un riesgo.

Un riesgo se define como la combinación de la probabilidad de que se produzca un evento y sus consecuencias negativas. El término riesgo hace referencia a la proximidad o contingencia de un posible daño.

El riesgo implica siempre dos características principales que son:

* **La incertidumbre:** El acontecimiento que caracteriza al riesgo de que pueda o no ocurrir. No hay un riesgo de 100% de probabilidad, ya que un riesgo del 100% es una limitación del proyecto.
* **Pérdida:** Si el riesgo se llega a realizar, ocurrirán consecuencias no deseadas y pérdidas que alteren el desarrollo correcto del proyecto.

Suele ser común el confundir preocupaciones, riesgos y problemas: mientras que una **preocupación** es cualquier situación sobre la cual existen dudas en algún determinado contexto y que, por lo tanto, será evaluada como un posible riesgo, un **problema** es un riesgo que, efectivamente, se ha producido. (Observa la siguiente figura, Preocupación – Riesgo – Problema).

De aquí la importancia de realizar un correcto y adecuado “Análisis y Gestión de Riesgos”, ya que ayuda a prevenir fallas en la estimación del tiempo y costo del proyecto.

Mediante el análisis de riesgos podrán identificarse las posibles amenazas que acechan a los diferentes componentes del sistema, y de allí determinar la vulnerabilidad del mismo y la estimación del impacto y grado de perjuicio que podrá tener.

Y según los resultados obtenidos de la gestión de riesgos se llegarán a establecer las medidas para prevenir, impedir, reducir o controlar los posibles aspectos negativos o perjuicios que pueden llegar a ocasionar.

1. **Propósito del Plan**

El propósito del plan de Análisis y Gestión de Riesgos es determinar, analizar, valorar, clasificar los riesgos para posteriormente implementar mecanismos que permitan controlarlo según la prioridad de los mismos.

1. **Visión General**
   1. ***Objetivos***

El objetivo principal de dicho plan es asegurar de que el proyecto se desarrolle en el tiempo estimado y el costo calculado, detectando oportunamente todos los riesgos posibles y definiendo las medidas de control, supervisión y resolución a tomar ante las amenazas de que estos riesgos puedan ocurrir.

* 1. ***Prioridades de aversión al riesgo***

Se definen para el presente plan las siguientes prioridades de aversión al riesgo:

1. Volatilidad de requisitos.
2. Cumplimiento de requisitos.
3. Cumplimiento de la planificación temporal.
4. Problemas de personal.
5. Problemas tecnológicos.
   1. ***Organización***
      1. ***Gestión***

La gestión del presente plan de proyecto deberá ser llevada a cabo en forma directa por el jefe de proyecto junto con el grupo de desarrolladores.

* + 1. ***Responsabilidades***
* ***Del Jefe de Proyecto***
* Resolver los problemas que se puedan presentar.
* Establecer las metodologías y herramientas a utilizar.
* Gestionar el presente plan y demás documentaciones del proyecto.
* Supervisar y ordenar las tareas del equipo de desarrolladores.
* Evaluar los cambios realizados junto con el equipo de desarrolladores.
* Supervisar si el cumplimiento correcto de los requisitos del sistema.
* ***Del equipo de desarrolladores***
* Realizar las tareas asignadas por el jefe de proyecto para el avance y desarrollo del proyecto.
* Interactuar con los demás miembros para implementar un ambiente de trabajo cooperativo y armonioso.

1. **Análisis y Gestión de Riesgo**
   1. ***Identificación***

El proceso de identificación de riesgos se enfoca en detectar cuales son las fuentes principales de riesgos. Identificar cuales tienen mayor impacto sobre el éxito o no del proyecto y de mayor probabilidad de ocurrencia.

Los riesgos deben ser definidos en forma clara, concisa e inequívoca. Para ello se realiza una lista extensa y detallada de los mismos, los cuales abarcan todas las áreas del proyecto.

* + 1. ***Clasificación (o taxonomía) de los riesgos – Fuentes***

La clasificación de los riesgos -también denominadas taxonomías de riesgos- puede servir de ayuda para elaborar un enfoque coherente, reproducible y medible. Las listas de clasificación permiten al equipo pensar con mayor amplitud sobre los riesgos que pueden afectar al proyecto dado que se dispone de una lista de áreas del proyecto susceptibles de esconder riesgos.

Existen muchas taxonomías o clasificaciones para los riesgos de proyectos generales de desarrollo de software. Para el presente trabajo se ha escogido la clasificación propuesta por el Software Risk Management (SRM) desarrollado por el Software Engineering Institute.

A continuación se presenta la *Clasificación de los elementos de la taxonomía del software,* en el marco del presente proyecto se definen las clase, elementos y atributos.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Elemento** | **Riesgo** | **Fuente** |
| ***Planificación*** | | | |
| R-00 | Proceso de Planificación | Errores en la estimación de tiempo y costo del proyecto | Jefe de Proyecto |
| ***Requerimientos*** | | | |
| R-01 | Captura de requerimientos | Técnicas utilizadas poco eficaces | Jefe de Proyecto |
| R-02 | Requerimientos | Definición de nuevos requisitos no definidos al principio. | Clientes/Usuarios |
| R-03 | Requerimientos | Necesidad de modificar la definición de requisitos. | Clientes/Usuarios |
| R-04 | Requerimientos | Eliminar requisitos ya definidos. | Clientes/Usuarios |
| ***Diseño*** | | | |
| R-05 | Diseño | Incumplimiento del estándar de interfaz | Jefe de Proyecto / Desarrollador |
| ***Usuarios/Clientes*** | | | |
| R-06 | Usuarios/Clientes | Escasa información por parte del cliente para la elaboración del proyecto. | Jefe de Proyecto |
| ***Recurso Tecnológico*** | | | |
| R-07 | Tecnología | Dificultades por falta de experiencia con las herramientas utilizadas. | Jefe de Proyecto/Desarrolladores. |
| R-08 | Tecnología | Daños accidentales a la infraestructura de la tecnología de la información | Jefe de Proyecto/Desarrolladores |
| ***Recurso Humano*** | | | |
| R-09 | Equipo de Trabajo | Falta de entendimiento entre compañeros de trabajo | Jefe de Proyecto/Desarrolladores |
| R-10 | Equipo de Trabajo | Falta de experiencia de los desarrolladores en la metodología utilizada. | Jefe de Proyecto/Desarrolladores |
| R-11 | Equipo de Trabajo | Ausencia de personal. | Jefe de Proyecto/Desarrolladores |
| ***Desarrollo*** | | | |
| R-12 | Desarrollo del Proyecto | Poca documentación en el código fuente. | Jefe de Proyecto/Desarrolladores |
| R-13 | Desarrollo del Proyecto | Formato no establecido para las documentaciones necesarias del software desarrollado | Jefe de Proyecto/Desarrolladores |
| R-14 | Desarrollo del Proyecto | Falta de métodos específicos para el diseño de los casos de prueba. | Jefe de Proyecto/Desarrolladores |
| R-15 | Desarrollo del Proyecto | Falta de experiencia con herramientas que apoyen la planificación y el seguimiento de las actividades | Jefe de Proyecto/Desarrolladores |

* + 1. ***Declaración de los Riesgos***

Las definiciones genéricas de un riesgo no hacen desaparecer la incertidumbre y dan lugar a distintas interpretaciones del riesgo. Las definiciones que no dejan lugar a dudas permiten a los equipos:

* Asegurarse de que todos los miembros del equipo comprenden el riesgo de la misma forma. Es decir al jefe de proyecto y el equipo de desarrolladores.
* Comprender la causa o causas del riesgo y la relación con los problemas que puedan surgir.

En las declaraciones de riesgos se definen en forma más precisa los riesgos identificados, siguiendo un proceso de declaración en dos partes (condición – consecuencia). La primera parte de la declaración de riesgo se denomina **condición** y describe una situación o atributo del proyecto existente que el equipo prevé que puede resultar en una pérdida en el proyecto o en una reducción de beneficios. La segunda parte de la declaración de riesgo se denomina **consecuencia** y describe el atributo o situación no deseable del proyecto. Además se incluyen los **efectos** que tendrían estos riesgos de no controlarse debidamente.

A continuación se encuentra la descripción de la lista de riesgos:

* **R-00:** *Errores en la estimación de tiempo y costo del Proyecto.*
* *Condición:* Errores en los factores utilizados para la estimación de tiempo de desarrollo del proyecto, y por ende el costo del mismo. (métrica utilizada incorrectamente, etc.). O caso contrario la sobreestimación del proyecto.
* *Consecuencia:* A causa de la falla en la estimación, no se dispone de suficiente tiempo para la terminación del proyecto. Sobrecarga de horas a trabajar.
* *Efecto:* Baja calidad del producto debido e insatisfacción por parte del cliente/usuario.
* **R-01:** *Técnicas utilizadas poco eficaces.*
* *Condición:* Las técnicas utilizadas para la captura de información no fueron suficiente para capturar todos los datos posibles que brinda los clientes/usuarios. O caso contrario no se utilizó correctamente (entrevista, encuestas, observación, etc).
* *Consecuencia:* Las informaciones capturadas con las técnicas utilizadas no son suficiente y por lo tanto, los requisitos y el modelado del mismo no representa lo que en realidad desea el cliente/usuario.
* *Efecto:* Baja calidad del producto ya que los requisitos definidos no son los necesarios y correctos.
* **R-02:** *Definición de nuevos requisitos no definidos al principio.*
* *Condición:* Este riesgo es uno de las más probables a ocurrir, debido a la volatilidad de los requerimientos del usuario.
* *Consecuencia:* El proyecto requerirá más tiempo de desarrollo debido a que los nuevos requisitos afectarán el desarrollo del mismo.
* *Efecto:* El proyecto no se termina en el tiempo establecido, y la consistencia del software desarrollado es bajo, ya que surgieron nuevos requisitos que deben ser adheridos en el período de desarrollo.
* **R-03:** *Necesidad de modificar la definición de requisitos*
* *Condición:* La necesidad de modificar un requisito se produce debido a errores en la definición del mismo, lo cual puede ser por falta de experiencia del personal encargado en conocimiento de la metodología, de la herramienta utilizada, etc.
* *Consecuencia:* Definición de requisitos no válidos, pérdida de recursos y tiempo.
* *Efecto:* El efecto principal es la baja calidad del producto, ya que si esto no se detecta a tiempo, implicaría la modificación en lo que va del desarrollo del producto y podría traer consigo errores difíciles de controlar. Pérdida de tiempo.
* **R-04:** *Eliminar requisitos ya definidos*
* *Condición:* Eliminar un requisito ya definido y aprobado al principio puede suceder por varios motivos, entre ellos: a medida del desarrollo del producto el cliente ve que el mismo requisito ya no es necesario; otro de los motivos puede ser la falla en la captura de información por lo tanto su definición no es necesaria.
* *Consecuencia:* Pérdida de tiempo en el desarrollo del proyecto (en especial para los recursos humanos) ya que el avance logrado sobre el requisito será al final un desperdicio.
* *Efecto:* El proyecto no se finaliza en el tiempo establecido, ya que se ha trabajado sobre un requisito no primordial ni necesario, y afecta con el avance de otras tareas más importantes.
* **R-05:** *Incumplimiento del estándar de interfaz*
* *Condición:* La falta de experiencia y/o creatividad del personal con la herramienta utilizada para realizar dicha tarea.
* *Consecuencia:* Se encuentran errores en el diseño de las interfaces, por lo tanto se retrasa el tiempo de desarrollo del producto.
* *Efecto:* Producto de mala calidad, problemas de manejo del sistema por parte de los usuarios.
* **R-06:** *Escasa información por parte del cliente para la elaboración del proyecto*
* *Condición:* El personal encargado de contactar con el cliente/usuario no captura la información necesaria, ya que el cliente no responde mucho a las preguntas que se le realiza o no aporta lo suficiente para que la etapa de capturar los datos se lleve a cabo de la forma más optima.
* *Consecuencia:* Llevar a cabo un proyecto con requisitos no suficiente ni muy claros.
* *Efecto:* Insatisfacción por parte del cliente, ya que el producto no es lo que esperaba.
* **R-07:** *Dificultades por falta de experiencia con las herramientas utilizadas*
* *Condición:* Poco conocimiento por parte del personal en las herramientas utilizadas para el desarrollo del proyecto (motor de base de datos, lenguaje de programación, etc).
* *Consecuencia:* Se requiere un mayor tiempo para el desarrollo del proyecto, para poder aprender y/o profundizar los conocimientos sobre de las herramientas utilizadas.
* *Efecto:* Baja calidad del producto. Se corre el riesgo de que tenga errores no controlados.
* **R-08:** *Daños accidentales a la infraestructura de la tecnología de información*
* *Condición:* Se producen daños accidentales en las máquinas utilizadas para el desarrollo del proyecto.
* *Consecuencia:* Los daños se producen debido a una falla de máquina, falla de las herramientas utilizadas, caída o golpes accidentales a los elementos hardware, etc.
* *Efecto:* Pérdida de información del proyecto, retraso en el tiempo.
* **R-09:** *Falta de comunicación y entendimiento entre compañeros de trabajo*
* *Condición:* Ambos desarrolladores se encuentran en lugares físicos distintos a la hora de trabajar por el proyecto, solo se reúnen una vez a la semana.
* *Consecuencia:* Poca comunicación personal que puede acarrear falta de entendimiento en las decisiones tomadas para el desarrollo del proyecto.
* *Efecto:* Lentitud en el avance del proyecto, dificultades de buen relacionamiento, retraso en la fecha de entrega del producto.
* **R-10:** *Falta de experiencia de los desarrolladores en la metodología utilizada.*
* *Condición:* Los personales no cuentan con suficiente conocimiento de las metodologías utilizadas como PF (Function Poiny) y el COCOMO II, etc.
* *Consecuencia:* Retraso en las tareas a realizarse durante el desarrollo del proyecto.
* *Efecto:* Retraso en el tiempo de entrega del producto.
* **R-11:** *Ausencia de personal*
* *Condición:* Se puede correr el riesgo de ausencia de personal por varios días, semanas o incluso meses.
* *Consecuencia:* La ausencia puede deberse por alguna enfermedad u otros casos graves, y no se cuenta con otros personales capacitados para cumplir con el rol del ausente.
* *Efecto:* Retraso en el tiempo de entrega del producto. Baja calidad del producto.
* **R-12:** *Poca documentación en el código fuente*
* *Condición:* Se puede correr el riesgo de que exista poca documentación en todo el código fuente del proyecto debido a la falta de un estándar bien definido de programación.
* *Consecuencia:* Falta de comprensión del código fuente por parte de los demás desarrolladores, difícil detección de errores, etc.
* *Efecto:* La baja calidad del producto.
* **R-13:** *Formato no establecido para las documentaciones necesarias del software desarrollado*
* *Condición:* Se corre el riesgo de no preparar una documentación clara y consistente de todo el proyecto.
* *Consecuencia:* Falta de experiencia por parte de los desarrolladores y jefe de proyecto. Falta de comunicación.
* *Efecto:* La baja calidad del producto
* **R-14:** *Falta de métodos específicos para el diseño de los casos de prueba*
* *Condición:* No se definen métodos claros para diseñar los casos de prueba a utilizar durante todo el ciclo del desarrollo del software.
* *Consecuencia:* Falta de experiencia de los personales.
* *Efecto:* Producto inconsistente y baja calidad.
* **R-15:** *Falta de experiencia con herramientas que apoyen la planificación y el seguimiento de las actividades*
* *Condición:* Se cuentan con herramientas suficientes para todas las etapas del ciclo de vida del desarrollo del software pero la falta de experiencia de los desarrolladores no hace efectivo su uso.
* *Consecuencia:* Nuevas herramientas, falta de experiencia con el lenguaje de programación, etc.
* *Efecto:* Baja calidad del producto.
  1. ***Análisis y prioridad de los riesgos***

La meta principal del análisis de riesgos consiste en establecer las prioridades de los elementos de la lista de riesgos y determinar cuál de ellos justifica la reserva de recursos para el planeamiento. Por otro lado la asignación de prioridades a los riesgos permitirá tratar en primer lugar los riesgos más importantes del proyecto.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rango de probabilidad** | **Promedio para el calculo** | **Expresión de lenguaje natural** | **Valor numérico** |
| de 1% a 10% | 5 % | Baja | 1 |
| de 11 % a 25% | 18 % | Poco probable | 2 |
| de 26% a 55% | 40 % | Media | 3 |
| de 56% a 80% | 68 % | Altamente probable | 4 |
| de 81% a 99% | 90 % | Casi seguro | 5 |

* + 1. ***Estimación de la probabilidad***

La probabilidad del riesgo es una medida que calcula la probabilidad de que la situación descripta en el apartado de consecuencias de los riesgos de la declaración de riesgos llegue a producirse de verdad.

Para cuantificar la incertidumbre acerca de la ocurrencia de los riesgos se emplearán las categorizaciones expresadas en lenguaje natural, en base a un rango de probabilidades establecido en un cuadro de referencia, lo cual se observa en la siguiente tabla:

En la siguiente tabla se expresan los riesgos identificados para el proyecto con la probabilidad estimada para cada uno de ellos:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Riesgo** | **Expresión** | **Probabilidad** |
| R-00 | Errores en la estimación de tiempo y costo del proyecto | Poco probable | 18% |
| R-01 | Técnicas utilizadas poco eficaces | Poco probable | 18% |
| R-02 | Definición de nuevos requisitos no definidos al principio. | Poco probable | 18% |
| R-03 | Necesidad de modificar la definición de requisitos. | Medida | 40% |
| R-04 | Eliminar requisitos ya definidos. | Poco probable | 18% |
| R-05 | Incumplimiento del estándar de interfaz | Media | 40% |
| R-06 | Escasa información por parte del cliente para la elaboración del proyecto. | Media | 40% |
| R-07 | Dificultades por falta de experiencia con las herramientas utilizadas. | Media | 40% |
| R-08 | Daños accidentales a la infraestructura de la tecnología de la información | Baja | 5% |
| R-09 | Falta de entendimiento entre compañeros de trabajo | Poco probable | 18% |
| R-10 | Falta de experiencia de los desarrolladores en la metodología utilizada. | Media | 40% |
| R-11 | Ausencia de personal. | Baja | 5% |
| R-12 | Poca documentación en el código fuente. | Altamente probable | 68% |
| R-13 | Formato no establecido para las documentaciones necesarias del software desarrollado | Media | 40% |
| R-14 | Falta de métodos específicos para el diseño de los casos de prueba. | Media | 40% |
| R-15 | Falta de experiencia con herramientas que apoyen la planificación y el seguimiento de las actividades | Media | 40% |

* + 1. ***Estimación del impacto***

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Retraso en la planificación** | **Valor numérico** |
| Insignificante | 1 semana | 1 |
| Marginal | 2 semanas | 2 |
| Medio | 1 mes | 3 |
| Crítico | 2 meses | 4 |
| Catastrófico | Más de 2 meses | 5 |

El impacto que tiene un riesgo dentro del proyecto es lo que permite calcular los efectos que pueden producirse estos, los costos y las pérdidas que pueden generar si el riesgo llega a producirse.

Para el presente análisis se empleará la escala de medición subjetiva expresada en la siguiente tabla.

En la siguiente tabla se observa el impacto que tendrá en el proyecto cada uno de los riesgos definidos según la probabilidad de que ocurran:

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgo** | **Impacto** | **Riesgo** | **Impacto** |
| R-00 | Marginal | R-08 | Insignificante |
| R-01 | Marginal | R-09 | Marginal |
| R-02 | Marginal | R-10 | Medio |
| R-03 | Medio | R-11 | Insignificante |
| R-04 | Marginal | R-12 | Crítico |
| R-05 | Medio | R-13 | Medio |
| R-06 | Medio | R-14 | Medio |
| R-07 | Medio | R-15 | Medio |

* + 1. ***Exposición al riesgo***

La exposición al riesgo calcula la amenaza general que supone el riesgo combinando la información que expresa la probabilidad de una pérdida real con información que indica la magnitud de la pérdida potencial en un único valor numérico.

Para el presente estudio la exposición al riesgo se calcula multiplicando la probabilidad de riesgo por el impacto. Luego se utilizará la magnitud de la exposición al riesgo para clasificar los riesgos.

Magnitud de exposición al riesgo:

Aprox. 1 = bajo riesgo.

Aprox. 2 = riesgo medio.

Aprox. 3 = alto riesgo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Exposición** |
| R-00 | Errores en la estimación de tiempo y costo del proyecto | 18% | 2 | 0.36 |
| R-01 | Técnicas utilizadas poco eficaces | 18% | 2 | 0.36 |
| R-02 | Definición de nuevos requisitos no definidos al principio. | 18% | 2 | 0.36 |
| R-03 | Necesidad de modificar la definición de requisitos. | 40% | 3 | **1.20** |
| R-04 | Eliminar requisitos ya definidos. | 18% | 2 | 0.36 |
| R-05 | Incumplimiento del estándar de interfaz | 40% | 3 | 1.20 |
| R-06 | Escasa información por parte del cliente para la elaboración del proyecto. | 40% | 3 | 1.20 |
| R-07 | Dificultades por falta de experiencia con las herramientas utilizadas. | 40% | 3 | **1.20** |
| R-08 | Daños accidentales a la infraestructura de la tecnología de la información | 5% | 1 | 0.05 |
| R-09 | Falta de entendimiento entre compañeros de trabajo | 18% | 2 | 0.36 |
| R-10 | Falta de experiencia de los desarrolladores en la metodología utilizada. | 40% | 3 | 1.20 |
| R-11 | Ausencia de personal. | 5% | 1 | 0.05 |
| R-12 | Poca documentación en el código fuente. | 68% | 4 | **2.72** |
| R-13 | Formato no establecido para las documentaciones necesarias del software desarrollado | 40% | 3 | 1.20 |
| R-14 | Falta de métodos específicos para el diseño de los casos de prueba. | 40% | 3 | **1.20** |
| R-15 | Falta de experiencia con herramientas que apoyen la planificación y el seguimiento de las actividades | 40% | 3 | 1.20 |

* 1. ***Gestión de los Riesgos***

La lista de riesgos presentada anteriormente servirá como punto de partida al proceso de administración de riesgos. Cabe mencionar que debido a que la gestión de riesgos es una actividad continua, esta lista es susceptible de presentar modificaciones a medida que avance el proyecto, y como consecuencia de ello deberá ser actualizada a lo largo de todo el ciclo de análisis, planeamiento y supervisión de los riesgos.

El análisis de riesgos presentado en el punto 4.2 examina la amenaza que representa cada riesgo como una ayuda para decidir en qué riesgos es conveniente aplicar una acción.

* + 1. ***Líneas de Acción***

Para ejercer una adecuada gestión y supervisión de los riesgos mencionados anteriormente, se elaborará un Plan de Acción y Un Plan de Contingencias para cada unos de ellos.

El **Plan de Acción** será utilizado para minimizar los riesgos mediante acciones preventivas. La probabilidad que un riesgo ocurra así como el impacto que el mismo podría ocasionar en el proyecto pueden ser mitigados encarando los problemas en forma proactiva.

El **Plan de Contingencia**, por el contrario intenta implementar respuestas rápidas para mitigar los efectos en el caso que los riesgos se concreten, es decir reducir el impacto de los mismos mediante una reacción planeada. Este plan, además definirá ciertos indicadores que permitirán poner en marcha las acciones previstas, es decir, en caso que se verifiquen ciertos disparadores se adoptarán las medidas indicadas.

1. **Riesgo R-03: Necesidad de modificar la definición de requisitos.**
   1. ***Aspectos a considerar***
      1. ***Porque*** el riesgo es importante: Ya que podría ocurrir que los riesgos tienen errores por qué no fueron interpretados correctamente o las herramientas utilizadas no fueron suficientemente eficiente para la captura de los mismos.
      2. ***Que información*** se necesita para seguir el estado del riesgo:

* Documentación de las técnicas de requisitos utilizados.
* Documentación de la definición de requisitos del sistema.
  + 1. ***Quien es responsable*** de realizar las actividades de control de riesgo: El responsable es el jefe del proyecto y los encargados de realizar la captura de requisitos y la definición de los mismos.
    2. ***Que recursos*** se necesitan para realizar las actividades de control de riesgo: Los recursos que se necesitan son los tecnológicos, las herramientas utilizadas, y los humanos que son los futuros usuarios del sistema.
  1. ***Plan de Acción***
     1. Revisar y analizar detalladamente los requisitos definidos al principio para ver los puntos principales que son necesarios modificar.
     2. Contactar y conversar con el cliente y los futuros usuarios para realizar las consultas y entrevistas que sean necesarias para capturar de manera eficiente todas sus necesidades en cuanto al sistema.
     3. Capturar toda la información que sea necesaria y que ayude a una mejor definición y comprensión de los requisitos.
     4. Analizar la información capturada con el jefe de proyecto y desarrolladores para una correcta definición del requisito.
     5. Comunicar de los cambios efectuados en los requisitos a los desarrolladores.
     6. Construir los prototipos necesarios para los usuarios, clientes y desarrolladores para que conozcan todos los requisitos del producto.
     7. Realizar un control para verificar que el desarrollo del producto cumple con todos los requisitos definidos nuevamente con los cambios ya efectuados.
  2. ***Plan de Contingencia***
     1. ***Disparador:*** Error en la interpretación de los requisitos definidos.
* Consultar con quienes no comprenden correctamente los requisitos para aclarar las dudas de los mismos.
* El jefe de proyecto debe comunicar directamente a los desarrolladores del productos de los cambios efectuados en la definición de requisitos para que reconozcan donde existe la diferencia con el anterior.

1. **Riesgo R-07: Dificultades por falta de experiencia con las herramientas utilizadas.**
   1. ***Aspectos a considerar***
      1. ***Porque*** el riesgo es importante: Las herramientas utilizadas como el motor de base de datos, herramientas para la documentación y principalmente el lenguaje de programación la falta de experiencia con estos son muy importante porque podrían comprometer la planificación de tiempo y costo del proyecto y también la calidad en si del software.
      2. ***Que información*** se necesita para seguir el estado del riesgo:

* Documentación del software para controlar la calidad del mismo y ver la correcta utilización de la herramienta.
* Informes diarios de los avances del proyecto durante toda la etapa de captura, diseño y desarrollo.
  + 1. ***Quien es responsable*** de realizar las actividades de control de riesgo: El responsable es el jefe del proyecto.
    2. ***Que recursos*** se necesitan para realizar las actividades de control de riesgo: No se necesita de recursos económicos pero si de humanos como ser el personal encargado de realizar el control y el desarrollo de las actividades del proyecto en general.
  1. ***Plan de Acción***
     1. Realizar un control a todo el proceso de desarrollo del software desde la documentación hasta lo principal que es el desarrollo del proyecto.
     2. Controlar la efectividad y productividad del trabajo de los desarrolladores.
     3. Identificar a los desarrolladores que cuentan con algún problema con las herramientas que son utilizadas.
     4. Conversar con estos y buscar la manera de ayudarlos con las herramientas.
     5. Buscar las técnicas que los ayude a conocer más sobre las herramientas que están siendo utilizadas, como ser: charlas, cursos rápidos, etc.
     6. Caso contrario, asignar al personal tareas donde tenga más capacidad y experiencia. Asignar tareas con respecto a las capacidades de los personales.
  2. ***Plan de Contingencia***
     1. ***Disparador:*** Atraso en el tiempo de la fecha de entrega del proyecto.
* El encargado de controlar el progreso de las actividades de los equipos de trabajo en caso de encontrar algún atraso informar al jefe de proyecto el motivo de esto.
* Capacitar a los personales o asignar tareas según la capacidad de cada uno, o buscar profesionales capacitados para cada área.

1. **Riesgo R-12: Poca documentación en el código fuente.**
   1. ***Aspectos a considerar***
      1. ***Porque*** el riesgo es importante: Es un riesgo importante porque la poca documentación en el código podría provocar un atraso del tiempo durante el desarrollo, ya que complica interpretar y entender el código del proyecto a los demás miembros del equipo; así también se desarrolla un producto de baja calidad.
      2. ***Que información*** se necesita para seguir el estado del riesgo:

* Control del código fuente por parte del jefe de proyecto para evaluar la calidad de trabajo de cada uno de los miembros del equipo.
* Informe diario de los avances diarios del proyecto.
* Documentación de donde se definen las reglas para la documentación del código fuente.
  + 1. ***Quien es responsable*** de realizar las actividades de control de riesgo: El responsable es el jefe del proyecto.
    2. ***Que recursos*** se necesitan para realizar las actividades de control de riesgo: No se necesita de recursos económicos pero si de humanos como ser el personal encargado de realizar el control y el desarrollo de las actividades del proyecto en general.
  1. ***Plan de Acción***
     1. Revisar la documentación donde se definió las reglas para la documentación del código fuente y los estándares de programación.
     2. Identificar las diferencias y carencias en las documentaciones del código fuente.
     3. Corregir la documentación basándose en las reglas ya definidas al principio.
     4. Aclarar las reglas para la documentación con todos los miembros del equipo de desarrolladores y demás miembros de trabajo.
     5. Modificar las documentaciones ya realizadas.
     6. Controlar nuevamente las modificaciones ya realizadas en la documentación del proyecto.
  2. ***Plan de Contingencia***
     1. ***Disparador:*** Dificultades para leer, comprender y detectar errores en el código fuente.
* El jefe de proyecto debe revisar urgentemente la documentación de los estándares de programación y mostrar al grupo de desarrolladores para la mejor interpretación del mismo. Aclarar todas las dudas para poder mejorar y corregir los errores.
* Corregir y mejorar la documentación del código fuente el jefe de proyecto con los desarrolladores basándose en lo que se definió en los estándares de programación.

1. **Riesgo R-12: Falta de métodos específicos para el diseño de los casos de prueba.**
   1. ***Aspectos a considerar***
      1. ***Porque*** el riesgo es importante: Es un riesgo importante porque su buen diseño e implementación durante todo el desarrollo del producto ayuda a detectar errores a tiempo y permite corregirlos. Pero la falta del mismo podría atrasar el tiempo de desarrollo por no detectar a tiempo los errores, o en caso de que esto no ocurra podría desarrollarse un producto de baja calidad.
      2. ***Que información*** se necesita para seguir el estado del riesgo:

* Documentación de los estándares de programación.
* Documentación de los diseños de caso de prueba.
  + 1. ***Quien es responsable*** de realizar las actividades de control de riesgo: El responsable es el jefe del proyecto y el equipo de desarrolladores.
    2. ***Que recursos*** se necesitan para realizar las actividades de control de riesgo: No se necesita de recursos económicos pero si de humanos como ser el personal encargado de realizar el control y el desarrollo de las actividades del proyecto en general.
  1. ***Plan de Acción***
     1. Controlar el código fuente y ver si se han definido los diseños de casos de prueba.
     2. Si se han definido los casos de prueba, probarlos ejecutando el código fuente del proyecto.
     3. En caso de identificar errores y los diseños no son muy específicos, informar de esto al jefe de proyecto.
     4. El jefe de proyecto se reúne con el grupo de desarrolladores para establecer la manera de definir los métodos correctos y específicos de los casos de prueba.
     5. Se definen correctamente los casos de prueba.
     6. Se controla nuevamente el código fuente mediante los casos de prueba.
  2. ***Plan de Contingencia***
     1. ***Disparador:*** Errores difíciles de controlar por la falta de métodos específicos para el diseño de los casos de prueba.
* El jefe de proyecto define el diseño para la definición de los casos de prueba junto con los propios desarrolladores quienes son los que conocen mejor los posibles errores que pueden generarse a medida que va avanzando el desarrollo del proyecto.
* Realizar pruebas mediante la compilación y/o depuración del proyecto para identificar los errores sin diseñar aún los casos de pruebas.

1. **Riesgo R-12: Errores y excepciones no controladas**
   1. ***Aspectos a considerar***
      1. ***Porque*** el riesgo es importante: Es un riesgo importante porque los errores y las excepciones no controladas pueden llegar a alterar el funcionamiento correcto del sistema, además sino se llegan a identificar a tiempo puede alterar la fecha de entrega del sistema y por ende la calidad del producto es bajo.
      2. ***Que información*** se necesita para seguir el estado del riesgo:

* Documentación del código fuente del software.
* Documentación de los errores encontrados durante el desarrollo del software.
  + 1. ***Quien es responsable*** de realizar las actividades de control de riesgo: El responsable es el jefe del proyecto y el equipo de desarrolladores.
    2. ***Que recursos*** se necesitan para realizar las actividades de control de riesgo: No se necesita de recursos económicos pero si de humanos como ser el personal encargado de realizar el control y el desarrollo de las actividades del proyecto en general, también la documentación mencionada anteriormente.
  1. ***Plan de Acción***
     1. Verificar el funcionamiento correcto del sistema utilizando casos de prueba.
     2. Depurar y compilar todo el código fuente del software.
     3. Identificar los errores lanzados por el sistema al igual que las excepciones no controladas.
     4. Identificar el origen del error y/o la excepción y buscar el origen del mismo.
     5. Informar acerca de los errores encontrados y no controlados al jefe de proyecto y demás miembros del grupo de desarrolladores.
     6. Buscar la solución del mismo, modificando código fuente o buscando demás soluciones.
     7. Una vez modificado el código fuente, utilizar un caso de prueba para verificar si el error fue controlado.
  2. ***Plan de Contingencia***
     1. ***Disparador:*** Atraso en la fecha de entrega del sistema y baja calidad del producto.
* Se informará al jefe de proyecto quien se encargará de llevar a cabo el control del caso con la ayuda del equipo desarrollador.
* Se realizará pruebas del sistema depurando todo el código fuente del proyecto.
* Se analizará la magnitud del error y se verá si es conveniente modificar el código fuente para poder controlarlo.

***Gestión de Calidad***

**CALIDAD DEL SOFTWARE**

1. **INTRODUCCION**

Se conoce como SQA (Software Quality Assurance) o GCS (Gestión de la Calidad del Software). Es una actividad que se aplica a lo largo de todo el ciclo de vida.

Según la IEEE, la Gestión de la Calidad del Software es el grado con el que un sistema, componente o proceso cumple con los requisitos especificados, las necesidades o expectativas del cliente o usuario.

Se debe conocer dos conceptos muy importantes que son:

* **Verificación:** Comprobar si los productos construidos en una fase del ciclo de vida satisfacen los requisitos establecidos en la fase anterior.
* **Validación:** Comprobar si el software construido satisface los requisitos del usuario.

Ahora es importante conocer que es la calidad del software, pero antes se debe diferenciar entre la calidad del *producto* y la calidad del *proceso* de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

Veamos las definiciones de calidad según algunos estándares internacionales:

*“La calidad es la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas”* (ISO 8402).

*“Capacidad del producto software para satisfacer los requisitos establecidos”* (DoD 2168).

A partir de estas definiciones podemos decir que la calidad es algo relativo. Siempre va depender de los requisitos o necesidades que se desee satisfacer. Por eso, la evaluación de la calidad de un producto siempre va a implicar una comparación entre unos requisitos preestablecidos y el producto realmente desarrollado.

Pero las definiciones que se han dado de la calidad son demasiado generales e imprecisas como para que resulten de utilidad a la hora de construir un software de calidad. Por eso surge el concepto de Modelo de Calidad, que nos ayuda a definir la calidad del software de una forma más precisa y útil.

1. **MODELOS DE CALIDAD DEL SOFTWARE**

En los modelos de calidad, la calidad se define de forma jerárquica. Resuelven la complejidad mediante la descomposición. La calidad es un concepto que se deriva de un conjunto de sub-conceptos.

Uno de los modelos de calidad más antiguos y extendidos es el de McCall (McCall, 1977), y de él han derivado otros modelos, como el de Boehm (Boehm, 78) o el SQM (Murine, 1988).

En los modelos de calidad, la calidad se define de forma jerárquica. Tienen una estructura, por lo general, en tres niveles:

* En el nivel más alto de la jerarquía se encuentran los FACTORES de calidad, que representan la calidad desde el punto de vista del usuario. Son las características que componen la calidad.
* Cada uno de los factores se descompone en un conjunto de CRITERIOS de calidad. Son atributos que, cuando están presentes, contribuyen al aspecto de la calidad que el factor asociado representa. Se trata de una visión de la calidad desde el punto de vista del producto software.
* Para cada uno de los criterios de calidad se definen entonces un conjunto de METRICAS, que son medidas cuantitativas de ciertas características del producto que, cuando están presentes, dan una indicación del grado en que dicho producto posee un determinado atributo de calidad.

La **ventaja** de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar.

Por lo tanto para medir la calidad de nuestro producto software vamos a utilizar uno de los modelos de calidad, este caso es el modelo de McCall.

* 1. **EL MODELO DE McCall**

El modelo de McCall organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad del producto:

* Operación del Producto.
* Revisión del Producto.
* Transición del Producto.

El modelo de McCall se basa en 11 factores de calidad, que se organizan en torno a los tres ejes mencionados de la siguiente manera:

**REVISION TRANSISION**

-Facilidad de mantenimiento -Portabilidad

-Flexibilidad -Reusabilidad

-Facilidad de prueba -Interoperabilidad

**OPERACIÓN**

-Corrección

-Fiabilidad

-Usabilidad

-Integridad

-Eficiencia

Estos factores de McCall se definen de la siguiente manera:

**1. Corrección:** Hasta qué punto un programa cumple sus especificaciones y satisface los objetivos del usuario. Por ejemplo, si un programa debe ser capaz de sumar dos números y en lugar de sumar los multiplica, es un programa incorrecto. Es quizás el factor más importante, aunque puede no servir de nada sin los demás factores.

**2. Fiabilidad:** Hasta qué punto se puede confiar en el funcionamiento sin errores del programa. Por ejemplo, si el programa anterior suma dos números, pero en un 25% de los casos el resultado que da no es correcto, es poco fiable.

**3. Eficiencia:** Cantidad de código y de recursos informáticos (CPU, memoria) que precisa un programa para desempeñar su función. Un programa que suma dos números y necesita 2 MB de memoria para funcionar, o que tarda 2 horas en dar una respuesta, es poco eficiente.

**4. Integridad:** Hasta qué punto se controlan los accesos ilegales a programas o datos. Un programa que permite el acceso de personas no autorizadas a ciertos datos es poco íntegro.

**5. Usabilidad:** El coste y esfuerzo de aprender a manejar un producto, preparar la entrada de datos e interpretar la salida del mismo.

**6. Facilidad de mantenimiento**: El coste de localizar y corregir defectos en un programa que aparecen durante su funcionamiento.

**7. Facilidad de prueba:** El coste de probar un programa para comprobar que satisface sus requisitos. Por ejemplo, si un programa requiere desarrollar una simulación completa de un sistema para poder probar que funciona bien, es un programa difícil de probar.

**8. Flexibilidad:** El coste de modificación del producto cuando cambian sus especificaciones.

**9. Portabilidad (o Transportabilidad):** El coste de transportar o migrar un producto de una configuración hardware o entorno operativo a otro.

**10. Reusabilidad:** Hasta qué punto se puede transferir un módulo o programa del presente sistema a otra aplicación, y con qué esfuerzo.

**11. Interoperabilidad:** El coste y esfuerzo necesario para hacer que el software pueda operar conjuntamente con otros sistemas o aplicaciones software externos.

Cada uno de estos factores, se dividen en un conjunto de atributos de calidad los cuales se puede observar en la siguiente tabla:

|  |  |
| --- | --- |
| **FACTOR** | **CRITERIOS** |
| **Corrección** | - Completitud  - Consistencia  - Trazabilidad |
| **Fiabilidad** | - Precisión  - Consistencia  - Tolerancia a fallos  - Modularidad  - Simplicidad |
| **Eficiencia** | - Eficiencia en ejecución  - Eficiencia en almacenamiento |
| **Integridad** | - Control de accesos  - Facilidad de auditoría |
| **Usabilidad** | - Facilidad de operación  - Facilidad de comunicación  - Facilidad de aprendizaje |
| **Facilidad**  **de Mantenimiento** | - Modularidad  - Simplicidad  - Consistencia  - Concisión  - Auto descripción |
| **Facilidad de Prueba** | - Modularidad  - Simplicidad  - Auto descripción  - Instrumentación |
| **Flexibilidad** | - Auto descripción  - Capacidad de expansión  - Generalidad  - Modularidad |
| **Portabilidad** | - Auto descripción  - Modularidad  - Independencia entre sistema y software  - Independencia del hardware |
| **Reusabilidad** | - Auto descripción  - Generalidad  - Modularidad  - Independencia entre sistema y software  - Independencia del hardware |
| **Interoperabilidad** | - Modularidad  - Compatibilidad de comunicaciones  - Compatibilidad de datos |

De estos 11 factores de McCall, seleccionamos 4 para aplicar a nuestro producto software y le damos un orden de importancia según se observa a continuación, además de las métricas aplicadas para cada uno de los factores de calidad:

|  |  |
| --- | --- |
| FACTOR | METRICAS APLICADAS |
| **Fiabilidad** | Numero de problemas que han aparecido en un mes de uso. |
| Numero de errores en la documentación. |
| Numero de errores en el programa. |
| Porcentaje de usuarios con problemas |
| **Corrección** | El diseño concuerda con los requisitos |
| El código concuerda con el diseño |
| Numero de cambios solicitados por el usuario. |
| Todas las funciones definidas son utilizadas. |
| **Integridad** | Permisos de usuarios para cada ABM |
| Nivel de Control de MySQL (malo, bueno, muy bueno). |
| **Usabilidad** | Tiempo de aprendizaje del sistema por parte de los usuarios. |
| Nivel de manejo del sistema por parte de los usuarios (malo, bueno, muy bueno). |
| Porcentaje de usuarios que realizaron reclamos por fallas del sistema. |

1. **ACTIVIDADES PARA EL CONTROL DE CALIDAD.**

Se pueden clasificar las actividades de control de calidad en dos categorías: análisis estáticos y análisis dinámicos. Los primeros analizan el objeto sin necesidad de ejecutarlo mientras que los segundos requieren la ejecución del objeto que está siendo probado.

* 1. **ANALISIS ESTATICOS**

Para el análisis de nuestro producto software “Sistema Académico CMJW”, vamos a utilizar el análisis estático manual disciplinado, que son dos: la auditoría y las revisiones. En este caso vamos a utilizar las revisiones, el cual se puede ver a continuación:

* + 1. **REVISIONES**

Se puede definir una revisión como una reunión formal en la que se presenta el estado actual de los resultados de un proyecto a un usuario, cliente u otro tipo de persona interesada, y se realiza un análisis estructurado de los mismos.

Uno de los objetivos fundamentales de las revisiones técnicas es ofrecer a los gestores información fiable acerca de los aspectos técnicos del proceso de desarrollo de software, de la misma forma que les llega información fiable acerca de los costes y la programación del trabajo, para que con esta información puedan tomar decisiones adecuadas para dirigir con éxito el proyecto.

Los tipos de revisiones a llevarse a cabo durante el desarrollo del proyecto son los siguientes:

* **Revisión de la documentación:** Esta revisión es fundamental, ya que es la base del proyecto, se realizará el control del mismo para ver si está completa y actualizada.
* **Revisión del Diseño:** Para tener un control de que los requisitos estén correctos y actualizados y se reflejen los mismos en las interfaces, y el modelo de datos.
* **Revisión del Código:** Para controlar si se cumplen los estándares de programación definidos.
* **Revisión de las Pruebas:** Para tener un control si son eficaces la aplicación de los mismos y si se está utilizando correctamente.
  1. **FORMULARIO DE REVISIONES**

Las revisiones realizadas estarán acompañadas por un formulario a rellenar, dicha revisión se realizará una vez a la semana a cargo de la responsable Mirian Saucedo, las cuales consistirán en:

* Para la revisión se tendrá como base las preguntas de comprobación correspondiente a cada formulario.
* Las tareas a ser realizadas por cada miembro del equipo serán asignadas por medio del GitHub.
* Las revisiones serán realizadas a cada ABM.
* Las revisiones deberán responder a tres preguntas fundamentales que son:
* ¿Quién revisó?
* ¿Qué fue revisado?
* ¿Qué inconvenientes se encontró y cuáles son las posibles soluciones, en caso de que no hubieron inconvenientes a que conclusiones se llegó?
* Al final de las revisiones, se debe preparar un informe general de la revisión técnica formal.
* Y por último se deben registrar los pedidos de cambios o errores al TRAC para que se busque su inmediata solución.
  + 1. **REVISION DE LA DOCUMENTACION**

|  |
| --- |
| **Tipo de Revisión:** De Documentación |
| **Fecha de Revisión: Hora de Revisión:** |
| **Encargado de Revisión:** |
| **Elemento de Configuración:** |
| **Preguntas de Comprobación:**   1. ¿La documentación se encuentra actualizada? 2. ¿Mantiene el mismo formato que se ha establecido en los estándares de la documentación? 3. ¿La redacción de la documentación es clara, correcta y precisa? 4. ¿La documentación general del proyecto, cuenta con todas las documentaciones generadas por el equipo hasta el momento? |
| **Resultado Obtenido:** |
| **Observaciones:** |

* + 1. **REVISION DEL DISEÑO**

|  |
| --- |
| **Tipo de Revisión:** Diseño de las Interfaces |
| **Fecha de Revisión: Hora de Revisión:** |
| **Encargado de Revisión:** |
| **Elemento de Configuración:** |
| **Preguntas de Comprobación:**   1. ¿Hay uniformidad en el diseño? 2. ¿Cubre el diseño todos los requisitos específicos definidos? 3. ¿Cumple el diseño con los estándares establecidos en el documento de Reglas y Formatos de diseño? 4. ¿Resulta fácil interactuar con la interfaz? 5. ¿El diseño es lo suficientemente detallado para que se pueda pasar a la fase de programación? 6. ¿El elemento se encuentra en revisiones actualizadas? |
| **Resultado Obtenido:** |
| **Observaciones:** |

|  |
| --- |
| **Tipo de Revisión:** Diseño de la Base de Datos |
| **Fecha de Revisión: Hora de Revisión:** |
| **Encargado de Revisión:** |
| **Elemento de Configuración:** |
| **Preguntas de Comprobación:**   1. ¿Las tablas definidas cumplen con la definición de requisitos? 2. ¿Para el diseño se utilizan las herramientas establecidas al principio? 3. ¿Cumple el diseño de las tablas con los estándares establecidos? 4. ¿El elemento se encuentra en revisiones actualizadas? |
| **Resultado Obtenido:** |
| **Observaciones:** |

* + 1. **REVISION DEL CODIGO**

|  |
| --- |
| **Tipo de Revisión:** Código Fuente y ABM |
| **Fecha de Revisión: Hora de Revisión:** |
| **Encargado de Revisión:** |
| **Elemento de Configuración:** |
| **Preguntas de Comprobación:**   1. ¿Se cumplen con los estándares de programación establecidos? 2. ¿Los ABM en revisión se comportan de acuerdo con cada uno de los casos de uso de requisitos que han sido especificado en la documentación? 3. ¿Existe alguna necesidad de modificar la funcionalidad del ABM? 4. ¿Existe la necesidad de definir nuevas funcionalidades para los ABM principales? 5. ¿El elemento se encuentra en revisiones actualizadas? |
| **Resultado Obtenido:** |
| **Observaciones:** |

* + 1. **REVISION DE LAS PRUEBAS**

|  |
| --- |
| **Tipo de Revisión:** Código Fuente y ABM |
| **Fecha de Revisión: Hora de Revisión:** |
| **Encargado de Revisión:** |
| **Elemento de Configuración:** |
| **Preguntas de Comprobación:**   1. ¿Cumple con el estándar establecido? 2. ¿El encargado del testeo del proyecto rellena correctamente los formularios del plan de pruebas? 3. ¿El elemento se encuentra en revisiones actualizadas? |
| **Resultado Obtenido:** |
| **Observaciones:** |

* 1. **ANALISIS DINAMICOS**

Para el análisis de nuestro producto software “Sistema Académico CMJW”, también vamos a utilizar el análisis dinámico, y principalmente el método de la **caja negra**.

* + 1. **METODO DE LA CAJA NEGRA**

Para el proyecto del Sistema Académico CMJW solo vamos a utilizar el método de la caja negra por el corto tiempo de desarrollo que se tiene y por ser más rápida su implementación en comparación con el método de caja blanca.

En este tipo de método, el objeto que se desea probar se ve como una caja negra. Esto quiere decir que la elección de los casos de prueba no se va a basar en el conocimiento que se tenga acerca de la estructura del objeto, sino en el conocimiento acerca de la funcionalidad deseada (descripción funcional).

A la prueba de caja negra también se le llama prueba funcional o prueba orientada al diseño.

Con la prueba de la caja negra, intentamos encontrar errores de las siguientes categorías:

* Funciones incorrecta o ausente
* Errores de interfaz
* Errores en estructuras de datos o en accesos a base de datos
* Errores de rendimiento

Una prueba de caja negra exhaustiva requeriría la generación de un caso de prueba por cada combinación posible (válida o no válida) de los valores de entrada, lo cual resulta imposible en la mayor parte de los casos por producirse una explosión combinatoria. Por eso se utilizan diferentes criterios a la hora de restringir el conjunto de casos de prueba.

Los métodos de selección del conjunto de casos de prueba a utilizar son: Caso de prueba por valores límites y caso de prueba de validación.

* + - 1. **CASO DE PRUEBA POR VALORES LIMITES**
* Los campos numéricos se probarán los valores limites inferiores y superiores que pueda soportar
* Los campos de texto se probarán los valores mínimos que pueda soportar.
  + - 1. **CASO DE PRUEBA POR VALIDACION**
* Campos de textos: asegurarnos de que no se ingresen valores inválidos.
  + 1. **FORMULARIOS A UTILIZAR PARA LOS CASOS DE PRUEBAS.**

Primeramente se harán las Pruebas de Unidades por cada ABM y luego se procederá a las Pruebas de Integración para poder comprobar las funcionalidades en conjunto.

|  |  |  |
| --- | --- | --- |
| **Número de Prueba** | **¿Se realizó correctamente la prueba?** | **¿Coinciden los resultados obtenidos con los resultados esperados?** |
|  |  |  |
|  |  |  |

* **MODELO DE FORMULARIO DE BASE DE PRUEBAS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** | | | |
| **Objetivo:** |  | | |
| **Caso de Prueba:** |  | | |
| **Resultado Esperado:** |  | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

* **CASO DE PRUEBA POR VALORES LIMITES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE ALUMNOS | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**a”**. * En el campo **Apellidos** ingresaré “**b”**. * En el campo **Lugar de Nacimiento** ingresaré “**a”**. * En el campo **Fecha** ingresaré **“2”**. * En el campo **Cédula de Identidad N°** ingresaré “**1”**. * En el campo **Edad** ingresaré **“1”**. | | |
| **Resultado Esperado:** | Que no permita introducir los valores con un solo dígito como se describió en los casos de pruebas y que muestre el mensaje de alerta o valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE ALUMNOS | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**Es un nombre extenso de una persona porque es un caso de prueba”**. * En el campo **Apellidos** ingresaré “**Es un apellido extenso de una persona porque es un caso de prueba”**. * En el campo **Lugar de Nacimiento** ingresaré “**El lugar de nacimiento es bastante extenso porque es un caso de prueba”**. * En el campo **Fecha** ingresaré **“123/13/2020”**. * En el campo **Cédula de Identidad N°** ingresaré “**145678900983”**. * En el campo **Edad** ingresaré **“123455”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE DOCENTES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**a”**. * En el campo **Apellidos** ingresaré “**b”**. * En el campo **Lugar de Nacimiento** ingresaré “**a”**. * En el campo **Fecha** ingresaré **“2”**. * En el campo **Cédula de Identidad N°** ingresaré “**1”**. * En el campo **Edad** ingresaré **“1”**. * En el campo **Matricula Número** ingresaré **“0”**. * En el campo **Titulo Obtenido** ingresaré **“a”**. | | |
| **Resultado Esperado:** | Que no permita introducir los valores con un solo dígito como se describió en los casos de pruebas y que muestre el mensaje de alerta o valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE DOCENTES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**Es un nombre extenso de una persona porque es un caso de prueba”**. * En el campo **Apellidos** ingresaré “**Es un apellido extenso de una persona porque es un caso de prueba”**. * En el campo **Lugar de Nacimiento** ingresaré “**El lugar de nacimiento es bastante extenso porque es un caso de prueba”**. * En el campo **Fecha** ingresaré **“123/13/2020”**. * En el campo **Cédula de Identidad N°** ingresaré “**194373438382382”**. * En el campo **Edad** ingresaré **“13234”**. * En el campo **Matricula Número** ingresaré **“034234234234232”**. * En el campo **Titulo Obtenido** ingresaré **“El título obtenido es una descripción muy extensa porque esto es un caso de prueba”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE PERSONAL ADMINISTRATIVO | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**a”**. * En el campo **Apellidos** ingresaré “**b”**. * En el campo **Lugar de Nacimiento** ingresaré “**a”**. * En el campo **Fecha** ingresaré **“2”**. * En el campo **Cédula de Identidad N°** ingresaré “**1”**. * En el campo **Edad** ingresaré **“1”**. * En el campo **Cargo** ingresaré **“b”**. * En el campo **Titulo Obtenido** ingresaré **“a”**. | | |
| **Resultado Esperado:** | Que no permita introducir los valores con un solo dígito como se describió en los casos de pruebas y que muestre el mensaje de alerta o valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE PERSONAL ADMINISTRATIVO | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Nombres** ingresaré “**Es un nombre extenso de una persona porque es un caso de prueba”**. * En el campo **Apellidos** ingresaré “**Es un apellido extenso de una persona porque es un caso de prueba”**. * En el campo **Lugar de Nacimiento** ingresaré “**El lugar de nacimiento es bastante extenso porque es un caso de prueba”**. * En el campo **Fecha** ingresaré **“235/13/2030”**. * En el campo **Cédula de Identidad N°** ingresaré “**1355354368977”**. * En el campo **Edad** ingresaré **“4241”**. * En el campo **Cargo** ingresaré **“Este es un nombre de cargo muy extenso porque es un caso de prueba”**. * En el campo **Titulo Obtenido** ingresaré **“El título obtenido es una descripción muy extensa porque esto es un caso de prueba”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRAR MATERIAS | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Disciplina** ingresaré “**a”**. | | |
| **Resultado Esperado:** | Que no permita introducir los valores con un solo dígito como se describió en los casos de pruebas y que muestre el mensaje de alerta o valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRAR MATERIAS | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Disciplina** ingresaré “**La descripción de la disciplina es muy extensa porque este es un caso de prueba”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE PUNTAJES Y/O CALIFICACIONES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Calificación** ingresaré “**9”**. * En el campo **Turno** ingresaré **“h”** * En el campo **Etapa** ingresaré **“u”** | | |
| **Resultado Esperado:** | Que muestre el mensaje de alerta o de valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE PUNTAJES Y/O CALIFICACIONES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Calificación** ingresaré “**993872372372873”**. * En el campo **Turno** ingresaré **“Este es un turno bastante extenso porque esto es un caso de pruebas”** * En el campo **Etapa** ingresaré **“Esta es una descripción de etapa bastante extenso porque esto es un caso de pruebas”** | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** PLANIFICACION DE TAREAS Y EXAMENES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Especificar** ingresaré “**a”**. * En el campo **Observación** ingresaré **“b”**. | | |
| **Resultado Esperado:** | Que muestre el mensaje de alerta o de valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** PLANIFICACION DE TAREAS Y EXAMENES | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Especificar** ingresaré “**Esta es una especificación muy extensa porque esto es un caso de pruebas”**. * En el campo **Observación** ingresaré **“La descripción de la planificación de tareas y exámenes es bastante extensa porque esto es un caso de pruebas, esta descripción puede alcanzar hasta 2000 caracteres”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** SUBIR MATERIALES DE APOYO | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Tema** ingresaré “**a”**. * En el campo **Fecha** ingresaré **“0”**. * En el campo **Detalles**  ingresaré **“b”**. | | |
| **Resultado Esperado:** | Que muestre el mensaje de alerta o de valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** SUBIR MATERIALES DE APOYO | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Tema** ingresaré “**Este es un tema bastante extenso porque esto es un caso de prueba que busca generar errores”**. * En el campo **Fecha** ingresaré **“12/23/1000”**. * En el campo **Detalles**  ingresaré **“Este es un detalle de descripción donde se sube materiales que es bastante extenso y puede alcanzar hasta 1000 caracteres porque es un caso de pruebas”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** INFORME DE ASISTENCIA | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Mes Correspondiente** ingresaré “**a”**. * En el campo **Día** ingresaré **“b”**. * En el campo **Fechas**  ingresaré **“1”**. * En el campo **Turno** ingresaré **“c”**. * En el campo **Horario** ingresaré **“0”**. | | |
| **Resultado Esperado:** | Que muestre el mensaje de alerta o de valores no válidos al presionar el botón **Guardar**. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** INFORME DE ASISTENCIA | | | |
| **Objetivo:** | Encontrar errores al ingresar datos en el campo de texto. | | |
| **Caso de Prueba:** | * En el campo **Mes Correspondiente** ingresaré “**Este es un mes bastante extenso porque esto es un caso de pruebas”**. * En el campo **Día** ingresaré **“Este es un día bastante extenso porque esto es un caso de pruebas”**. * En el campo **Fechas**  ingresaré **“123/13/2014”**. * En el campo **Turno** ingresaré **“Este es un turno bastante extenso porque esto es un caso de pruebas”**. * En el campo **Horario** ingresaré **“100:00”**. | | |
| **Resultado Esperado:** | Que no permita escribir mayor cantidad de caracteres respecto al tipo de datos y con respecto a los números de caracteres que soporta la base de datos. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

* **CASO DE PRUEBA POR VALIDACION**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRO DE ALUMNOS, DOCENTES Y PERSONAL ADMINISTRATIVO | | | |
| **Objetivo:** | Encontrar errores al presionar el botón **Guardar**. | | |
| **Caso de Prueba:** | * Todos o algunos de los campos estarán vacios. * Todos los campos o algunos estarán con sus valores por defecto. * En el campo **Nombres, Apellidos, Lugar de Nacimiento** ingreso **“números con letras”, “solamente números”.** Ejemplo: **“Mari23a”, “Sanabr9u9a”, “Encarnación83239”.** * En el campo **Fecha de Nacimiento** ingreso **“74/febrero/90”.** * En el campo **Cédula de Identidad** ingreso **“99i9998i0”** * En el campo **Edad** ingres **“12años”** | | |
| **Resultado Esperado:** | Que muestre un mensaje donde diga que los valores introducidos son inválidos o cualquier mensaje de alerta, al presionar el botón **Guardar.** | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** LOGIN | | | |
| **Objetivo:** | Verificar que se validen correctamente los datos ingresados en el login. | | |
| **Caso de Prueba:** | **Usuario:adminaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaabbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbcccccccccccccccccccccccccccccccccccccc**  **Contraseña: había una vez un gato montés que tenia la cola al revez havia una vez un gato montes que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez havia una vez un gato montés que tenia la cola al revez.** | | |
| **Resultado Esperado:** | Como los datos ingresados son inválidos debería avisar al usuario del mismo por medio de un mensaje de alerta. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRAR MATERIAS | | | |
| **Objetivo:** | 1. Registrar una nueva materia y validar la restricción de longitud de los caracteres utilizados. 2. Eliminar una materia que está siendo utilizada. 3. Almacenar una materia sin nombre | | |
| **Caso de Prueba:** | 1. **Disciplina:** Esta es una materia que a muchos no les gusta por la dificultad que tiene. Esta es una materia que a muchos no les gusta por la dificultad que tiene. 2. **Disciplina:** \_\_ o %%% o 12433452345234 | | |
| **Resultado Esperado:** | 1. Como el nombre de la **disciplina o materia** excede la longitud permitida, no debería permitirse ingresar todos los caracteres. Informar que para disciplina solo se permite letras para denominar el nombre de la materia. 2. Informar que no se puede eliminar la materia por un mensaje de alerta porque está siendo utilizada. 3. Informar que no se puede almacenar una materia sin nombre. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** REGISTRAR GRADOS/CURSOS | | | |
| **Objetivo:** | 1. Registrar un nuevo grado/curso y validar la restricción de longitud de los caracteres utilizados. 2. Eliminar un grado/curso que está siendo utilizada. 3. Almacenar un grado/curso sin nombre | | |
| **Caso de Prueba:** | 1. **Grado/Curso:** Este es un grado al que corresponde una cantidad de alumnos. 2. **Grado/Curso:** \_\_ o %%% o 12433452345234 | | |
| **Resultado Esperado:** | 1. Como el nombre del **grado o curso** excede la longitud permitida, no debería permitirse ingresar todos los caracteres. Informar que para grado/curso solo se permite letras para denominar el nombre de la grado o curso. 2. Informar que no se puede eliminar el grado o curso seleccionado por un mensaje de alerta porque está siendo utilizada. 3. Informar que no se puede almacenar un grado o curso sin nombre. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de Prueba:** | **Sistema:** | **Entorno Tecnológico:** | **Testeador:** |
| **Módulo o ABM:** PLANIFICAR TAREAS O EXAMENES | | | |
| **Objetivo:** | 1. Verificar que las fechas establecidas para las tareas o exámenes no son inferior a la fecha del día de la planificación. 2. Verificar la validación de datos. | | |
| **Caso de Prueba:** | 1. Ingresar una fecha vieja con respecto a la fecha del sistema. 2. Ingresar números en el campo de **Puntaje Establecido** | | |
| **Resultado Esperado:** | 1. Informar que la fecha establecida del examen o la fecha de entrega de alguna tarea ya es una fecha inválida porque la fecha actual supera al mismo. 2. Informar que en puntaje establecido no se pueden ingresar letras u otros caracteres, solamente números. | | |
| **Resultado Obtenido:** |  | | |
| **Observación:** |  | | |

***Gestión de Configuración***

Dentro de un proyecto de desarrollo de software las tareas para la gestión de configuración son aquellas encargadas de mantener la integridad de los elementos involucrados. La gestión de configuración se enfoca en administrar los cambios a realizarse sobre elementos de los cuales ya se estableció una línea base y tener un encargado que decida si los cambios son necesarios y oportunos. Los formularios para la gestión de configuración son:

* Catálogo de elementos.
* Formulario para solicitud de cambios.
* Formulario de aceptación/rechazo de cambios.

El formulario de solicitud de cambios estará disponible para todos los integrantes del grupo, sin embargo el formulario de aceptación/rechazo de cambios es de uso exclusivo de la líder de proyecto, ya que será la encargada de aprobar los cambios de ser necesaria.

1. **LÍNEAS BASE**

Las líneas base o más conocidas por su término en inglés, "baselines", son "una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior pudiendo cambiarse solamente a través de procedimientos formales de control de cambios” según la traducción del estándar IEEE.

Cuando, en un proceso de desarrollo de software se realiza un producto (un documento, código fuente u otros), este producto pasa una etapa de producción en la que se puede modificar sin impedimentos. Sin embargo, una vez terminado este producto, tendrá una revisión formal y se aprobará por el equipo. Una vez que el producto ha sido aprobado, ya no podrá ser modificado de modo informal, sino que deberá seguirse un estricto control de cambios realizados sobre dicho producto para controlar correctamente su evolución.

Para establecer una línea base sobre determinado elemento se tuvo en cuenta la última versión estable sobre la cual se llegó a un acuerdo.

1. **CATÁLOGO DE ELEMENTOS**

Un elemento es, dentro de la configuración del software la información creada como parte del proceso de ingeniería es un documento, un conjunto completo de casos de prueba o un componente de un programa dado. Los siguientes **elementos dentro del proyecto “Sistema Académico CMJW”** son el objetivo de las técnicas de gestión de configuración:

1) Especificación del sistema.

2) Especificación de requisitos.

3) Especificación de diseños.

3.1. Descripción del diseño de datos

3.2. Descripciones del diseño de interfaces

4) Documento de gestión y análisis de riesgo.

5) Documentos de gestión de configuración

a. Estándares de programación

b. Estándares de diseño de interfaz

c. Catálogo de elementos

6) ABMs desarrollados.

7) Plan de pruebas.

El **catálogo de elementos** identifica a un elemento, su tipo, ubicación, línea base, propietario, versión. Se utiliza para identificar los elementos con que cuenta el proyecto.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Catálogo de Elementos V1.0 | | | | | |
| **Elemento de Configuración** | **Línea Base** | **Tipo de Elemento** | **Lugar de Almacenamiento** | **Responsable** | **Fecha de Actualización** |
| Especificación del Sistema | **V 1.0**  Establecida según la documentación del 23.03.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico/blob/master/README.md> | Pedro González | 23.03.13 |
| Especificación de Requisitos (Modelado de Requisitos) | **V 1.0**  Establecida según la documentación del año pasado. | Documentación | <https://github.com/pedrodamiangv/SistAcademico> | Mirian Saucedo y Pedro González | Noviembre.2012 |
| Diseño de Base de Datos | **V 1.0**  Diseño establecido el 30.03.13 | Base de Datos | <https://github.com/pedrodamiangv/SistAcademico/blob/master/Modelado%20de%20Datos.DM1> | Mirian Saucedo | 30.03.13 |
| Especificación de Módulos | **V 1.0**  Establecida en el documento relacionado el 30.03.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico/blob/master/PlanificacionSistAcademico.gan> | Pedro González | 30.03.13 |
| Especificación de Interfaces | **V 1.0**  Establecida en el documento relacionado el 20.04.13 | Documentación e Interfaz | <https://github.com/pedrodamiangv/SistAcademico/blob/master/Est%C3%A1ndares%20de%20dise%C3%B1o.docx> | Pedro González y Mirian Saucedo | 20.04.13 |
| Catálogo de elementos | **V 1.0**  Establecido en el documento relacionado el 22.04.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico/blob/master/catalogo%20de%20elementos.doc> | Pedro González | 22.04.13 |
| ABMs desarrollados | **V1.0**  Establecido desde el 22.04.13 | Especificado por Módulo | <https://github.com/pedrodamiangv/SistAcademico> | Equipo de desarrollo | - |
| Análisis de Riesgo | **V 1.0**  Establecida el 15.04.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico> | Mirian Saucedo | 15.04.13 |
| Plan de Pruebas | **V 1.0**  Establecida el 23.04.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico> | Mirian Saucedo | 23.04.13 |
| Estándares de programación | **V 1.0**  Establecido en el documento relacionado el 20.04.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico/blob/master/estandares%20de%20Programacion.doc> | Pedro González | 20.04.13 |
| Estándares de documentación | **V 1.0**  Establecido en el documento relacionado el 20.04.13 | Documentación | <https://github.com/pedrodamiangv/SistAcademico/blob/master/estandares%20de%20Programacion.doc> | Mirian Saucedo | 20.04.13 |

1. **CONTROL DE CAMBIOS**

En todo desarrollo se recomienda realizar los cambios de manera disciplinada. Las herramientas de control no garantizan un desarrollo razonable si cualquier miembro del equipo puede realizar los cambios que quiera e integrarlos en el repositorio sin ningún tipo de control.

Para garantizar que siempre disponemos de una línea base adecuada para continuar el desarrollo es necesario aplicar controles al desarrollo e integración de cambios.

1. **FORMULARIO DE SOLICITUD DE CAMBIO**

Es el formulario para solicitar un cambio el elemento en caso de que el cambio sea necesario y el elemento ya posea línea base.

|  |  |
| --- | --- |
| **N° de Solicitud:** | **Fecha de la Solicitud:** |
| **Elemento a cambiar:** | |
| **Responsable:** |  |
| **Persona que solicita:** |  |
| **Motivos o descripción del cambio:** |  |

1. **FORMULARIO DE ACEPTACION Y/O NEGACION DEL CAMBIO**

Una vez realizada la solicitud de cambio de algún elemento por parte de un integrante del equipo, la líder de proyecto analiza la solicitud y da a conocer el siguiente formulario:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fecha:** | | | **Elemento a cambiar:** | | |
| **Según solicitud de Cambio N°:** | | | **Responsable del Elemento:** | | |
| **Aprobado:** | **SI** |  | | **NO** |  |
| **Motivos:** |  | | | | |

***Estándares de Programación***

**Código de Aplicaciones**

1. **Nombre del Proyecto, archivos fuentes y estructura del proyecto**

Las aplicaciones Ruby on Rails generan automáticamente directorios, el nombre del proyecto estará relacionado con el modulo a desarrollar y estará todo en minúsculas y en caso de estar compuesto estarán separadas las palabras por el guion bajo.

Al desarrollar una aplicación Ruby on Rails por defecto se generan las siguientes carpetas:

<nombre\_proyecto>/

La carpeta principal que contiene todas las subcarpetas y archivos del proyecto. Tiene el mismo nombre del proyecto.

<nombre\_proyecto>/app

La carpeta que contiene subcarpetas en la que mayormente trabajaremos. Contiene el código fuente principal del proyecto.

<nombre\_proyecto>/app/controller

Esta carpeta contiene todos los archivos controladores, los nombres deben de comenzar siempre con la palabra controller.

<nombre\_proyecto>/app/helpers

Esta carpeta los archivos de ayuda para las clases controladores.

<nombre\_proyecto>/app/models

Esta carpeta contiene los modelos, que deben de ir relacionados con las tablas de base de datos. Si existe alguna tabla llamada usuario, debe existir también un modelo llamado usuario con sus mismos atributos llamados de la misma forma que en la base de datos.

<nombre\_proyecto>/app/views

Esta carpeta contiene las vistas del proyecto, todos los archivos html.erb, formularios y demás interfaces del sistema. Dentro de esta carpeta debemos de crear otras por cada modelo creado, con el mismo nombre del modelo que contendrá los métodos del controlador como ser show, index, new, edit.

<nombre\_proyecto>/app/views/layouts

Esta carpeta contiene los archivos que serán parte de todas las vistas.

<nombre\_proyecto>/config

La carpeta contendrá archivos de configuración que se cargarán al levantar la aplicación. Varias gemas utilizadas deberán ser configuradas aquí.

<nombre\_proyecto>/db: para archivos relacionados a base de datos, ya sea para su creación, edición o delete.

<nombre\_proyecto>/doc: para la documentación.

<nombre\_proyecto>/lib: para librerías adicionales

<nombre\_proyecto>/log: contiene el log del proyecto.

<nombre\_proyecto>/public: contiene los archivos públicos del proyecto.

<nombre\_proyecto>/script: contiene los scripts del proyecto.

<nombre\_proyecto>/test: aquí realizaremos los test de Ruby del proyecto

<nombre\_proyecto>/tmp: archivos temporales

<nombre\_proyecto>/vendor

En este directorio se almacenan bibliotecas provistas por terceras partes (como por ejemplo bibliotecas de seguridad o utilitarios de base de datos más allá de la distribución básica de Rails).

<nombre\_proyecto>/README

Este archivo contiene detalles básicos sobre Rails Application y la descripción de la estructura de archivos descripta arriba.

<nombre\_proyecto>/Rakefile

Este archivo es similar al Makefile de Unix, y sirve para construir, empaquetar y probar el codigo Rails. Este archivo es usado por el utilitario rake provisto con la instalación Ruby.

1. **Ubicación de los archivos de código fuente**

El código fuente principal se dividirá según lo establecido por la forma de programación del MVC, dividiendo todo lo relacionado con las clases, ya sean los atributos, las validaciones y métodos propios de cada clase estarán bajo una carpeta llamada model en el proyecto, cada clase deberá de tener la extensión de .rb y el nombre de la clase deberá ser la misma utilizada en la tabla de base de datos.

Todo lo que sea métodos para comunicar la vista (la página html) con la clase (con el modelo) estará bajo una carpeta llamada controller, que todos los nombres de los archivos ahí dentro deberán empezar con la palabra controller y tendrá la extensión de .rb así como en el modelo.

Todo lo relacionado a páginas html estará bajo la carpeta view y tendrá la extensión .html.erb y se dividirán básicamente en show (para mostrar datos), edit(para editar datos), new(para agregar datos), index(para listar datos), y otras en casos especiales. Cada modelo tendrá todos estos archivos bajo su propia carpeta en la carpeta view.

*Por ejemplo:*

Si tengo un modelo llamado usuarios.rb

Dentro de la carpeta view debo de tener la carpeta usuarios y dentro de ella tendré los archivos show.html.erb, index.html.erb, edit.html.erb, new.html.erb…

model/usuarios.rb

controller/controller\_usuarios.rb

view/usuarios/new.html.erb

view/usuarios/show.html.erb

……

1. **Nombre de objetos**

Dentro de la aplicación Web, vamos a tener distintos tipos de objetos, los nombres de los mismos estarán en minúscula y deberán expresar la función a desarrollar por el objeto. Estarán completamente en minúscula y en no podrán ser compuestos.

1. **Clases**

Los nombres de las clases deben empezar en mayúscula y si posee un nombre compuesto se utilizará la técnica de la notación Camel empezando siempre en mayúscula. El nombre del archivo deberá estar en minúscula y si es un nombre con palabras compuestas estás deberán estar separadas por el guion bajo, siguiendo con la convención de Ruby.

*Por ejemplo:*

Si queremos crear una clase para gestionar los usuarios, está deberá llamarse Usuarios y el archivo se guardará como usuarios.rb

Al crear un objeto de la clase Usuarios esté podrá llamarse usuario, en caso de ser compuesto usuario\_nombre.

1. **Métodos**

Los nombres de los métodos deben de expresar la función del mismo. En caso de llevar parámetros, éstos deberán estar dentro de un paréntesis separado por comas.

def <nombre\_metodo>

Los nombre de los métodos deben de estar en minúsculas y en caso de ser compuestos, cada palabra deberá estar separado por el guion bajo.

1. **Constantes**

Los nombres de las constantes globales y locales deberán expresar la intención del mismo, en caso de ser compuestos estarán separados por el guion medio.

Los nombres de constantes siempre se escribirán en MAYUSCULA

Por ejemplo DIRECCION-LOCAL

Como el nombre de la constante lo va a definir el consultor, se debe de ser lo más descriptivo posible, dejando a entender el valor del mismo. Si el nombre de la constante esta compuesta de varias palabras, debemos de digitar las palabras unidas como nombre de la constante, las palabras deben estar en mayúscula separadas por el guion medio.

1. **Variables**

Las variables dependiendo de su alcance se dividen en globales y locales, el nombre de la variable consta de un prefijo seguido del nombre de la variable definido por el consultor.

El nombre debe de tener la siguiente sintaxis:

<ubicación><tipo de dato>\_<nombre de la variable>

El tipo de dato puede omitirse en caso de ser una cadena o un tipo numérico.

Los prefijos son los siguientes:

Alcance:

Global (g)

Local (l), todas las variables locales pueden omitir el uso de este prefijo

***Tipo de Dato***

|  |  |
| --- | --- |
| Nombre | Prefijo |
| String | s |
| Boolean | b |
| Fixnum | n |
| Integer | i |
| float | l |
| List | lst |
| Array | arr |
| Objeto | obj |
| DateTime | dt |
| Date | dt |
| Time | dt |

Como el nombre de la variable lo va a definir el consultor, debe ser lo más claro posible, dejando a entender el valor del mismo. Si el nombre de la variable consta de varias palabras, debemos de digitar las palabras separadas por el guion bajo como nombre de la variable.

*Por ejemplo:*

Si vamos a necesitar una variable que va a contener un primer nombre (dos palabras) de una persona, debe ser como el siguiente ejemplo:

ls\_nombre\_empleado

Si necesitamos crear una variable que va a contener un estado (una palabra), debe ser como el siguiente ejemplo:

ls\_estado o s\_estado, como es um string puede solo llamarse estado.

**Variables de contadores**

Se debe utilizar i, j, k, l, m, n para los nombres para las variables de contadores utilizados.

**Posición de variables dentro del Código Fuente**

Todas las variables sin distinción deben quedar declaradas en la cabecera o header del método, función o evento. A pesar que el código fuente permita otra acción se requiere que todas queden declaradas en la posición indicada.

*Por ejemplo:*

def XYZ

ln\_edad = 1

nombre = “Pedro”

array\_usuarios = []

Un mal ejemplo

def XYZ

ln\_edad = 1

nombre = “Pedro”

if llamarDatos()

#cualquier\_cosa

end

array\_usuarios = []

1. **Controles Visuales**

Los controles visuales deben poseer un nombre estándar, el nombre debe ser el mismo con el cual fue nombrado la clase, el mismo nombre del atributo declarado en el modelo del objeto deberá de ser usado en la vista, es decir, en las páginas html.erb ya sea para mostrar el atributo, para editar, guardar o listar.

1. **Documentación interna**
   1. **Métodos**

Para los métodos debemos de seguir los siguientes estándares. Antes de la declaración de cada método debemos de colocar un encabezado.

*Por ejemplo:*

#Explicar para qué sirve el método creado

def método

end

Este encabezado debe ser agregado una línea encima del método, debe explicar cuál es el propósito del método creado.

Dentro de cada método opcionalmente po<nombre\_proyecto>s poner líneas con comentarios si es que escribimos algún código con dificultad de entenderse. Éstas no deberán de superar una línea como máximo.

* 1. **Código fuente comentado**

Se entiende como código fuente comentado aquellas líneas de código fuente que fueron puestas entre comentarios por fines de reemplazo de código. Este código comentado no tiene funcionalidad alguna dentro del sistema por lo tanto debe de ser eliminado por completo del mismo, solo en casos excepcionales.

*Por ejemplo:*

def cargarSegundoCampo

#condicion = false

#if condición

return CAMPO

end

Las líneas anteriores comentadas no deben de permanecer en el código fuente estable que se usará para realizar el sistema.

Esta virtud de programación será altamente evaluada durante el proceso de auditoría.

* 1. **Tabulación**

Se respetará lo establecido convencionalmente por Ruby, la tabulación será de 2(dos) espacios entre cada bloque.

*Por ejemplo:*

Class MyClass

Def initialize

@my\_class = “Mi clase”

end

* 1. **Asignación de Variables y el uso de signos.**

Para inicializar cualquier objeto en el sistema seguiremos también un estándar, pondremos el nombre del objeto y/o variable, un espacio, el signo = seguido de un espacio y el valor asignado. Siempre que escribamos un signo ya sea el signo + - \* = dejaremos espacios antes y espacios después, no es necesario estos espacios entre signos.

*Por ejemplo:*

array\_usuarios = []

suma += 1

if a == b

# Formatos

**Formato de Fecha**

Los formatos de fechas deben ser dd/MM/yyyy.

**Formato de Mensajes**

Mensajes de validación de campos requeridos

Los mensajes de validación de campos requeridos deben ir al costado de los campos y deben ser de color rojo. Ejemplo:

Ingrese Descripción

**Mensajes de validación de fechas**

Los mensajes de validación de fechas deben ir al costado de los campos y deben ser de color rojo. Ejemplo:

Fecha (dd/mm/yyyy) entre: 1/1/2000-31/12/2078

**Mensajes de validación de búsqueda de información**

Los mensajes de validación de búsqueda de información deben ir en la parte inferior de la pantalla y deben ser “No se encuentran registros”.

***Estándares de Diseño***

En este artículo se muestra el diseño de clases web que se propone para desarrollar el sistema y los diagramas necesarios para la implementación de la solución propuesta.

**Principios de Diseño del Sistema**

El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

**Estándares en la Interfaz de la Aplicación**

La primera impresión del usuario cuando visita una aplicación web la brinda el diseño de la interfaz. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación se detallan. En el sistema, el diseño de la interfaz está basado en páginas Web, se utilizan las tonalidades suaves y refrescantes. El vocabulario manejado es lo menos técnico posible, acercándose al utilizado por los usuarios.

Se utiliza una gema de Ruby on Rails para tal efecto llamada boostrap. Se utilizan imágenes identificativas como vínculos para la navegación dentro del sitio web. La letra utilizada en la mayor parte del sistema es Times New Roman (12, 16) lográndose un diseño estándar en casi todo el sitio. Los mensajes de error son pequeños, rojos y en Español.

El sistema no contendrá imágenes, pero sí puede contener iconos pequeños para las acciones.

El fondo de las páginas es de color blanco para mayor frescura de la vista. Todo esto se ha hecho con el objetivo de que el uso del sitio brinde comodidad y confort al usuario.

**Concepción General de la Ayuda**

La ayuda constituye una parte imprescindible en todo sistema. En el menú principal aparece una opción Ayuda que explicará de forma detallada cómo funciona el sistema, tratando de aclarar los puntos que podría causar duda al usuario. Este menú aparece en todas las páginas para mayor comodidad del usuario.

Cada una de las opciones del sistema, así como las consideraciones que se asumen en la ejecución de ellas, está propiamente documentada para evitar cualquier tipo de confusión por parte del usuario. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario.

**Tratamiento de Excepciones**

El diseño de la interfaz ha estado dirigido a evitar errores, teniendo en cuenta paralelamente la creación de interfaces útiles y amigables. Se ha buscado simplificar la validación de los datos garantizando una validación intrínseca de los mismos, procurando facilitar la corrección de errores lógicos tanto en la introducción de la información como en cualquier otro momento del tratamiento de la misma.

La técnica para el manejo de los errores en el sistema se concebirá de manera que cuando ocurra un error se genere una excepción; es decir, la ejecución normal se detenga y se transfiera el control a la zona de tratamiento de excepciones. Las excepciones internas se generan automáticamente por el sistema.

Los mensajes de error que emita el sistema ya sea de la base de datos o de la aplicación cliente se captarán y se traducirán a un lenguaje comprensible para el usuario.

Los formularios manejan los datos en memoria y solo se actualiza en la base de datos cuando se indique salvarlos.

**Estándar de Diseño**

El sistema está compuesto por una serie de plantillas, unas para el ingreso de información por parte del usuario (Entradas) y otras para enviar información solicitada al usuario (Salidas), todas ellas trabajarán con un diseño definido.

A continuación presentamos un esquema que muestra la estructura y distribución de las secciones que forman parte de las plantillas del sistema (Figura 1.1)

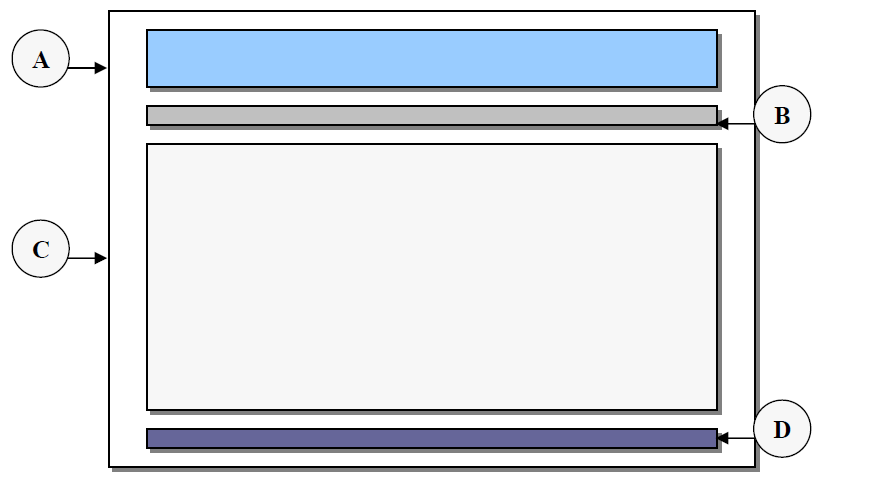


Figura 1.1

Las secciones serán las siguientes:

1. Barra de título (Header)
2. Barra de navegación
3. Área de trabajo
4. Pie de página (Footer)

El sistema tendrá varias pantallas, diseñadas en el análisis de diseño usada en la metodóloga RUP, entre ellas mencionamos las principales a continuación:

* **Pantalla de Ingreso:** Pantalla de autenticación de usuarios, aquí se solicita la identificación de usuario y clave de acceso. Posteriormente el sistema verifica la validez de información, de lo contrario se deniega el acceso.
* **Pantalla de Bienvenida e Información al sistema:** Esta pantalla es la siguiente que verá un usuario, ya luego de logearse. La página tendrá una breve información de la institución y las acciones que puede hacer el usuario logeado de acuerdo a su rol: alumno, docente o administrativo.
* **Pantalla de listado de Alumnos:** Muestra los alumnos registrados en el sistema, ya sea por año, curso o materia.
* **Pantalla de Registro de Alumnos:** Muestra un formulario con el cual se puede registrar alumnos en el sistema.
* **Pantalla de listado de Docentes:** Muestra los docentes registrados en el sistema, ya sea por año, curso o materia.
* **Pantalla de Registro de Docentes:** Muestra un formulario con el cual se puede registrar docentes en el sistema.
* **Pantalla de listado de Administrativos:** Muestra los administrativos registrados en el sistema.
* **Pantalla de Registro de Administrativos:** Muestra un formulario con el cual se puede registrar administrativos en el sistema.
* **Pantalla de Registro de Puntajes y Calificaciones**: Muestra un formulario, en el cual el docente puede ingresar las calificaciones alumno por alumno de acuerdo a una tarea asignada.
* **Pantalla de** **asignación de tareas:** Mostrará un formulario donde el docente puede asignar tareas a sus alumnos.
* **Pantalla de materiales de clases:** Muestra un formulario desde donde el docente puede subir materiales para que queden disponibles para sus alumnos.
* **Pantalla de Materias:** Muestra las tareas de la materia, y los puntajes y calificaciones obtenidos por cierto alumno hasta el momento. Muestra además los materiales de clase que subió el profesor.

Todas estas pantallas no todos los usuarios podrán ver. Depende del rol que ejerza el usuario logeado podrá ver estas pantallas. Éstas básicamente son las pantallas principales del sistema.

***Estándar de Diseño de Modelado de Datos***

* **Nombres de las Tablas**

Los nombres de las tablas se nombran en plural y completamente en mayúscula.

Ejemplo: “ALUMNOS”

* **Claves Primarias**

Los nombres de las claves primarias se realizarán con el nombre de la tabla pero en singular, seguido de un guión bajo y la palabra “Id”.

Ejemplo: “Alumno\_Id”

* **Nombre de los Campos**

Los nombres de los campos se escriben las primeras letras en mayúsculas, las demás en minúsculas; para separar dos palabras utilizamos el guión bajo.

Ejemplo: “Fecha\_Nacimiento”

***Recursos del Proyecto***

* **RECURSOS SOFTWARE**
* GanttProject 2.0.10
* Cocomo II
* Git y GitHub
* SQL Power Architect
* Ruby on Rails, Gem Boostrap
* MySQL
* Xampp
* Visual Paradigm
* **ORGANIZACIÓN DE LOS RECURSOS HUMANOS**
* **Roles asignados al equipo de trabajo**

**Líder del Proyecto:** Mirian Saucedo

**DBA:** Pedro González y Mirian Saucedo

**Documentación:** Pedro González y Mirian Saucedo

**Tester:** Pedro González

**Desarrolladores:**

* Pedro González
* Mirian Saucedo
* **Responsabilidades asignadas**

**Del líder del proyecto:**

* + Gestionar el presente plan.
  + Interactuar con el cliente y futuros usuarios del Sistema para evaluar el surgimiento de requisitos no contemplados o para debatir sobre posibles solicitudes de cambio.
  + Analizar con el encargado de gestión de configuración los posibles cambios solicitados por el cliente y/o futuros usuarios en el caso que se presenten.
  + Informar a los desarrolladores sobre los cambios que se deben implementar y asignar las tareas de modificación a los miembros responsables de ellos en el equipo de desarrollo.
  + Controlar que las actividades se cumplan de acuerdo al plan del proyecto.
  + Tomar las medidas necesarias para asegurar el cumplimiento del plan en el tiempo establecido.
  + Convocar reuniones de revisión de todo el trabajo de manera que más de una persona este familiarizada con el trabajo
  + Verificar que el software cumpla con los requisitos establecidos.
  + Implementar normas dentro del equipo de trabajo para así evitar retrasos.

**De los desarrolladores del Proyecto:**

* + Comunicarse con los integrantes del equipo de desarrollo para encontrar posibles problemas técnicos o de personal.
  + Elaborar informes semanales y entregar al encargado de desarrollo.
  + Cumplir con las tareas encomendadas en el plazo establecido.
  + Informar a tiempo de cualquier retraso o problema que surja durante alguna tarea que se le haya sido asignada
  + Informar cuando una tarea haya sido culminada exitosamente.