***PARTE 1***

**“ PERFIL DEL PROYECTO ”**

En esta primera parte se hará una descripción general del proyecto a ser desarrollado. Se dará a conocer lo siguiente: (1) **La** **situación problemática** a la cual el proyecto dará solución y (2) **Los antecedentes** sobre el surgimiento, desarrollo de este tipo de software y (3) **Los objetivos** que se pretenden alcanzar. Todo esto con el fin de tener un enfoque general que nos servirá como guía hacia el entendimiento, estudio y posterior desarrollo del presente proyecto.

# 

# PERFIL DEL PROYECTO

### Título

**SOFTWARE PARA EL DISEÑO Y ELABORACIÓN DE EXÁMENES EDUCATIVOS**

**(GESTOR-TEST)**

### Introducción

Con la evolución de la tecnología, hoy en día se ha conseguido automatizar los procesos en cualquier actividad con el propósito de alivianar la carga y tensión de trabajo por parte de los involucrados; la automatización de procesos les permite el ahorro de costo, tiempo y esfuerzo lo cual los guiará hacia la concentración de tareas más críticas que demandan mucho más cuidado, para poder así, alcanzar mejores resultados de calidad y productividad.

La aplicación de la tecnología en el ámbito educativo no es una excepción. Se ha conseguido automatizar la gestión de exámenes, haciendo que el personal docente se centre más en pruebas objetivas en beneficio del aprendizaje del alumno, mediante la interpretación de resultados obtenidos de un proceso sencillo de evaluación y no en una corrección cuantitativa que involucra esfuerzo de elaboración, pérdida de tiempo y costo de recurso de materiales.

Las nuevas tecnologías al ámbito de la docencia han supuesto un cambio sustancial en los modelos clásicos de enseñanza-aprendizaje, ahora mucho más centrados en el alumno. La utilización adecuada de las TIC en la enseñanza puede contribuir sustancialmente a mejorar la calidad de la educación y la formación y a adaptarlas a los requisitos de una sociedad basada en el conocimiento.

Se han desarrollado muchos softwares para asistir a la gestión de exámenes, y basados en estos, se consigue el desarrollo de software con más beneficios para los interesados. Pero es sabido que la ingeniería de software obliga a los desarrolladores a trabajar bajo ciertos estándares con el fin de conseguir calidad en el producto y productividad en el proceso de desarrollo para ello deben regirse a dichos estándares para obtener un trabajo organizado, serio y disciplinado.

Con el fin de establecer una línea base o marco de trabajo sobre el cual se sostendrá el proyecto para ser ejecutado y pueda terminarse con éxito.

En este artículo se presenta una herramienta que asiste al profesor en la tarea completa de la evaluación, dotando de mecanismos de edición y selección de preguntas para asistir en la fase del diseño del examen, proporcionando una plataforma segura cliente-servidor para realizar la prueba y efectuando la corrección automática de preguntas.

El presente documento describe el proceso para el desarrollo de GestorTest, una herramienta software que asista a la gestión de exámenes, dirigido para el personal docente de entidades educativas. El proceso de desarrollo se rige bajo la metodología PUDS (proceso unificado de desarrollo de software), descrito en sus cuatro flujos de trabajo: captura de requisitos, análisis, diseño e implementación, mediante los cuales, se describen y diseñan modelos correspondientes a cada fase, haciendo uso de la herramienta UML 2.0, con el fin de exponer de manera clara y precisa todos los requisitos del software en cada una de sus fases. Previo al proceso de desarrollo se plantea el perfil del proyecto, en donde daremos a conocer de manera general la herramienta a ser desarrollada, planteando objetivos y requerimientos; tanto funcionales como no funcionales.

Por último, en el apartado de conclusiones se exponen los beneficios y las dificultades que se podrían derivar de su uso y se apuntan posibles mejoras.

### Antecedentes

La utilización del ordenador como asistente en el proceso de evaluación se ha centrado en dos áreas principales. (a) Por un lado la gestión administrativa del diseño y composición y la valoración automática de la prueba, y (b) la realización online de la prueba por parte del estudiante. Herramientas como PILOT[5], QUIZIT[7] y ASSYST[8] fueron los primeros desarrollos de sistemas de evaluación online, en la actualidad representados por una nueva generación que integran estas capacidades en completas plataformas de elearning como WebCT[6] y Blackboard[9].

Una característica común de todas ellas es que su objetivo ha sido la enseñanza a distancia y no la formación presencial, donde los niveles de ritmo, exigencia y equidad de evaluación son distintos. Y a nivel de la implementación, no utilizar procedimientos estandarizados de acceso a la información a través de Internet.

Los Sistemas Generadores de Exámenes, responden a una necesidad interna de las Unidades Académicas por querer trasparentar los procesos internos de evaluaciones ordinarias, semestrales y hasta extraordinarias; todo lo anterior en base a los antecedentes históricos que se han presentado y que hoy en día no podemos decir que sean prácticas actuales, pero sí que aún se llegan a dar casos aislados de mercadeo de calificaciones en las unidades de aprendizaje consideradas como duras.

* 1. **Descripción del problema**

Si bien es sabido que la evaluación es una actividad sistemática y permanente, integrada dentro del proceso educativo con el fin de ayudar y orientar al alumno y, finalmente, criticar y revisar programas, métodos y recursos. Este debería ser, más que un examen tradicional, una valoración cualitativa y no cuantitativa. Para ello, el personal docente debe, quien aparte de diseñar, seleccionar y estructurar pruebas objetivas de valoración debe analizar los resultados y emitir, bajo criterio formal, un informe de aprovechamiento general del alumno.

Prácticamente en cualquier institución educativa se construyen pruebas de evaluación, sin embargo ésta es una tarea compleja que pocas veces se realiza con las debidas consideraciones.

En una gran mayoría de las Unidades Educativas e Instituciones bolivianas, el docente se encuentra con dificultades al momento de realizar sus exámenes, intentando evitar o disminuir problemas comunes entre sus estudiantes. Tradicionalmente una buena parte de los profesores han considerado la tarea de corregir exámenes algo tedioso.

Todos sabemos que la evaluación es fundamental para el aprendizaje, pero también que si se hace bien lleva mucho trabajo.

Esta situación nos motiva al desarrollo de una herramienta para asistir al proceso de gestión de exámenes, pensada para aminorar el trabajo del personal docente, quien podrá desarrollar sus exámenes de una manera más sencilla y sin requerir de esfuerzos innecesarios.

### 1.4.Objetivos

* **Objetivo General**

Desarrollar un software para la elaboración y evaluación de test y/o exámenes para Centros Educativos.

* **Objetivos Específicos**
* Definir técnicas de preguntas que permita la elaboración de diferentes modelos de exámenes.
* Definir las funcionalidades del software acorde a los requerimientos obtenidos del cliente.
* Determinar las herramientas adecuadas a ser utilizadas para el diseño de modelos e implementación del software.

### Alcance del Proyecto

* **Requerimientos Funcionales**
* **Diseñar Exámenes:**

Debe permitir la creación y/o elaboración, modificación y eliminación de los diferentes modelos de exámenes categorizados por materias; de uso exclusivo para profesores. Los modelos de exámenes serán diseñados por el profesor haciendo uso de diferentes opciones; tales como la elección de tipos de preguntas con sus correspondientes respuestas y puntuaciones asignadas por el profesor, la adición de imágenes en cada una de ellas en caso que se requiera y la asignación de la duración del tiempo de la prueba. Las notas deben ser gestionadas desde la PC del profesor.

Debe poder visualizar los resultados de todas las pruebas categorizadas por alumno y ver la nota y/o calificación de cada prueba rendida, también deberá permitir visualizar el examen previamente elaborado en PDF para su posterior impresión.

* **Visualizar Exámenes:**

Los exámenes serán vistos por el alumno a través del visualizador de exámenes el cual debe permitir la ejecución del mismo. A través del visualizador, el alumno podrá rendir el examen durante el tiempo definido en dicha prueba, el cual a su vez, debe realizar la respectiva corrección una vez concluida la prueba y el profesor podrá recuperar los resultados y visualizarlos desde su PC.

### Requerimientos no funcionales

* **Rendimiento**

Se espera un tiempo de respuesta óptimo, y para ello debemos definir una arquitectura eficiente ya que el nivel de incidencia en cuanto a la arquitectura es alto.

* **Fiabilidad**

Debido a que el software a desarrollar solo asiste al proceso de gestión de exámenes y no tiene interdependencia con otros software consideramos que no es un “Software Crítico“, pero debido a que se podrían verse afectados los intereses tanto de los Estudiantes como de los Docentes, es necesario un buen grado de fiabilidad en cuanto a las operaciones de evaluación y corrección de exámenes.

* **Restricciones**
  + - **Restricciones Técnicas**
* El Sistema Operativo que podrá utilizarse será Windows XP.
* Se aplicará en el desarrollo del proyecto, el Proceso Unificado de Software (PUDs), utilizando el Lenguaje Unificado de Modelado (UML 2.1).
* Para el desarrollo del Sistema se utilizará como lenguaje de programación Netbeans IDE 7.1 y estándares XML.
* Para la documentación será necesario el uso de un editor de textos de Office XP profesional. Y Para la documentación del desarrollo se dará uso de una herramienta CASE que permite diseño del estándar UML 2.0.
  + - **Restricciones Legales**
* Ninguna funcionalidad del software estará bajo reglamentación de alguna autoridad. Por lo tanto no se tendrá ninguna restricción legal.
  + - **Restricciones De Recursos**
* El tiempo es un recurso limitado y no deberá ser mayor a 3 meses (no incluye el tiempo de realización del plan de proyecto).
* En cuanto a recursos humanos, se cuenta con un total de tres personas para la elaboración del proyecto.

### Recursos Para Desarrollar El Proyecto

**Tabla de Recurso**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Recurso*** | ***Cant.*** | ***Costo***  ***Unitario [$us]*** | ***%Depr.*** | ***Costo Unit Neto [$us]*** | ***Costo Totol [$us]*** | ***Modalidad***  ***de***  ***Adquisición*** |
|
| ***Hardware*** |  |  |  |  |  |  |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif PC [Anexo1] | 4 | 450 | 25 | 112,5 | 112,5 | Compra |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Servidor [Anexo 2] | 1 | 1410 | 25 | 352,5 | 352,5 | Compra |
| ***Software*** |  |  |  |  |  |  |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Windows XP | 1 | 200 | 30 | 60 | 60 | Compra |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif NetBeans IDE 6.9.1 | 1 | 387 | 30 | 116.1 | 116.1 | Compra |
| ***Gente*** |  |  |  |  |  |  |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Analista | 1 | 1000 | - | 1000 | 1000 | Contrato |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Desarrollador | 2 | 500 | - | 1000 | 1000 | Contrato |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Diseñador | 2 | 1400 | - | 2800 | 2800 | Contrato |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Secretaria | 1 | 100 | - | 100 | 100 | Contrato |
| ***Infraestructura*** |  |  |  |  |  |  |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Local | 1 | 150 | - | 150 | 150 | Alquiler |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Escritorios | 4 | 80 | 10 | 8 | 8 | Compra |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Serv. de Agua Potable | - | 25 | - | 25 | 25 | Servicio |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Internet | - | 40 | - | 40 | 40 | Servicio |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Teléfono fijo | - | 25 |  | 25 | 25 | Servicio |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Serv. de Electricidad | - | 38 | - | 38 | 38 | Servicio |
| ***Logística*** |  |  |  |  |  |  |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Material de Escritorio | - | 250 | - | 250 | 250 | Compra |
| C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Bullets\BD15061_.gif Refrigerio | - | 35 | - | 35 | 35 | Compra |

**Total costo de Producción**

**6112.1**

### 1.8.Beneficios para el cliente por parte del software

* Intuitivo y fácil de usar.
* El usuario podrá diseñar cualquier tipo de examen haciendo uso de cuatro distintas modalidades de preguntas: de selección múltiple, verdadero/falso, de completar y de responder.
* Sencilla y completa opción de asistencia a todo el proceso de gestión de exámenes.
* Diseño ampliamente configurable.
* Distribución inmediata de exámenes, previamente diseñados, hacia uno o varios nodos de la red Ethernet.
* Administrar notas por alumno, para tener un control de los exámenes realizados por el mismo.

### Estrategia De Desarrollo

* **PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE** **(PUDS)**

En el presente trabajo se hará uso del **Proceso Unificado** como la estrategia de desarrollo a ser utilizada. La elección de esta estrategia se debe a que posee características de proceso de desarrollo confiable que permiten la construcción de un software robusto y adaptable.

Sus características generales son las siguientes:

* Es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad”.
* Guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto.
* Describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.
* Describe qué entregables producir, cómo desarrollarlos y también provee patrones.
* Es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.
* El Proceso Unificado ha adoptado un enfoque que se caracteriza por:
* Interacción con el usuario continua desde un inicio
* Mitigación de riesgos antes de que ocurran
* Liberaciones frecuentes
* Aseguramiento de la calidad
* Involucramiento del equipo en todas las decisiones del proyecto
* Anticiparse al cambio de requerimientos

### 1.9.Herramientas de implementación

* + **JAVA**

Java será la utilizada herramienta como entorno de desarrollo para la implementación del software propuesto debido a las características y ventajas que ofrece a la hora de programar.

Puesto que Java se define como un poderoso lenguaje de programación orientado a objetos. Su sintaxis es muy similar a la de C o C++, lo cual representa un esfuerzo pequeño en la transición de un lenguaje a otro.

Esta herramienta, permite:

* Proporcionar un código compilado que se ejecute en distintos sistemas operativos con diferentes plataformas de hardware.
* Producir programas que sean robustos, libres de errores y fáciles de programar.
* Ser de fácil aprendizaje y uso.
* Servir como lenguaje de programación para red.

Las características y estándar de implementación de Java se describe en el apartado del [ANEXO 4] para más detalle de información sobre el mismo.

* **UML 2.0**

El Lenguaje Unificado de Modelado (UML, Unified modeling Language) será la herramienta que permitirá visualizar, especificar, construir y documentar los artefactos del software a ser desarrollado. Se hará uso UML debido a que proporciona una forma estándar de representar los planos de un sistema o software, y comprende tanto **elementos conceptuales** (como los procesos del negocio y funciones del sistema) y **elementos concretos** (como las clases escritas en un lenguaje de programación específico, esquemas de base de datos y componentes software reutilizables).

***PARTE 2***

**“FUNDAMENTACIÓN TEÓRICA”**

En esta sección se mostrará la información recopilada producto de una investigación sobre conceptos específicos que se precisan conocer para la adquisición de conocimiento y dominio del tema para posterior implementación al desarrollo del proyecto. Se presentan descripciones sobre técnicas de diseño de exámenes y modalidades de preguntas, se describe, de manera general, el PUDS (debido a su uso como estrategia de desarrollo para el presente proyecto), UML 2.0 como herramienta de modelado, se presentan también, uso de socket para la distribución de archivos, descripciones sobre XML como estándar para la distribución de archivos. Temas que son de interés y complemento en el presente proyecto.

### 2.1. FUNDAMENTACIÓN TEÓRICA

* 1. **PUDS (Proceso Unificado de desarrollo de software)**

Es un proceso Iterativo e Incremental, Usa los modelos con UML, los cuales proporcionan representaciones ricas desde el punto de vista semántico. La documentación se puede mantener y controlar electrónicamente con el uso de herramientas case. Minimiza la sobrecarga asociada con la generación y mantenimiento de la documentación.

Está centrado en la arquitectura, Las actividades de desarrollo bajo el RUP están dirigidas por los casos de uso, este soporta técnicas orientadas a objetos, es un proceso configurable que impulsa un control de calidad y una gestión del riesgo.

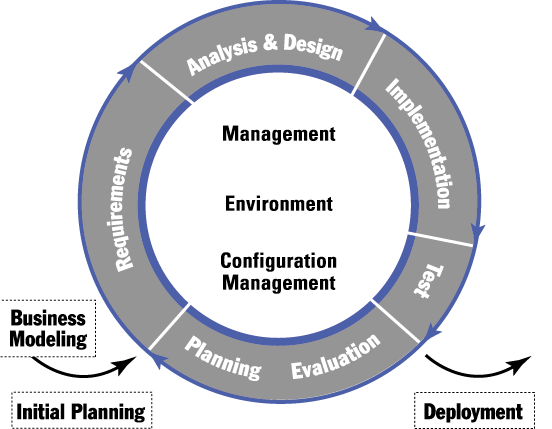


Fig. Flujos de Trabajo del RUP

**El Proceso Unificado de Desarrollo de Software (PUDS)**

El Proceso Unificado de Desarrollo de Software, está basado en la producción y mantenimiento de modelos con un enfoque orientado a objetos y empleo de UML, es un proceso iterativo e incremental, centrado en la arquitectura, dirigido por casos de uso y los riesgos.

El Proceso Unificado proporciona todas las actividades necesarias para el desarrollo de software, utilizando la notación UML para preparar todos los esquemas de un sistema software, dirige trabajadores, provee un marco de trabajo, un conjunto de actividades necesarias para transformar los requisitos del cliente en un producto software.

El proceso unificado de desarrollo se divide en cuatro fases: Inicio, elaboración, construcción y transición, también se estructura en base a flujos de trabajo fundamentales: requisitos análisis, diseño, implementación y prueba.

**Artefacto**

Los sistemas con los que nos encontramos en el mundo real así como en el mundo del Software son muy complejos y complicados a la hora de tratar de entenderlos, para ello una buena manera de salvar este problema es dividirla en *modelos* o fragmentos que describan o abstraigan sus aspectos esenciales y sus maneras de comunicarse.

Los modelos se componen de otros modelos o artefactos.

Los artefactos entonces pueden ser documentos, informes o cualquier archivo ejecutable que se produzca, se manipula o se utiliza.

**Actividad**

Una actividad describe las tareas (pasos de concepción, realización y revisión) que llevan a cabo los trabajadores para crear o modificar los artefactos, junto con las técnicas o guías para ejecutar las tareas, incluyendo quizá el uso de herramientas para ayudar a automatizar una de ellas.

**Fases del Proceso Unificado**

Las fases del Proceso Unificado son el intervalo de tiempo entre dos hitos importantes del proceso durante el cual se cumple un conjunto bien definido de objetivos, se completan artefactos y se toman las decisiones sobre si pasar a la siguiente fase.

Una fase está compuesta por iteraciones (versiones), cada iteración es un mini proyecto que resulta en un incremento del producto final.

Cada fase termina con hito, el cual se determina por la disponibilidad de un conjunto de artefactos (modelos, documentos), permitiendo tomar la decisión de continuar con la siguiente fase.

Las Fases que describe el PUDS son:

* ***Fase de inicio.*** Se pone en marcha el proyecto, se desarrolla una descripción del producto final, se representa el análisis de negocio cuanto será el costo y cuanto traerá beneficio, se delimita el alcance del proyecto y se decide la factibilidad del proyecto.
* ***Fase de Elaboración.*** Se obtienen requerimientos más detallados, realizando un análisis y diseño de alto nivel para establecer la arquitectura base, y se crea el plan para la construcción.
* ***Fase de construcción.*** El objetivo primordial de esta fase es dejar listo un producto software en su versión inicial, a veces llamada “versión beta”. Se desarrolla iterativa e incrementalmente el producto a través de las etapas de análisis, diseño, implementación y prueba, para ser entregado al usuario.
* **Fase de transición.** El objetivo de esta fase es dejar satisfacción en todos los usuarios del software, implantando el producto en su entorno de operación, corrigiendo defectos detectados y efectuando pruebas de aceptación. Al concluir esta fase se decide si los objetivos han sido cumplidos, y si se debe iniciar otro ciclo de desarrollo. Esta fase finaliza con el lanzamiento del producto.

Flujos de Trabajo Fundamental

Un flujo de trabajo muestra todas las actividades que se pueden hacer para producir un conjunto particular de artefactos.

Los 6 flujos de trabajo principales son: Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas y Puesta en Servicio.

A continuación se detallan los flujos de trabajo fundamentales del proceso unificado de desarrollo de software.

**Captura de Requisitos**

El propósito de la captura de requisitos es ayudar al desarrollo del sistema correcto, permite a desarrolladores y clientes ponerse de acuerdo en esa descripción, como el cliente/usuario no es especialista en ordenadores se describe utilizando el lenguaje del cliente. El trabajo de los requisitos se hace fundamentalmente durante el inicio y la elaboración.

**Análisis**

El Análisis proporciona una visión general del sistema, se estudia la descripción de requisitos obtenidos en el flujo de trabajo anterior, refinándolos y estructurándolos con el propósito de alcanzar un mejor entendimiento de los mismos y obtener una descripción de requisitos del sistema centrándose en aspectos como la flexibilidad ante cambios en los requisitos (fácilmente entendible, mantenible) y reusabilidad cuando se construyan sistemas parecidos, el análisis se describe utilizando el lenguaje de los desarrolladores.

**Diseño**

El propósito del diseño es encontrar la forma del sistema que cumpla con los requisitos, teniendo como entrada los resultados del análisis, que es expandido a una solución técnica. Nuevas clases son agregadas para proporcionar una infraestructura técnica: las interfaces de usuario, el manejo de bases de datos para almacenar objetos en una base de datos. La estructura que impone el modelo de análisis deberá conservarse. El diseño es el centro de atención al final de la elaboración y el comienzo de las iteraciones de construcción.

**Implementación**

Se implementa el software requerido por el cliente, las clases de la fase de diseño son convertidas a código actual en un lenguaje de programación orientado a objetos. Se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El propósito de la implementación es el desarrollar la arquitectura y el sistema como un todo. La implementación abarca principalmente la fase de construcción continua durante la fase de transición para tratar defectos tardíos.

**Prueba**

En esta fase se verifica el sistema entero, se realizan pruebas a cada resultado de la implementación para corregir la existencia de errores, al final de la prueba el sistema puede ser entregado al cliente. Se verifican los resultados la implementación, probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales a ser entregadas a terceros. Las pruebas se centran principalmente en las fases de elaboración cuando se inicia la base ejecutable, construcción cuando el sistema esta implementado y transición para la corrección los defectos finales.

### 2.2. UML 2.0

**¿QUÉ ES UML?**

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

***Visualizar:*** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

***Especificar:*** UML permite especificar cuáles son las características de un sistema antes de su construcción.

***Construir:*** A partir de los modelos especifica-dos se pueden construir los sistemas diseñados.

***Documentar:*** Los propios elementos gráficos sirven como documentación del sistema des-arrollado que pueden servir para su futura re-visión.

Un modelo UML está compuesto por tres clases de bloques de construcción:

***Elementos:*** Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).

***Relaciones:*** relacionan los elementos entre sí.

***Diagramas:*** Son colecciones de elementos con sus relaciones.

***PARTE 3***

**“FUNDAMENTOS DE DESARROLLO DE SOFTWARE”**

En esta sección se presenta el proceso de desarrollo de software basado en la metodología del Proceso Unificado de Desarrollo de Software. Se detallaran los flujos de trabajo: captura de requisitos, análisis, diseño e implementación.

### *Identificación de actores y casos de uso*

* ***Actores***

******

******

* ***Casos de uso***

**Display**

### *Priorización de Casos de Uso*

|  |  |  |
| --- | --- | --- |
| **ID** | **CASO DE USO** | **PRIORIDAD** |
| **CU1** | Gestionar Asignatura | MEDIA |
| **CU2** | Gestionar Examen | ALTA |
| **CU3** | Administrar Notas | MEDIA |
| **CU4** | Realizar Examen | ALTA |
| **CU5** | Terminar Examen | MEDIA |
| **CU6** | Generar PDF | MEDIA |

### *Diagrama General de Casos de Uso*



### *Modelo de dominio*

Descripción: C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Lines\BD21390_.gifDescripción: C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE12\Lines\BD21319_.gif

**InsertCoin**

Interface

Requerida

# FLUJO DE TRABAJO ANÁLISIS

* ***Caso de Uso (CU1):*** Gestionar asignatura



***Detalle***

|  |  |
| --- | --- |
| **CASO DE USO** | **CU1:GESTIONAR ASIGNATURA** |
| **Propósito** | Administrar las diferentes asignaturas. |
| **Actores** | Profesor |
| **Actor iniciador** | Profesor |
| **Pre-condición** | Ninguna |
| **Post-condición** | Ninguna |
| **CAMINO BASICO** | |
| **Acciones del Usuario** | **Respuestas del Software** |
| **Guardar**  1.-El usuario introduce los datos.  2.- El usuario pulsa el botón guardar. | 3.-El software verifica los datos.  4.-El software guarda los datos. |
| **Modificar**  1.-El usuario selecciona una asignatura.  2.-El usuario modifica los datos.  3.-El usuario pulsa el botón modificar. | 3.-El software modifica la asignatura seleccionada. |
| **Eliminar**  1.-El usuario selecciona una asignatura.  2.-El usuario pulsa el botón eliminar. | 3.-El software elimina la asignatura seleccionada. |

|  |
| --- |
| **CAMINO ALTERNATIVO** |
| **Guardar**  3.1.- Campo descripción está vacío. |

* ***Caso de Uso (CU2):*** Gestionar Examen



***Detalle***

|  |  |
| --- | --- |
| **CASO DE USO** | **CU2:GESTIONAR EXAMEN** |
| **Propósito** | Diseñar los exámenes para los alumnos. |
| **Actores** | Profesor |
| **Actor iniciador** | Profesor |
| **Pre-condición** | Ninguna |
| **Post-condición** | Transferir Examen |
| **CAMINO BASICO** | |
| **Acciones del Usuario** | **Respuestas del Sistema** |
| **Nuevo**  1.-El usuario selecciona la opción Nuevo.  2.-El usuario introduce los datos y selecciona la materia para el diseño del examen.  3.-El usuario pulsa el botón aceptar. | 2.-El software muestra un formulario.  4.-El software muestra el formulario para el diseño de exámenes. |
| **Pregunta**  **Adicionar Pregunta**  1.-El usuario introduce el puntaje respectivo para la pregunta a realizar.  2.-El usuario pulsa el botón nuevo.  4.-El usuario introduce la/las pregunta.  5.-El usuario pulsa el botón adicionar.  **Modificar Pregunta**  1.-El usuario selecciona una pregunta.  3.-El usuario modifica la pregunta.  4.-Pulsa el botón modificar.  **Eliminar Pregunta**  1.-El usuario selecciona una pregunta.  2.-Pulsa el botón eliminar.  **Cargar Imagen**  1.-El usuario escribe una pregunta.  2.-Pulsa el botón cargar.  4.-Seleciona un archivo imagen.  5.-Pulsa el botón abrir.  **Modificar Imagen**  1.-El usuario pulsa el botón modificar.  3.-El usuario selecciona un archivo imagen.  4.-Pulsa el botón abrir. | 3.-El software limpia los datos.  6.-El software adiciona la/las pregunta al examen y lo muestra.  2.-El software muestra la pregunta seleccionada.  5.-El software guarda la pregunta.  3.-El software elimina la pregunta seleccionada.  3.-El software muestra los archivos de las imágenes a seleccionar.  6.-El software carga y muestra la imagen seleccionada a la pregunta.  2.- El software muestra los archivos de las imágenes.  5.-El software carga y muestra la imagen seleccionada. |
| **Respuesta**  **Adicionar Respuesta**  1.-El usuario introduce la/las respuesta  2.-El usuario selecciona la opción correcta o incorrecta el caso que el examen sea solo de tipo selección.  3.-El usuario selecciona la opción adicionar.  **Modificar Respuesta**  1.-El usuario selecciona una respuesta.  3.-El usuario modifica la respuesta.  4.-Pulsa el botón modificar.  **Eliminar Respuesta**  1.-El usuario selecciona una respuesta  2.-Pulsa el botón eliminar. | 4.-El software adiciona la/las respuesta a la pregunta realizada y la muestra.  2.-El software muestra la respuesta seleccionada.  5.-El software guarda la respuesta.  3.-El software elimina la respuesta seleccionada de la pregunta realizada. |
| **Aplicar**  1.-El usuario añade la ponderación para el examen, selecciona el tipo de examen, introduce el tiempo respectivo e introduce las preguntas con sus respectivas respuestas y puntajes.  2.-Pulsa el botón aplicar. | 3.-El software guarda el examen diseñado. |
| **Guardar**  1.-El usuario diseña el examen.  2.-Pulsa el botón guardar.  4.-El usuario pone un nombre al examen.  6.-Pulsa el botón guardar. | 3.-El software muestra las unidades del computador.  7.-El software guarda el examen. |

|  |
| --- |
| **CAMINO ALTERNATIVO** |
| **Nuevo**  3.1.-El software verifica que se hayan introducido los datos y que no existan campos vacíos en caso que existiera campos vacíos muestra un mensaje “Introducir datos”.  **Adicionar Pregunta**  5.1.- El software verifica que no exista campo vacío al introducir la pregunta en caso que exista campo vacío muestra un mensaje “Introducir Pregunta”.  **Adicionar Respuesta**  3.1.- El software verifica que no exista campo vacío al introducir la pregunta en caso que exista campo vacío muestra un mensaje “Introducir Respuesta”. |

***Diagramas de Comunicación***

* ***Caso de Uso (CU1):*** Gestionar Asignatura

****

* ***Caso de Uso (CU2):*** Gestionar Examen



### *Análisis de clases*

Las siguientes clases interfaz están organizadas alfabéticamente por nombre de Casos de Usos.

* ***Interfaces***
* ***Clase Interfaz:*** FrmDiseño de Exámenes

|  |  |
| --- | --- |
| **Nombre** | FrmDiseño de Exámenes |
| **Propósito** | Permitir Diseñar diferentes tipos de exámenes |
| **Tipo** | Formulario |
| **Atributos** | Jruta,Jpregunta,Jpuntaje,Jtipo,Jexamenes,JpanelPreguntas,JPanelRespuestas,Jeditor,Jcargar,Jnuevo,Jmodificar,Jadicionar,Jcorrecto,Jincorrecto,JrutaExamenes. |
| **Operaciones** | Aplicar,AdicionarPregunta,AdicionarRespuesta,ModificarRespuesta,EliminarRespuesta,CargarImagen,ModificarImagen,Nuevo |
|  |  |

******

* ***Controles***
* ***Clase Control:***Examen

|  |  |
| --- | --- |
| **Nombre** | Examen |
| **Propósito** | Poder guardar los exámenes con sus respectivas preguntas y respuestas. |
| **Entrada** | Descripcion,FechaCreacion,FechaRealizacion,Duración |
| **Salida** | Examen guardado. |
| **Flujo** | 1. Adicionar Pregunta 2. Adicionar Respuesta 3. Guardar() 4. ModificarPregunta() 5. ModificarRespuesta() 6. EliminarPregunta() 7. EliminarRespuesta() 8. SetnodoXML() 9. GetnodoXML() 10. Guardar. |

* ***Clase Control:*** Pregunta

|  |  |
| --- | --- |
| **Nombre** | Pregunta |
| **Propósito** | Permite adicionar preguntas al examen diseñado |
| **Entrada** | Descripcion,puntaje,tipo,dir\_Imagen |
| **Salida** | Datos guardardos. |
| **Flujo** | 1. AdicionarPregunta() 2. ModificarPregunta() 3. EliminarPregunta() 4. SetnodoXML() 5. GetnodoXML() |

* ***Clase Control:*** Respuesta

|  |  |
| --- | --- |
| **Nombre** | Respuesta |
| **Propósito** | Permite adicionar respuestas a las preguntas. |
| **Entrada** | Descripcion,correcto,marcado |
| **Salida** | Datos Guardados |
| **Flujo** | 1. AdicionarRepuesta() 2. ModificarRespuesta() 3. EliminarRespuesta() 4. SetNodoXML() 5. GetnodoXML() |

* ***Clase Control:*** Asignatura

|  |  |
| --- | --- |
| **Nombre** | Asignatura |
| **Propósito** | Permite registrar las asignaturas |
| **Entrada** | Nombre |
| **Salida** | Datos Guardados |
| **Flujo** | 1. Guardar() 2. Modificar() |

* ***Entidades***
* ***Clase Entidad*:** ArchivoXML

|  |  |
| --- | --- |
| **Nombre** | ArchivoXML |
| **Responsabilidad** | Guardar y abrir los examenes |
| **Atributos** | No tiene. |
| **Relaciones** | Examen |