**Diseño de Sistema Registro Académico y Finanzas**

**Universidad Técnica de Concepción**

Nombre del Alumno

Taller de Aplicaciones para Internet en Java

Primer Semestre, 2019

# 1. Índice

[1. Índice 2](#_Toc14273813)

[2. Objetivos 5](#_Toc14273814)

[2.1. Objetivo general 5](#_Toc14273815)

[2.2. Objetivos específicos 5](#_Toc14273816)

[3. Contexto del Sistema 7](#_Toc14273817)

[4. Restricciones del sistema 10](#_Toc14273818)

[4.1. Políticas de la Universidad 10](#_Toc14273819)

[4.2. Políticas de la Empresa Desarrolladora 10](#_Toc14273820)

[4.2.1. Políticas de Usabilidad. 10](#_Toc14273821)

[4.2.2. Otras Especificaciones No Funcionales. 11](#_Toc14273822)

[4.2.3. El software base para la ejecución del software debe tener como mínimo: 12](#_Toc14273823)

[4.2.4. El hardware base para la ejecución del software debe tener como mínimo: 12](#_Toc14273824)

[5. Estudio de Factibilidad 13](#_Toc14273825)

[5.1. Factibilidad técnica 13](#_Toc14273826)

[5.2. Factibilidad operativa 15](#_Toc14273827)

[5.3. Factibilidad económica 15](#_Toc14273828)

[5.3.1 Estudios de costos del sistema 15](#_Toc14273829)

[6. Requerimientos funcionales del sistema 16](#_Toc14273830)

[6.1. Requerimiento funcional 16](#_Toc14273831)

[6.2. Requerimiento funcional 17](#_Toc14273832)

[6.3. Requerimiento funcional 18](#_Toc14273833)

[6.4. Requerimiento funcional 19](#_Toc14273834)

[6.5. Requerimiento funcional 20](#_Toc14273835)

[6.6. Requerimiento funcional 21](#_Toc14273836)

[6.7. Requerimiento funcional 22](#_Toc14273837)

[7. Diagrama Jerárquico del Software 23](#_Toc14273838)

[7.1. Menú Principal 23](#_Toc14273839)

[7.2. Menú Alumno 23](#_Toc14273840)

[7.3. Menú Docente 23](#_Toc14273841)

[7.4. Menú Notas 23](#_Toc14273842)

[7.5. Menú Módulo 23](#_Toc14273843)

[8. Modelo Vista Controlador 24](#_Toc14273844)

[8.1. Descripción 24](#_Toc14273845)

[8.2. Modelo 24](#_Toc14273846)

[8.3. Controlador 25](#_Toc14273847)

[8.4. Vistas 25](#_Toc14273848)

[8.5. Frameworks Laravel 26](#_Toc14273849)

[8.6. Modelo vista controlador de registro académico 28](#_Toc14273850)

[9. Atributos de Seguridad 29](#_Toc14273851)

[10. Atributos de Mantenimiento 30](#_Toc14273852)

[10.1. Descripción de la mantención 30](#_Toc14273853)

[10.2. Tipos de mantención 30](#_Toc14273854)

[10.2.1. Mantenimiento preventivo. 30](#_Toc14273855)

[10.2.2. Mantenimiento predictivo. 30](#_Toc14273856)

[10.2.3. Mantenimiento correctivo. 30](#_Toc14273857)

[10.2.4. Mantenimiento adaptativo 31](#_Toc14273858)

[10.2.5. Mantenimiento evolutivo. 31](#_Toc14273859)

[10.2.6. Mantenimiento perfectivo o funcional. 31](#_Toc14273860)

[10.3. Plan de Mantención 32](#_Toc14273861)

[11. Bases de Datos 33](#_Toc14273862)

[11.1 Diseño de la Base de datos 33](#_Toc14273863)

[11.2 Diseño de tablas 33](#_Toc14273864)

[12. Diagramas UML 34](#_Toc14273865)

[12.1. Descripción de los diagramas 34](#_Toc14273866)

[12.1.1. Diagrama de casos de uso 34](#_Toc14273867)

[12.1.2. Diagrama de secuencias 34](#_Toc14273868)

[12.1.3. Diagrama de colaboración 34](#_Toc14273869)

[12.1.4. Diagrama de clases 34](#_Toc14273870)

[13. Prueba del sistema 35](#_Toc14273871)

[13.1. Descripción de la prueba 35](#_Toc14273872)

[13.1.1. Prueba de la caja negra 35](#_Toc14273873)

[13.1.2. Diseño de la prueba de la caja negra 36](#_Toc14273874)

[14. Implantación del sistema 37](#_Toc14273875)

[14.1. Procedimiento de Implantación del software 37](#_Toc14273876)

[15. Anexos 38](#_Toc14273877)

[15.1. Cuestionario de preguntas 39](#_Toc14273878)

[15.1.1 Cliente de enlace. 39](#_Toc14273879)

[15.1.2 Secretaria registro académico. 39](#_Toc14273880)

[15.3. Gestión de riesgos del sistema 40](#_Toc14273881)

[15.3.1. Descripción de riesgo 40](#_Toc14273882)

[15.3.2. Identificar y planificar el riesgo 41](#_Toc14273883)

[16. Metodologías de desarrollo ágil 48](#_Toc14273884)

[16.1. Metodología ágil Scrum 48](#_Toc14273885)

[16.1.1 Organización del trabajo scrum 48](#_Toc14273886)

[17. Metodología de programación extrema 49](#_Toc14273887)

[17.1. Formato de tarjetas 49](#_Toc14273888)

[18. Glosario 52](#_Toc14273889)

# 2. Objetivos

## 2.1. Objetivo general

Desarrollar un sistema que permita gestionar de forma rápida, eficiente y segura el registro académico y finanzas de los estudiantes de la Universidad Técnica de Concepción (UTDC).

## 2.2. Objetivos específicos

* Facilitar el registro, flujo, procesamiento y almacenamiento de la información, relacionada con las actividades de Registro Académico y Finanzas
* Gestionar los datos de Registro Académico y Finanzas para brindar información confiable utilizando la automatización de procesos
* Asegurar que los datos ingresados y almacenados en la base de datos, reflejan la realidad de cada alumno del establecimiento
* Generar informes automatizados

# 3. Contexto del Sistema

La Universidad Técnica de Concepción (UTDC), necesita desarrollar un producto software que permita automatizar los Procesos de Registro Académico, la que actualmente, es registrada utilizando planillas Excel.

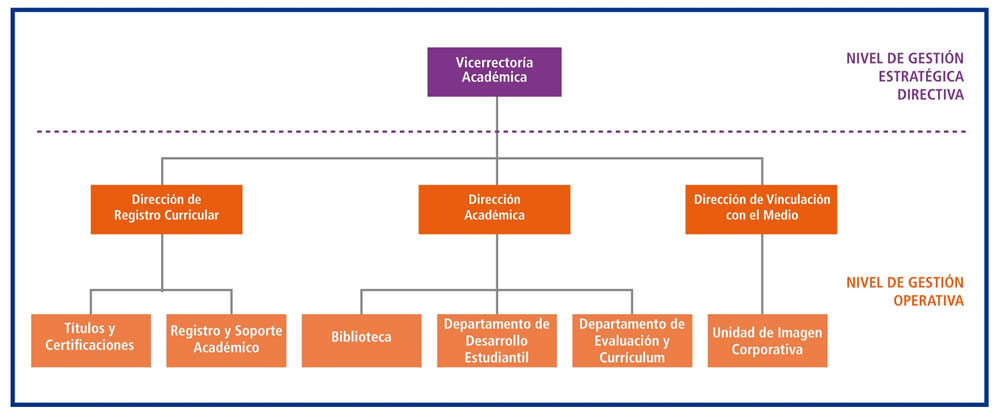
Hasta el momento, la UTDC ha logrado asegurar los datos, almacenando los registros en los computadores de las secretarias, pero existe la inquietud que dichos registros puedan eliminarse o alterarse accidental o premeditadamente, lo cual sería una situación grave para la institución, pues sus consecuencias legales y económicas serian nefastas perjudicando la imagen de la institución.

La UTDC está dispuesta a invertir en el desarrollo de un sistema informático, por lo que encargo a la Dirección de Informática analizar la factibilidad de abordar el desarrollo e implementación del software, la conclusión de este análisis se presenta el este documento (estudio de factibilidades). La Dirección de Informática cuenta con dos servidores para instalar el sistema, estos servidores deben permanecer en la Casa Central de la Universidad para luego abordar la Fase 2 del proyecto, la cual implica la implementación remota del software en las sedes de la Universidad.

Por lo tanto, el desarrollo del software tendrá una duración de un año (fase 1 y 2) en las instalaciones de la UDTC (casa central).

Es preciso indicar que los procesos de registro académico están homologados en cada sede (requisito esencial para mantener la acreditación institucional), teniendo esto en cuenta se proyecta que la implementación de la solución informática a nivel de sedes no debe presentar mayores inconvenientes. Aun así, la Dirección de Informática aconseja la marcha blanca para probar el rendimiento del sistema al integrar las sedes regionales de la Universidad (fase 2)

A continuación, se muestran los respectivos organigramas vinculados al registro académico de la Universidad.



**Figura 1**. Vicerrectoría Académica

|  |
| --- |
| **Figura 2**. Nivel Operativo - Sede |

Los principales usuarios del sistema son:

* Director de sede
* Encargado de registro académico
* Secretarias académicas
* Jefes de carrera
* Encargado de administración y finanzas

Sin embargo, las políticas internas de la UTDC, considera que, para periodos altos de trabajo, las unidades de registro académico y finanzas puedan ser apoyados por funcionarios idóneos en las actividades propias de cada unidad.

# 4. Restricciones del sistema

## 4.1. Políticas de la Universidad

* Política de gestión financiera para la estabilidad y viabilidad institucional
* Política de inversión y financiamiento
* Política de control presupuestario
* Políticas del área de adquisiciones
* Política de financiamiento y cobranza estudiantil
* Política de financiamiento estudiantil
* Política respecto al proceso de renuncia por parte de los estudiantes
* Política respecto a la devolución de aranceles
* Política respecto a la devolución de matrícula
* Política de control de calidad y custodia de pagares
* Política de la dirección de sistemas de información
* Política del sistema de registro curricular
* Sistema de aseguramiento de la calidad

## 4.2. Políticas de la Empresa Desarrolladora

### 4.2.1. Políticas de Usabilidad.

* ISO 13407 proceso de diseño centrado en el usuario para sistemas interactivos
* ISO/TR 16982: Métodos de usabilidad que soportan diseño centrado en el usuario
* ISO 9241-11: Guía de especificaciones y medidas de usabilidad
* ISO 9241-14: Diálogos de menús
* Reglas de usabilidad de Jacob Nielsen

### 4.2.2. Otras Especificaciones No Funcionales.

* El acceso al software será restringido a funcionarios de la Universidad y solamente aquellos que tienen un rol activo en los procesos que se definen en la especificación de requerimientos
* El acceso al software será limitado a la red de tipo corporativa (intranet)
* Los datos serán guardados en dos servidores ubicados en la casa central de la Universidad
* Los datos deben ser validados según su formato antes de guardarse
* La base de datos principal y su backup serán encriptados como medida de seguridad
* La base de datos será respaldada en un segundo servidor al terminar la jornada académica
* 2 cuentas de tipo administrador (encargado de finanzas y encargado de registro académico)
* 1 cuenta restringida para el Director de la Sede con acceso para consultar y generar informes de finanzas y registro académico
* Cada funcionario administrativo dependientes de finanzas y registro académico tendrá acceso restringido al servidor para efectuar consultas y generar informes
* Cada informático de la Sede deberá tener una cuenta de tipo administrador para habilitar procesos administrativos (habilitar usuarios o funciones) previa consulta y autorización del Director de Sede.
* El idioma de comunicación entre el software y el usuario debe ser Español – Chile
* La interfaz de comunicación con el usuario debe tener incorporado técnicas responsivas para facilitar las consultas desde otros medios digitales
* La ejecución del software será independiente del navegador web
* El computador del funcionario no debe ser facilitado a personal no administrativo o alumno de la institución.

### 4.2.3. El software base para la ejecución del software debe tener como mínimo:

* Sistema operativo windows 10 home edition
* Software utilitario office 2016
* Drive o controlador de impresión
* Navegadores Edge, Mozilla, Opera y Chrome

### 4.2.4. El hardware base para la ejecución del software debe tener como mínimo:

* 8 gb de ram (ampliable a 16)
* Procesador de doble núcleo 2.8 ghz como mínimo
* Monitor de 15”
* Tarjeta de red Lan y Wan
* Teclado y mouse

# 5. Estudio de Factibilidad

Para conocer si la UTDC tiene la capacidad técnica, operativa y económica para desarrollar el sistema de registro académico y finanzas, es necesario plasmar en este documento las conclusiones realizadas por la Universidad y la Empresa Desarrolladora del Sistema Informático.

## 5.1. Factibilidad técnica

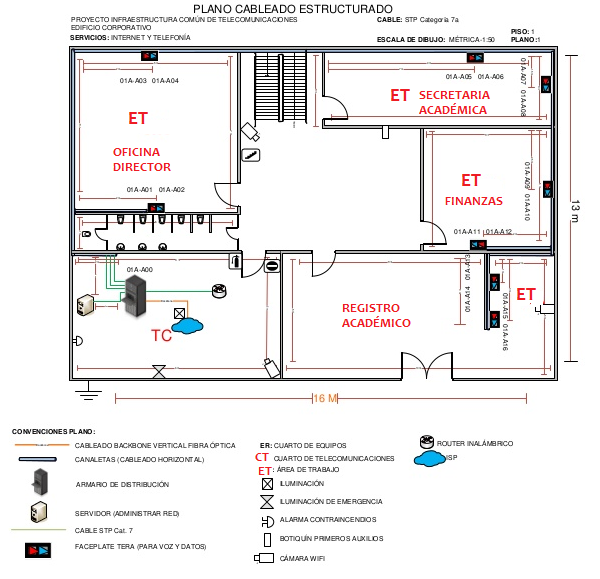
La Universidad cuenta con el equipo necesario para la implementación del sistema informático, todos los equipos de trabajo (PC), tanto del área administrativa, como del resto de la sede, están muy bien equipadas, de acuerdo al siguiente detalle:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EQUIPOS** | **MARCA** | **MODELO** | **CARACTERISTICAS** | **CANT.** |
| Desktop | HP | VALUE | Intel core i5-7400 - 3.0 GhZ c/vPro  Ram: 8GB DDR4 2133 Mhz.  DD 1 Tb 7200 Rpm  Pantalle LED 19,5” HD | 20 |
| Notebook | ACER | ASPIRE 5 | Intel core i5 – 1,6 Ghz  Ram 8GB DDR4  SDD 256 Gb  Pantalla 15,6” | 3 |
| Impresoras | HP | P2055 DN | Laser | 10 |

Los dos servidores ya existen en la Universidad y tienen las siguientes características:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EQUIPOS** | **MARCA** | **MODELO** | **CARACTERISTICAS** | **CANT.** |
| Servidor | DELL | POWER EDGE T30 | Intel Xeon E3-1225 v5 - 3.3 GhZ Quad core  Ram: 8GB DDR4 2133 Mhz.  DD 1 Tb 7200 Rpm Sata 6 Gbps.  Pantalle LED 19,5” HD | 2 |

La red ya está implementada en la Universidad y cuenta con su homologación respectiva, lo que garantiza su uso para el proyecto a desarrollar. A continuación se detalla su ubicación física.



**Figura 3**.Plano cableado estructurado

Asimismo, se instalará en los servidores la base de datos MariaDB MySql, servidor web apache y paquetes relacionados a Php.

## 5.2. Factibilidad operativa

El personal administrativo y directivo de la UTDC se encuentran involucrados y motivados para apoyar el desarrollo del sistema. Por lo que están dispuestos para aportar ideas y experiencia en el diseño e implementación del sistema. La automatización de procesos no presenta inconvenientes al personal de la Universidad ya que existe otros sistemas de gestión automatizados, lo cual se considera una ventaja al momento del auto aprendizaje del nuevo sistema.

Operativamente, la implementación del nuevo sistema es considerado una ventaja estratégica ya que beneficiará las actividades de los funcionarios aportando calidad en sus respectivos puestos de trabajo y vida personal.

## 5.3. Factibilidad económica

Como ya se mencionó en el estudio de factibilidad técnica, la Universidad cuenta con las herramientas necesarias para la implementación del proyecto, así como también con un personal dispuesto a recibir el adiestramiento necesario para manipular la nueva aplicación informática, por lo cual el desarrollo de la propuesta no requerirá de una inversión inicial.

Por el contrario, se puede afirmar que, gracias a la automatización de este proyecto, se reducen y se aligeran las cargas laborales del personal, que normalmente dedican bastante tiempo durante sus jornadas laborales a atender exclusivamente funciones relacionadas con los procesos que originaron este sistema automatizado, y por tal motivo, se puede emplear el tiempo que se ahorran con su puesta en funcionamiento, en otras actividades laborales.

### 5.3.1 Estudios de costos del sistema

* Método de obtención de costos (no aplica para este módulo)

# 6. Requerimientos funcionales del sistema

Requerimientos funcionales solicitados por el cliente:

1.- Ingresar y modificar datos del alumno

2.- Ingresar y modificar datos del docente

3.- Ingresar y modificar notas del alumno

4.- Ingresar y modificar datos del módulo

5.- Ingresar y modificar administrador

6.- Generar una copia impresa del alumno

7.- Generar una copia impresa de los datos del módulo

## 6.1. Requerimiento funcional

|  |  |
| --- | --- |
| **Rf 1.** | **Ingresar y modificar datos del alumno** |
| [Dependencias de procesos] | Login del sistema ,inicio de sesión ingresar, y modificar alumnos |
| Descripción del proceso | Ingresar datos del alumno, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Nombres, apellidos, Rut, edad fecha de nacimiento, país, ciudad, provincia, teléfono, dirección y sexo. |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información del alumno en la BD,    -Salir: permite salir del proceso sin registrar los datos del alumno en la BD.  -Modificar: permite modificar datos de un alumno.  -Imprimir: permite imprimir un documento.  -Eliminar: permite eliminar datos de un alumno. |
| Observaciones | Si alumno ya está registrado en el sistema (se mostrara en pantalla un mensaje que dirá que la persona ya está registrada en el sistema). |

## 6.2. Requerimiento funcional

|  |  |
| --- | --- |
|  | **Ingresar y modificar datos de padres** |
| [Dependencias de procesos] | Login del sistema ,inicio de sesión, ingresar y modificar padres |
| Descripción del proceso | Ingresar datos de los padres, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Nombres, apellidos, Rut, profesión, teléfono, celular, dirección. |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información del alumno en la BD.  -Salir: permite salir del proceso sin registrar los datos de padres en la BD.  -Modificar: permite modificar datos de un padre.  -Eliminar: permite eliminar datos de un padre. |
| Observaciones | Padres ya están registrados en el sistema (se mostrara en pantalla un mensaje que dirá que las personas ya están registradas en el sistema). |

## 6.3. Requerimiento funcional

|  |  |
| --- | --- |
|  | **Ingresar y modificar datos de administrador** |
| [Dependencias de procesos] | Login del sistema ,inicio de sesión, ingresar y modificar usuario |
| Descripción del proceso | Ingresar datos de usuarios, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Nombre, apellidos, Rut, usuario y Password. |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información del alumno en la BD.  -Salir: permite salir del proceso sin registrar los datos de usuarios en la BD.  -Modificar: permite modificar datos de un usuario.  -Eliminar: permite eliminar datos de un usuario. |
| Observaciones | Usuario ya está registrados en el sistema (se mostrara en pantalla un mensaje que dirá que las personas ya están registradas en el sistema). |

## 6.4. Requerimiento funcional

|  |  |
| --- | --- |
|  | **Ingresar o modificar docentes** |
| [Dependencias de procesos] | Login del sistema, inicio de sesión, ingresar y modificar modulo. |
| Descripción del proceso | Ingresar datos de docentes, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Nombres, apellidos, cédula, edad, fecha de nacimiento, país, ciudad, provincia, teléfono, dirección, sexo, profesión, posgrados. |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información del módulo en la BD.  -Salir: permite salir del proceso sin registrar los datos de docente en la BD.  -Modificar: permite modificar datos de un docente.  -Eliminar: permite eliminar datos de un docente. |
| Observaciones | Docente ya está registrado en el sistema (se mostrara en pantalla un mensaje que dirá que el docente ya están registradas en el sistema). |

## 6.5. Requerimiento funcional

|  |  |
| --- | --- |
|  | **Ingresar o modificar Modulo** |
| [Dependencias de procesos] | Login del sistema, inicio de sesión, ingresar y modificar modulo. |
| Descripción del proceso | Ingresar datos de modulo, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Nombre, carrera, semestre, sede, inicio, termino, sala, bloque (Diurno / Vespertino). |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información del módulo en la BD.  -Salir: permite salir del proceso sin registrar los datos de modulo en la BD.  -Imprimir: permite imprimir un documento.  -Modificar: permite modificar datos de un módulo.  -Eliminar: permite eliminar datos de un módulo. |
| Observaciones | Modulo ya está registrado en el sistema (se mostrara en pantalla un mensaje que dirá que el modulo ya están registradas en el sistema). |

## 6.6. Requerimiento funcional

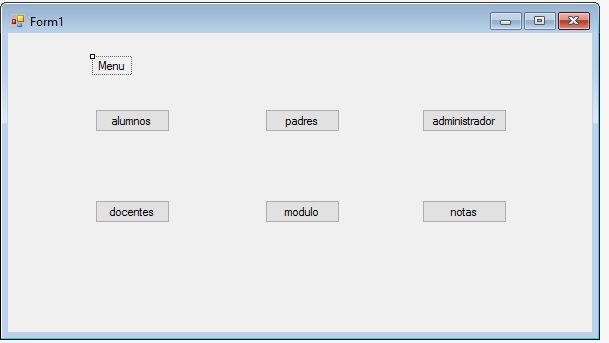
|  |  |
| --- | --- |
|  | **Ingresar y modificar notas** |
| [Dependencias de procesos] | Login del sistema, inicio de sesión, ingresar y modificar modulo. |
| Descripción del proceso | Ingresar notas, para su almacenamiento o modificación en la base de datos. |
| Datos de entrada | Evaluación 1, evaluación 2, evaluación 3, nota examen, examen 1, examen 2, nota Final, estado (aprobado / reprobado) |
| Salida | -Mostrar mensaje de advertencia si falta un dato por ingresar.  -mostrar mensajes para confirmar acciones por parte del usuario. |
| [Importancia] | Alta |
| [Prioridad] | Alta |
| Botones de la interfaz | -Guardar: permite registrar la información de notas en la BD.  -Salir: permite salir del proceso sin registrar los datos de notas en la BD.  -Modificar: permite modificar datos de una nota.  -Eliminar: permite eliminar datos de una nota. |
| Observaciones | Modulo ya está registrado en el sistema (se mostrara en pantalla un mensaje que dirá que el modulo ya están registradas en el sistema). |

## 6.7. Requerimiento funcional

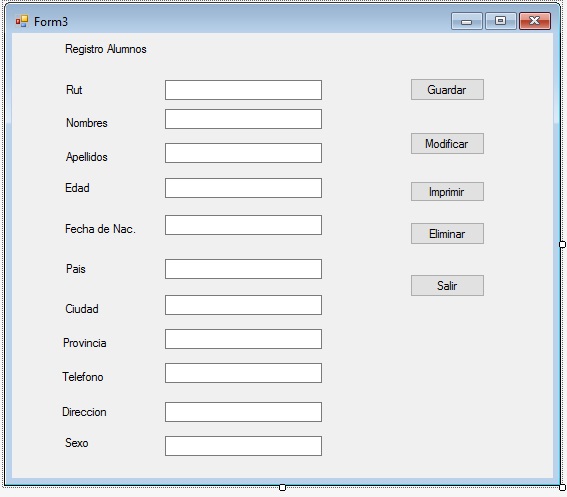
|  |  |
| --- | --- |
|  |  |
| [Dependencias de procesos] |  |
| Descripción del proceso |  |
| Datos de entrada |  |
| Salida |  |
| [Importancia] |  |
| [Prioridad] |  |
| Botones de la interfaz |  |
| Observaciones |  |

# 7. Diagrama Jerárquico del Software

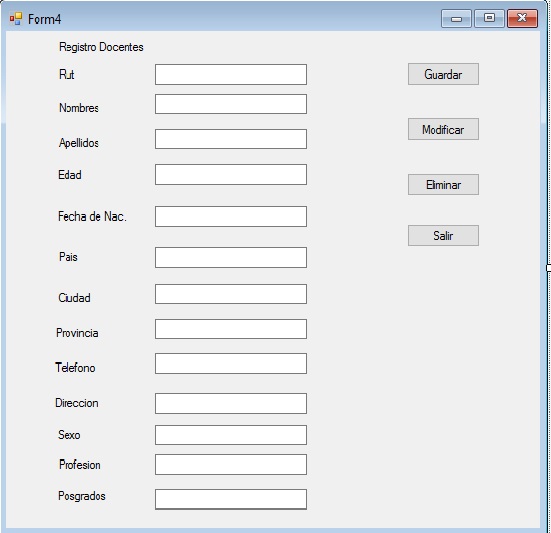
## 7.1. Menú Principal



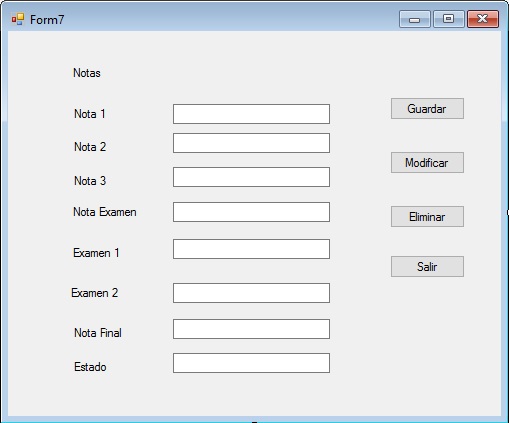
## 7.2. Menú Alumno



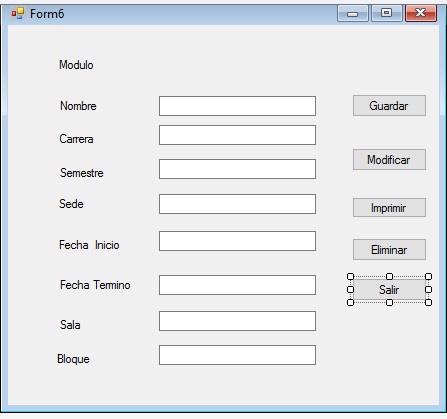
## 7.3. Menú Docente



## 7.4. Menú Notas



## 7.5. Menú Módulo



**7.6. Diagrama de interfaces**

# 8. Modelo Vista Controlador

## 8.1. Descripción

Es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón, estos frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

## 8.2. Modelo

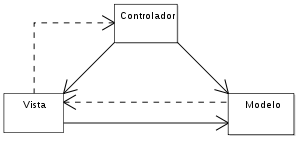
Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador.

## 8.3. Controlador

El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

## 8.4. Vistas

Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.



**Figura 4**. Un diagrama sencillo que muestra la relación entre el modelo, la vista y el controlador. Nota: las líneas sólidas indican una asociación directa, y las punteadas una indirecta.

## 8.5. Frameworks Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.2​

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha de Lanzamiento** | **Versión Php** |
| **1.0** | June 2011 |  |
| **2.0** | September 2011 |  |
| **3.0** | February 22, 2012 |  |
| **3.1** | March 27, 2012 |  |
| **3.2** | May 22, 2012 |  |
| **4.0** | May 28, 2013 | ≥ 5.3.0 |
| **4.1** | December 12, 2013 | ≥ 5.3.0 |
| **4.2** | June 1, 2014 | ≥ 5.4.0 |
| **5.0** | February 4, 2015 | ≥ 5.4.0 |
| **5.1 LTS** | June 9, 2015 | ≥ 5.5.9 |
| **5.2** | December 21, 2015 | ≥ 5.5.9 |
| **5.3** | August 23, 2016 | ≥ 5.6.4 |
| **5.4** | January 24, 2017 | ≥ 5.6.4 |
| **5.5 LTS** | August 30, 2017 | ≥ 7.0.0 |
| **5.6** | February 7, 2018 | ≥ 7.1.3 |
| **5.7** | September 4, 2018 | ≥ 7.1.3 |
| **5.8** | February 26, 2019 | ≥ 7.1.3 |

## 8.6. Modelo vista controlador de registro académico

# 9. Atributos de Seguridad

Los atributos de seguridad son los siguientes:

* Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
* Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales como documentos, archivos y contraseñas.
* Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Intranet, con la intención de consultar y subir información pertinente para cada una de ellas.

**Tabla de requerimientos de seguridad**

|  |  |
| --- | --- |
| **Referencia** | **Nombre** |
| **RS01** |  |
| **RS02** |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 10. Atributos de Mantenimiento

## 10.1. Descripción de la mantención

Es la actividad o conjunto de actividades que se realiza al “Producto Software” posterior a su puesta en marcha; para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno. (ANSI-IEEE)

## 10.2. Tipos de mantención

### **10.2.1. Mantenimiento preventivo.**

Revisión constante del software para detectar posibles problemas en el futuro.

### **10.2.2. Mantenimiento predictivo.**

Evalúa el flujo de ejecución del programa para predecir con certeza el momento en el que se producirá la falla, y así determinar cuándo es adecuado realizar los ajustes correspondientes. ¿Es posible hacer esta actividad?, ¿cómo debemos proceder para hacer una mantención predictiva?

### 10.2.3. Mantenimiento correctivo.

El objetivo es localizar y eliminar los posibles defectos de los programas. Se considera el defecto del sistema como una característica con el potencial de provocar un fallo.

### 10.2.4. Mantenimiento adaptativo

Consiste en la modificar de un programa debido a cambios en el entorno (hardware o software) en el que se ejecuta.

### 10.2.5. Mantenimiento evolutivo.

Es un caso especial donde la adaptación resulta prácticamente obligatoria, ya que de lo contrario el programa quedaría obsoleto con el paso del tiempo.

### 10.2.6. Mantenimiento perfectivo o funcional.

Por distintas razones, el usuario puede solicitar el agregado de nuevas funcionalidades o características no contempladas al momento de la implementación del software.

Se divide en dos:

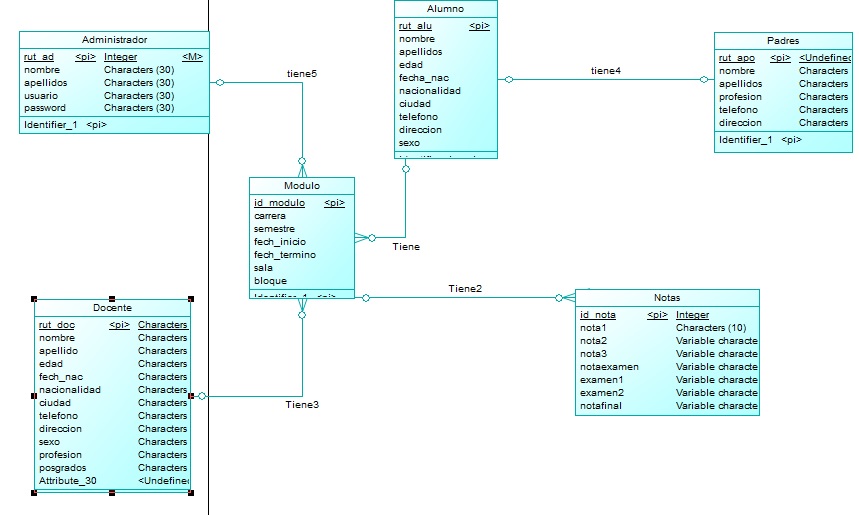
* Mantenimiento de Ampliación: incorporación de nuevas funcionalidades (nuevos requerimientos de usuarios).
* Mantenimiento de Eficiencia: mejora de la eficiencia de ejecución. ¿cómo cuál o cuáles?

## 10.3. Plan de Mantención

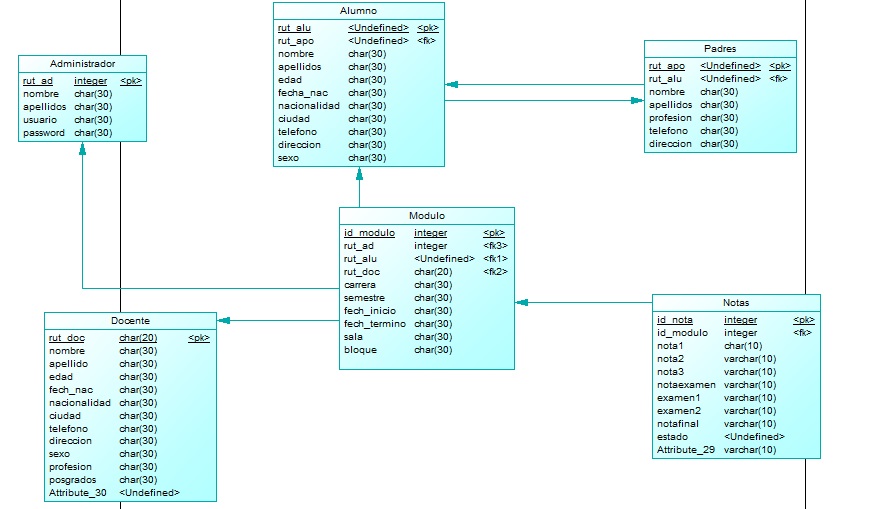
# 11. Bases de Datos

## 11.1 Diseño de la Base de datos

Modelo conceptual



Modelo físico

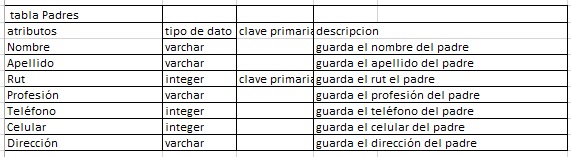
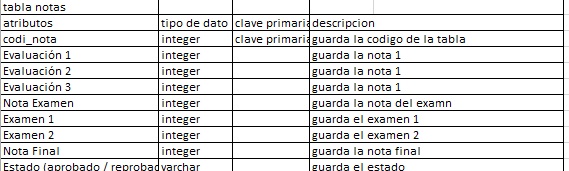
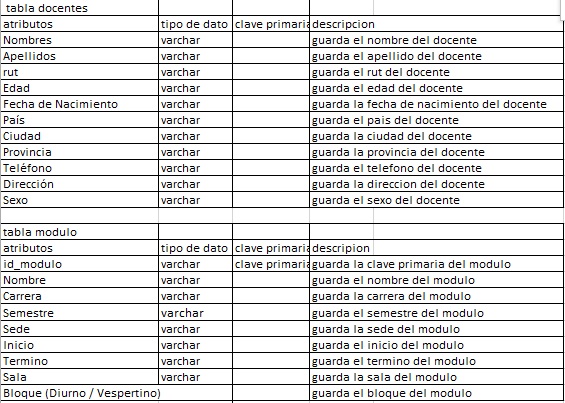
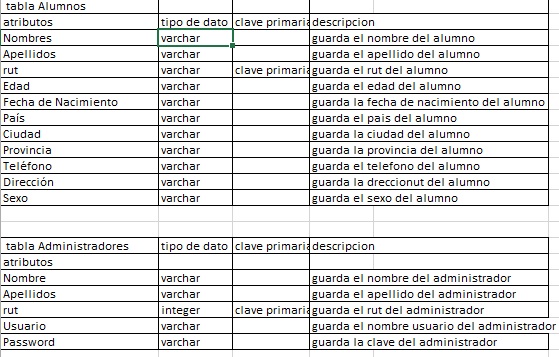


## 11.2 Diseño de tablas

|  |  |  |  |
| --- | --- | --- | --- |
| **Atributo** | **Tipo de dato** | **Clave principal/secundaria** | **Descripción** |
|  |  |  |  |
|  |  |  |  |
|  | | | |

# 

# 12. Diagramas UML



## 12.1. Descripción de los diagramas

### 12.1.1. Diagrama de casos de uso

### 12.1.2. Diagrama de secuencias

### 12.1.3. Diagrama de colaboración

### 12.1.4. Diagrama de clases

# 13. Prueba del sistema

## 13.1. Descripción de la prueba

La fase de pruebas de los sistemas tiene como objetivo la verificación de los diversos procesos, para comprobar si éstos cumplieron con sus requisitos. En esta fase se desarrollaron distintos tipos de pruebas, como son: las funcionales, de usabilidad, de rendimiento.

Las pruebas del sistema se realizan a lo largo del desarrollo del mismo y no simplemente al final. Esto significa plasmar todos los problemas no conocidos y demostrar la perfección de manuales de equipos.

### 13.1.1. Prueba de la caja negra

Las pruebas de la caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

* Las funciones del software son operativas.
* La entrada se acepta de forma correcta.
* Se produce una salida correcta.
* La integridad de la información externa se mantiene.

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

* Funciones incorrectas o ausentes.
* Errores en la interfaz.
* Errores en estructuras de datos o en accesos a bases de datos externas.
* Errores de rendimiento.
* Errores de inicialización y de terminación.

### 13.1.2. Diseño de la prueba de la caja negra

Permite realizar las funciones de buscar, grabar, modificar los datos del alumno en la BD, previa validación de su rut, de acuerdo al formato establecido en la descripción de requerimientos.

|  |  |
| --- | --- |
| Id\_prueba | Ing\_alu\_01 |
| Caso de prueba |  |
| Descripción |  |
| Fecha inicio |  |
| Fecha término |  |
| Datos de entrada |  |
| Resultado esperado |  |
| Resultado obtenido |  |
| Requerimientos de ambiente de prueba |  |
| Otros requerimientos de prueba |  |
| Dependencia con otros casos de pruebas |  |
| Observaciones |  |

# 14. Implantación del sistema

## 14.1. Procedimiento de Implantación del software

# 15. Anexos

## 15.1. Cuestionario de preguntas

### 15.1.1 Cliente de enlace.

* ¿Cuantas personas trabajan el registro académico y quiénes son?
* ¿Con quienes tiene interacción en el proceso de matrícula?
* ¿Qué sistema ocupa para el desarrollo del sistema?
* ¿Tiene un respaldo esta información?
* ¿Se mantiene copia de estos registros?
* ¿Qué problemas presenta el actual método de registro?
* ¿Qué persona necesita que utilicen este sistema?
* ¿Trabajan todos en el mismo piso o nivel?

### 15.1.2 Secretaria registro académico.

* ¿Cómo es el proceso de registro académico? Explique.
* ¿Cuánto tiempo le toma este proceso normalmente?
* ¿Qué otros procesos Ud., tiene que realizar con el alumno nuevo?
* ¿Cree Ud. que este proceso necesite que se mejore? ¿Como?
* ¿Tiene experiencia con uso de aplicaciones para computadores?
* ¿Cree Ud. que una aplicación computacional mejorará su trabajo?
* Si eso ocurre ¿Estaría dispuesto a apoyar con consejos, opiniones e ideas para mejorarla?
* ¿Estaría dispuesta a recibir capacitación de este sistema?

## 15.3. Gestión de riesgos del sistema

### 15.3.1. Descripción de riesgo

Como parte de la gestión del riesgo, es preciso definir una política de riesgos del proyecto con objeto de mantener los riesgos inherentes dentro de límites definidos y aceptados. Esta política debe estar de acuerdo con la política de riesgos de la organización, de manera que la identificación y el tratamiento de los riesgos sean consistente y homogéneo en todos los proyectos.

Se identificó las siguientes categorías de riesgos:

* Nivel operativo de la sede.
* Nivel operativo de la empresa desarrolladora
* Costos del proyecto.
* Niveles de Comunicación.
* Elementos Técnicos.
* Presupuestos.

### 15.3.2. Identificar y planificar el riesgo

|  |
| --- |
| **Nivel operativo de la sede** |

|  |  |
| --- | --- |
| **Riesgo** | **Solución** |
| La contraparte técnica (encargado de registro académico) abandona la Universidad. | La Universidad contrata nuevo personal o asciende una secretaria académica |
| La contraparte técnica (encargado de finanzas) abandona la Universidad. | La Universidad contrata nuevo personal |
| Funcionarios de la Universidad pierden el interés de sus deberes con el proyecto, ignorando los procesos de comunicación establecidos para la retroalimentación entre el nivel operativo y la empresa de desarrollo. | El equipo muestra los avances del sistema a los funcionarios, para que se motiven nuevamente. |
| La contraparte técnica (encargado de registro académico) abandona la Universidad. | La Universidad designa una nueva persona que esté al tanto del proyecto y sus alcances |
| Existen conflictos entre funcionarios del nivel operativo de la Universidad, con respecto a procedimientos administrativos de registro académico. | La Universidad debe solucionar el conflicto, para no afectar el trabajo de desarrollo. |
| Existen conflictos entre funcionarios del nivel operativo de la Universidad, con respecto a procedimientos de finanzas. | La Universidad debe solucionar el conflicto, para no afectar el trabajo de desarrollo. |
| Falta de compromiso de los niveles operativos involucrados en el proyecto. | La Universidad debe solucionar el conflicto, para no afectar el trabajo de desarrollo. |
| Los funcionarios de la Universidad tienen una actitud negativa y les gustaría ver fracasar el proyecto. | El equipo deberá mostrar los avances a los funcionarios negativos, para volverlos a encantar con el sistema. |
| Las contribuciones de los funcionarios son de baja calidad o irrelevante. | La Universidad debe exigir a su personal sobre la importancia de su labor |
| Falta de compromiso del Director de Sede. | La Casa Central debe solucionar este problema, para no perjudicar el sistema. |
| La sede sufre de pérdidas financieras que paralizan una sección de la Universidad | El presupuesto está financiado totalmente por la Casa central. |

|  |  |
| --- | --- |
| **Nivel operativo de la Empresa desarrolladora** | |
| **Riesgo** | **Solución** |
| El equipo de programadores incorpora sus propios requerimientos funcionales | Líder debe comunicarse diariamente con su equipo, las mejoras a los requerimientos. |
| El equipo de programadores no planifica las pruebas de sistemas | Líder debe estar constantemente controlando las pruebas. |
| El equipo de programadores ha perdido la comunicación | Se debe efectuar las reuniones diarias, según método Scrum. |
| El analista líder del proyecto abandona la empresa | Se debe contratar personal con experiencia en métodos ágiles, liderazgo y manejo habilidades blandas. |
| Rotación conduce a retrasos y aumento de los costos. | Tener respaldo de personal con experiencia ante esta situación. |
| El personal no tiene las habilidades para el desarrollo del software | Tener respaldo de personal con experiencia ante esta situación. |
| El personal no recibió el entrenamiento adecuado. | Tener respaldo de personal con experiencia ante esta situación o apoyarlo en su trabajo. |
| No existe personal suficiente. | Tener respaldo de personal con experiencia ante esta situación. |
| No todo el personal conoce el desarrollo completo del sistema. | La metodología Scrum permite que esto no pase, al mantener constantemente comunicación con todo el equipo. |

|  |  |
| --- | --- |
| **Costo del proyecto** | |
| **Riesgo** | **Solución** |
| Las estimaciones de costos del proyecto son inexactas. | El presupuesto se debe hacer por etapas del desarrollo. |
| El tiempo estimado del proyecto, no se ajusta a la realidad. | Se debe cumplir los plazos en las fechas establecidas, para no alargar el proyecto. |
| El personal contratado es insuficiente. | Tener respaldo de personal con experiencia ante esta situación. |
| Existen requerimientos que no se desarrollaron en el diseño. | Se debe cumplir los requerimientos por etapas y con entrega al cliente cada vez.. |
| El cliente se niega a pagar, costos que no estaban en el contrato. | Presupuesto tiene respaldo de la Casa Central de la Universidad. |
| El producto no se entrega en la fecha señalada. | Se debe cumplir los plazos en las fechas establecidas, rigurosamente. |
| El programador se retira del proyecto. | Tener respaldo de personal con experiencia ante esta situación o apoyarlo en su trabajo. |
| El cliente de enlace abandona la empresa. | La Universidad designa una nueva persona que esté al tanto del proyecto y sus alcances |
| El analista se enferma gravemente. | Tener respaldo de personal con experiencia ante esta situación. |
| Un integrante del equipo muere. | Tener respaldo de personal con experiencia ante esta situación. |

|  |  |
| --- | --- |
| **Comunicación** | |
| **Riesgo** | **Solución** |
| Los requisitos son mal interpretados por el equipo del proyecto se producirá un desfase entre las expectativas, demandas y el trabajo en su conjunto. | Estandarizar la comunicación entre las partes y dejarlas estipuladas en el contrato. |
| Los canales de comunicación no son ocupados por las partes interesadas | Establecer los canales de comunicación y dejarlas estipuladas en el contrato. |
| El cliente de enlace no participa de las reuniones de desarrollo. | Exigir a la Universidad su participación o pedir el cambio de ella. |
| El equipo no se entiende con el cliente y piden su cambio. | Se debe tratar de solucionar de inmediato el problema, a través de reuniones o solicitar el cambio de la persona. . |
| El cliente de enlace exige requisitos no incorporadas en el contrato | Se debe cumplir con el contrato escrito, por parte de la Universidad. Cualquier requerimiento adicional tiene otro costo. |

|  |  |
| --- | --- |
| **Técnicos** | |
| **Riesgo** | **Solución** |
| Los recursos técnicos no tienen interfaces estándar. | Estandarizar la interfaz de los equipos adquiridos y revisar los existentes. |
| Los recursos técnicos tienen vulnerabilidades de seguridad. | Establecer estándar de seguridad |
| Los recursos técnicos fallan después de un corto período de tiempo. | Exigir cumplimiento de las garantías |
| No hay documentación de apoyo técnico. | Crear documento de apoyo. |
| La interfaz no gusta a los usuarios. | Presentar interfaz para su aprobación al cumplir este requisito no funcional. |
| No se realizan las mejoras solicitadas por los usuarios. | Cumplir mejoras una vez solicitadas por los usuarios |
| La tecnología usada pierde la licencia. | Renovar licencia de los equipos. |

|  |  |
| --- | --- |
| **Presupuesto** | |
| **Riesgo** | **Solución** |
| El presupuesto entregado está demasiado alto. | Estandarizar la interfaz de los equipos adquiridos y revisar los existentes. |
| El presupuesto entregado está bajo presupuesto | Establecer estándar de seguridad |
| El tiempo estimado es demasiado corto o largo | Exigir cumplimiento de las garantías |
| Se contrató poco personal. | Crear documento de apoyo. |
| El personal presento licencias. | Presentar interfaz para su aprobación al cumplir este requisito no funcional. |
| El personal abandonó el proyecto y hay que contratar personal nuevo. | Cumplir mejoras una vez solicitadas por los usuarios |
| No se consideró equipos periféricos, necesarios para el proyecto. | Renovar licencia de los equipos. |

# 16. Metodologías de desarrollo ágil

## 16.1. Metodología ágil Scrum

### 16.1.1 Organización del trabajo scrum

# 17. Metodología de programación extrema

## 17.1. Formato de tarjetas

Historia del proyecto: tiene por objetivo proporcionar información básica del proyecto que se está abordando.

|  |  |
| --- | --- |
| **HISTORIA DEL PROYECTO** | |
| **Número de Proyecto:** permite identificar el proyecto informático. | **Usuario:** usuarios que utilizarán las funcionalidades del software. |
| **Nombre del Proyecto:** describe de manera general el propósito del software. | |
| **Fecha de Inicio:** inicio del proyecto. | **Fecha de término:** fin del proyecto. |
| **Prioridad del Negocio:** grado de importancia que el cliente asigna al proyecto informático.  **[1-3] 1:baja – 2: Media – 3: Alta** | **Riesgo del Desarrollo:** valor de complejidad que el proyecto representa para el equipo de desarrollo.  **[1-3] 1:baja – 2: Media – 3: Alta** |
| **Equipo de Desarrollo:**   * Líder: persona encargada de dirigir el proyecto informático. * Programadores * Otros Cargos | |
| **Cliente:** persona vinculada a la empresa (contraparte técnica) | |
| **Descripción:** describe de manera detallada las funcionalidades principales del software. | |
| **Observaciones:** información relevante del proyecto (objetivos generales, específicos, alcances, cuestiones de diseño, acuerdos entre el cliente y el equipo de desarrollo, etc) | |

Historia de usuario: tiene por objetivo proporcionar información relevante para la etapa de programación de cada uno de los requerimientos funcionales del sistema.

|  |  |
| --- | --- |
| **HISTORIA DEL USUARIO** | |
| **Número de Proyecto:** Permite identificar el proyecto informático. | **Usuario:** Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario. |
| **Número de Historia:** Permite identificar la historia del usuario. |
| **Nombre de la Historia:** Describe de manera general la historia o característica a desarrollar. | |
| **Tipo de Tarea:**  análisis – diseño – programación | |
| **Fecha de Inicio:** Inicio de la historia. | **Fecha de término:** Fin de la historia. |
| **Prioridad en Negocio:** Grado de importancia que el cliente asigna a la historia del usuario.  [**1-3] 1:baja – 2: Media – 3: Alta** | **Riesgo en Desarrollo:** Valor de complejidad que la historia del usuario representa para el equipo de desarrollo.  [**1-3] 1:baja – 2: Media – 3: Alta** |
| **Puntos Estimados:** Número de semanas que se necesitará para el desarrollo de una historia de usuario.  **[1 – N] semanas** | **Iteración Asignada:** Número de iteración, en que el cliente desea que se implemente una historia de usuario.  **[1 – N] semanas** |
| **Programador:** Persona encargada de programar la historia de usuario. | |
| **Descripción:** Información detallada de la historia del usuario. | |
| **Observaciones:** Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de la historia del usuario. | |

Prueba de aceptación: tiene por objetivo proporcionar información de cómo abordar la etapa de prueba de cada requerimiento funcional del sistema.

|  |  |
| --- | --- |
| **PRUEBA DE ACEPTACIÓN** | |
| **Código:** permite identificar la prueba de aceptación. | **Número de Proyecto:** permite identificar el proyecto informático. |
| **Número de Historia:** permite identificar la historia del usuario que está siendo sometida a pruebas. |
| **Fecha de Inicio:** inicio de la prueba. | **Fecha de término:** fin de la prueba. |
| **Tester:** persona encargada de realizar la prueba. | **Metodología de Prueba:** nombre de la metodología. |
| **Tipo de Test**: nombre del test que se aplica a la historia del usuario. |
| **Condiciones de Ejecución:** condiciones previas que deben cumplirse para realizar la prueba de aceptación. | |
| **Entrada de datos/Pasos de Ejecución de la prueba:** pasos que siguen los usuarios para probar la funcionalidad de la tarea. | |
| **Resultado Esperado:** respuesta del sistema que el cliente espera, después de haber ejecutado una funcionalidad. | |
| **Observaciones de la evaluación de la prueba:** nivel de satisfacción del cliente sobre la respuesta del sistema.  Los niveles son: aprobada y no aprobada. | |

# 18. Glosario