< RQSRSG9v3>-<ICADE VIRTUAL>

Especificación de Requerimientos del Software (SRS)

Versión 2.0

Trujillo, 2011

Tabla de Contenidos

1. Introducción 3

1.1 Propósito 3

1.2 Alcance 3

1.3 Definiciones, Acrónimos y abreviaturas 3

1.4 Referencias 3

1.5 Generalidades 3

2. Descripción Global 4

2.1 Reporte del Modelo de Casos de Uso 4

2.2 Consideraciones y Dependencias 4

3. Requerimientos Específicos 5

3.1 Funcionalidad 5

3.2 Facilidad de Uso 5

3.3 Confiabilidad 5

3.4 Desempeño 6

3.5 Facilidad de Soporte 6

3.6 Restricciones de Diseño 7

3.6.1 R**estricciones sobre la Arquitectura** 7

**3.6.2** **Restricciones sobre los Componentes** 7

**3.6.3** **Interfaces de Usuarios** 7

**3.6.4** **Interfaces de Hardware** 7

3.6.5 Interfaces de Software 7

3.6.6 Interfaces de Comunicación 8

3.7 Documentación en Línea y Requerimientos de Ayuda del Sistema 8

3.8 Requerimientos de Licencia 8

3.9 Metodología de Desarrollo 8

3.10 Componentes Adquiridos 8

- Adquisición de Licencia para IBM Rational Software Architect, IBM Requisite PRO 8

- Adquisición de Licencia para el uso de SQL SERVER 2012. 8

- Adquisición de Licencia para el uso de VISUAL.NET 2012. 8

3.11 Otros Estándares Aplicables 8

Especificación de Requerimientos del Software (SRS)

# Introducción

Los requerimientos de software son muy importantes dentro de un proyecto ya que nos indican cuales son los recursos que se utilizaran en su desarrollo es por esto que hemos realizado listados de todos los requerimientos funcionales y no funcionales que nos permitan conocer las necesidades de nuestro producto y el ambiente en que se implantará.

## Propósito

Lograr que la comunicación entre las áreas este basada en requerimientos claramente definidos de manera que puedan ser filtrados y monitoreados haciendo posible las evaluaciones objetivas de funcionalidad y performance; pero sobre todo que los errores sean más fácilmente detectables, logrando así que el sistema satisfaga las necesidades del usuario.

## Alcance

Este proyecto servirá de modelo precedente para comenzar a sistematizar todas las áreas relacionadas con las ventas online en lo que se refiere a proveer a los comerciantes minoristas que carecen de tiempo para hacer sus compras las cuales las podría realizar desde la comodidad de su hogar y no solo se tendrá en cuenta las áreas administrativas sino que los futuros software fijaran su objetivo en otros ambientes de la empresa.

## Definiciones, Acrónimos y abreviaturas

Ver Glosario de Términos

## Referencias

Libro: UML GOTA A GOTA, Martin Fowler y Kendall Scott, Primera Edición., Addison Wesley

http://technet.microsoft.com/es-es/library/aa998776(EXCHG.65).aspx

## Generalidades

Empezaremos con modelado los casos de uso del sistema, luego esto se procederá a realizar los diagramas necesarios para la descripción de los mismos, se diseñarán las interfaces de usuario y se especificarán los requerimientos necesarios del software.

# Descripción Global

## Reporte del Modelo de Casos de Uso

Lista de los actores de los casos de uso

* Participante
* Administrador

Lista de los casos de uso

* Gestionar participante
* Gestionar administrador
* Autenticar usuario
* Gestionar profesión
* Gestionar categoría
* Gestionar diplomado
* Gestionar seminario
* Gestionar tema
* Gestionar material multimedia
* Gestionar evento seminario
* Registrar matricula
* Registrar pago
* Visualizar material multimedia
* Descargar material multimedia

## Consideraciones y Dependencias

* Identificar los objetivos de la recopilación de los datos.
* Documentar la infraestructura de la aplicación.
* Establecer una vista del entorno de desarrollo y los procesos de avance de las aplicaciones.
* Inventario de las aplicaciones que se van generando
* Inicialmente nuestro sistema será independiente pero se diseñará con características de adaptabilidad para integraciones futuras.
* El producto de software deberá operar 12 horas al día, 365 días al año.
* Aplicaremos la metodología RUP para el desarrollo de sistema, contando con herramientas como Rational Software Architect, para el diseño de la arquitectura de software.

# Requerimientos Específicos

* + Identificación y documentación de las necesidades de los usuarios y StakeHolders.
  + Análisis y Validación del documento de requerimientos para asegurar consistencia y viabilidad.
  + Refinar los requerimientos del sistema Hardware y Software.

## Funcionalidad

* + El software ICADE VIRTUAL podrá ser interoperable con los demás sistemas de la empresa, tanto en servidores (Windows Server 2012 o 2010) como estaciones de trabajo (Windows XP Professional, Windows 7 y Windows 8).

## Facilidad de Uso

* + El estado del sistema es siempre visible
  + Se utiliza el lenguaje de los usuarios
  + El usuario tiene control y libertad
  + Hay consistencia y se siguen estándares
  + Existe prevención de errores
  + Se minimiza la carga de la memoria del usuario
  + Existe flexibilidad y eficiencia de uso
  + Al utilizar la ayuda, se reconocen, diagnostican, y se recuperan de los errores
  + Existe ayuda y documentación

## Confiabilidad

* + Disponibilidad: El software estará disponible en un 80%; en cuanto a su mantenimiento será realizado por el Administrador del sistema y lo realizará deshabilitando puntos específicos cuando se encuentre realizando las modificaciones.
  + Seleccionar las áreas de mejora y reducción de las exigencias de mantenimiento realizando un cálculo rápido entre fallas continuas, lo que demorará aproximadamente 1 día, debido a que se realizarán reportes de mantenimiento continuos.
  + De mostrar fallas más severas el software será programado para hacer un rescate de toda la información contenida hasta el momento de la interrupción de las acciones del sistema.

## Desempeño

* + El desempeño es un aspecto fundamental en una aplicación, ya que si ésta no responde en el debido tiempo, se pueden perder clientes, o dañar la imagen ante los usuarios por eso ICADE VIRTUAL ofrece un buen desempeño en el software.
  + No importa el número de usuarios, ICADE VIRTUAL está diseñado para responder a las peticiones sin consumir muchos recursos.
  + Se planeara una prueba de desempeño para encontrar la carga pico y probar la aplicación contra esta carga.
  + ICADE VIRTUAL mide el rendimiento de procesamiento, en la cantidad de peticiones servidas por unidad de tiempo.
  + El tiempo de respuesta de una aplicación en ICADE VIRTUAL empieza desde que el usuario inicia una petición hasta que el resultado es visualizado.
  + Para ICADE VIRTUAL las pruebas de desempeño son una actividad importante del desarrollo de una aplicación.

## Facilidad de Soporte

* + Ejecución eficiente; uso mínimo de recursos de tiempo y memoria.
  + Fácil de leer, comprender, fácil de depurar y mantener.
  + Interfaz de usuario independiente de las funciones de cálculo.
  + Las variables son descritas de manera estándar y convencional.
  + Los formularios están diseñados con controles cuya identificación también es estándar
  + Las funciones y los procedimientos contienen una breve descripción del problema que se resuelve.

## Restricciones de Diseño

### R**estricciones sobre la Arquitectura**

#### Los casos de uso de agrupan en subsistemas

#### Subsistema Administración

#### Subsistema Matriculas

#### Subsistema Seguridad y Autenticación

### **Restricciones sobre los Componentes**

#### **Aplicación de Patrones de Diseño**

Los componentes se aplicarán siguiendo estándares de diseño para optimización de sus funciones. Destacan los siguientes:

* + Uso del Patrón Bridge para separar la interface de la implementación.
  + Uso del Patrón Factory para crear familias de clases.
  + Uso del Patrón DAO para poder insertar, actualizar, eliminar, buscar datos dentro de una Base de Datos

## Interfaces

### **Interfaces de Usuarios**

Se mostrara en le otra documento

### **Interfaces de Hardware**

El software dará soporte a diferentes periféricos de entrada y salida en los que se incluyen:

* Medios de almacenamiento externo:
* Discos compactos escribibles y re-escribibles
* Memorias con interface USB
* Impresoras:
* El sistema será compatibles con impresoras de diferentes marcas y de inyección de tinta para los reportes.

### Interfaces de Software

### Interfaces de Comunicación

El sistema estará implementado para una arquitectura de red local la cual será conectada al servidor de la compañía para solicitar algunos datos que ya se encuentran en BD

## Documentación en Línea y Requerimientos de Ayuda del Sistema

La documentación de ayuda brindará al usuario un manual que contemple los eventuales problemas y soluciones de instalación y configuración. Además de ayuda para el manejo de los módulos implementados.

## Requerimientos de Licencia

Para el diseño de la arquitectura de software y de los procesos se ha adquirido el software IBM Rational Software Architect, IBM Requisi PRO SQLSERVER 2012 Y VISUAL STUDIO.NET 2012

## Metodología de Desarrollo

La metodología de desarrollo a utilizar es RUP.

## Componentes Adquiridos

## Adquisición de Licencia para el uso de IBM Rational Software Architect, IBM Requisi PRO

## Adquisición de Licencia para el uso de SQL SERVER 2012.

## Adquisición de Licencia para el uso de VISUAL.NET 2012.

## Otros Estándares Aplicables

No definido.

Historia de las Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| <30/06/2013> | <2.0> | <se detallan los requerimientos recolectados de los interesados de nuestro sistema web> | <Grupo 9> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |