|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cod. Software** | AP0155 | **Nombre**  **Software** | SIGED | |
| **Creación** | 22/01/2018 | **Creador** | Juan Perez | **Versión** |
| **Modificación** | 25/01/2018 | **Modificador** | Carlos Chavez | **1.x** |

# INTRODUCCIÓN

El siguiente formato provee una visión general de la Arquitectura de Software, usando un número de diferentes vistas de arquitectura para describir diferentes aspectos del Software, es decir captura y cubre las decisiones tomadas para elegir la arquitectura bajo la cual se construirá el software.

Por ejemplo:

El presente documento provee una visión general de la arquitectura del Software X, usando diferentes vistas para apreciar los diferentes aspectos del Software, las cuales están basados en los estándares del Proceso Unificado de Desarrollo de Software (RUP) y utilizando el Lenguaje de Modelamiento Unificado (UML).

# DEFINICIONES Y ABREVIACIONES

[Esta sección brinda la definición de aquellos términos y abreviaciones requeridas para interpretar adecuadamente el contenido de este documento.]

# REFERENCIAS

Indicar la documentación referenciada para la elaboración del documento.

Por ejemplo:

* Procesos de Ingeniería de la Metodología Integrada de Osinergmin.
* IBM Corp. "Rational Unified Process® Versión 1.2" Rational IBM 2005.

# ALCANCE

Por ejemplo:

Detalla la arquitectura propuesta por el equipo de desarrollo y contempla la interrelación con los otros subsistemas, modelos de dominio y datos, además de los diagramas de diseño necesario para comprender el comportamiento de los componentes.

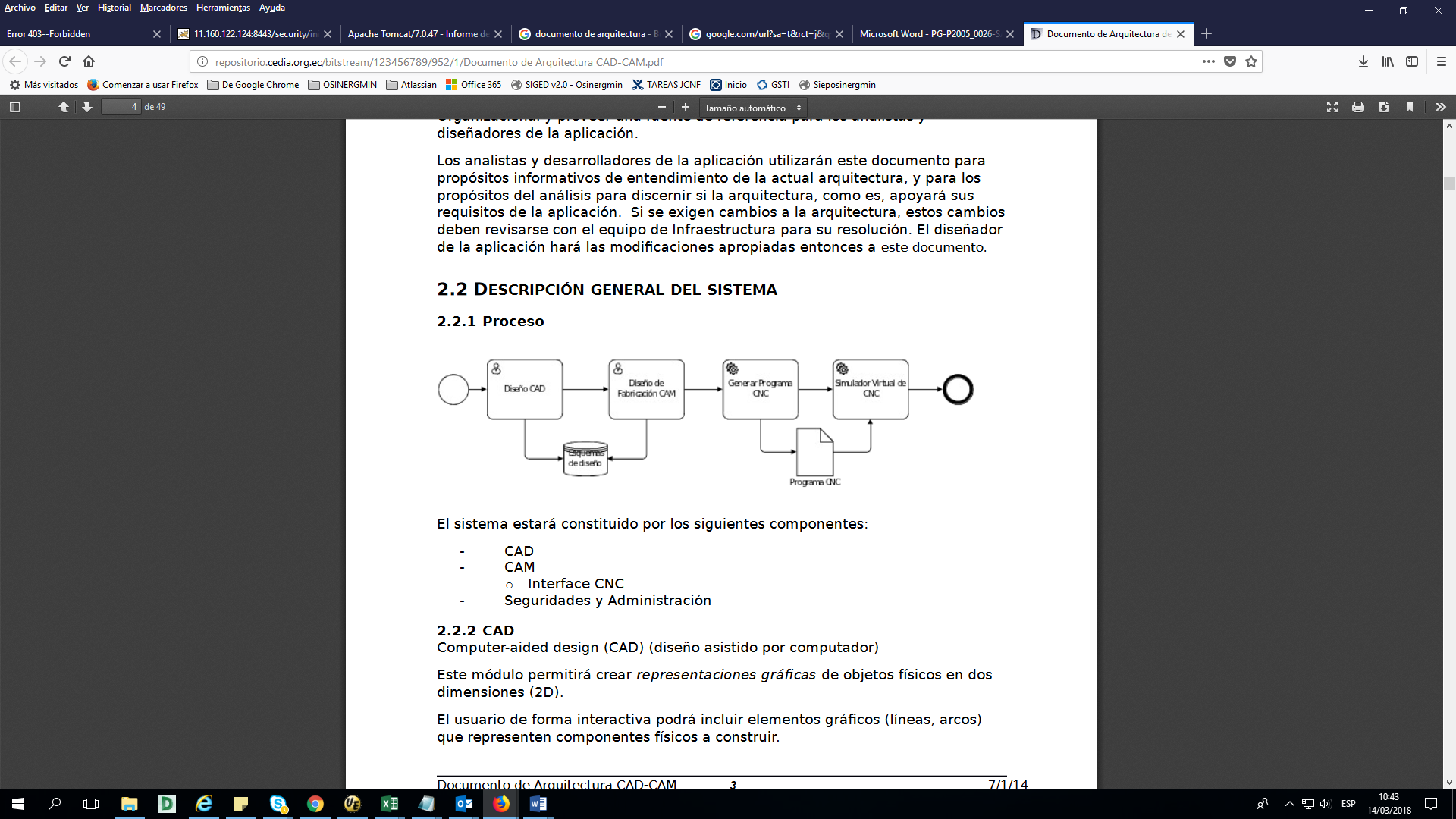
# METAS Y RESTRICCIONES DE LA ARQUITECTURA

Esta sección describe los requerimientos de software y objetivos que tienen algún significativo impacto sobre la arquitectura; por ejemplo: seguridad, privacidad de uso del producto, portabilidad, distribución y reuso. Esto también captura las restricciones especiales que quizás apliquen en la: Estrategia de diseño e implementación, herramientas de desarrollo, estructura del equipo, cronograma, código legal y otros.

# DESCRIPCIÓN GENERAL DEL SISTEMA

## PROCESO

*Ejemplo:*



*El Software estará constituido por los siguientes componentes:*

* ***CAD (Computer-aided design)***

*Este módulo permitirá crear representaciones gráficas de objetos físicos en dos dimensiones (2D).*

*El usuario de forma interactiva podrá incluir elementos gráficos (líneas, arcos) que representen componentes físicos a construir.*

*En la primera etapa se permitirá crear dibujos en 2D, pero deberá contemplarse para una segunda etapa la construcción de dibujos en 3D.*

*Estos módulos tendrán las siguientes características*

* *Estos dibujos estarán basados en vectores*
* *Permitirá establecer un orden entre elementos gráficos creados*
* *Incluirá Información de las Geometrías*
* *Incluirá posiciones, dimensiones, unidades.*
* *Documentación descriptiva de cada componente*
* *Integración con CAM*
* ***CAM (Computer-Aided Manufacturing)***

*Se encargará generar los programas en Código CNC para las máquinas a partir de las representaciones gráficas producidas en el CAD.*

*Estos módulos tendrán las siguientes características:*

* *Integración con el módulo CAD*
* *Definir un plan de fabricación, herramientas*
* *Generar los programas Unidad de control numérico (CNC): Creará*
* *instrucciones detalladas (G-código)*
* *Gestionar de archivos para intercambio con el simulador virtual de*
* *máquinas CNC*
* *Para la comunicación entre componentes físicos en el caso del*
* *“Simulador virtual de CNC” se realizará a través de intercambio de*
* *archivos.*
* *Se podrá exportar los Esquemas de Diseño a través de formatos XML,*
* *de igual manera se podrá realizar importación de Diseños a través de*
* *formatos de XML*
* ***Seguridades y Administración***

***…..***

# DEFINICIÓN DE LA ARQUITECTURA

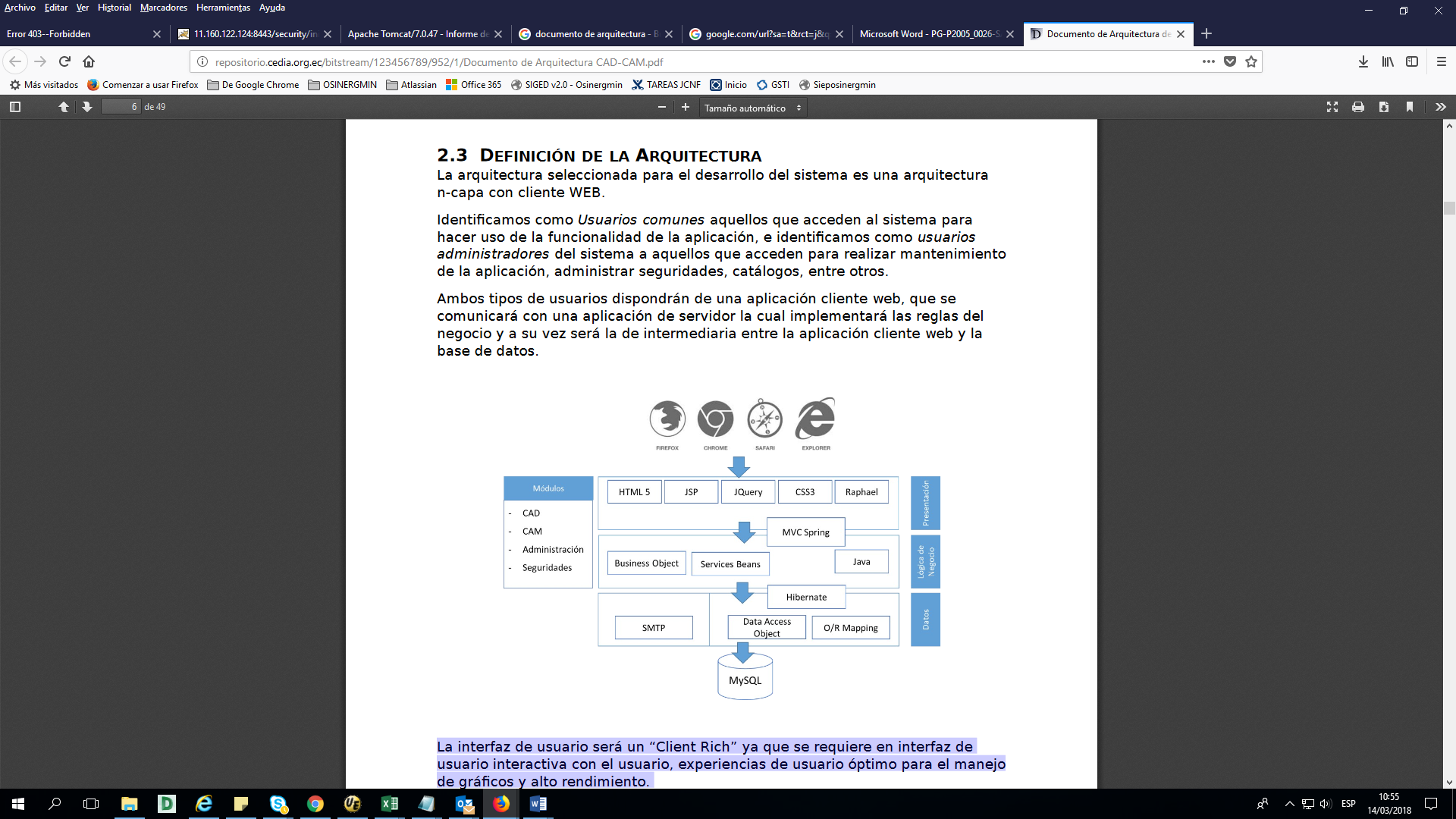
Esta sección describe la arquitectura de software para el Software actual, y cómo este será representado. De las vistas de Casos de Uso, Lógica, Procesos, Despliegue e Implementación; se enumerarán las vistas necesarias, y por cada vista, explica qué tipo de elementos del modelo contienen.

***Ejemplo****:*

*La arquitectura seleccionada para el desarrollo del Software es una arquitectura n-capa con cliente WEB.*

*Identificamos como Usuarios comunes aquellos que acceden al Software para hacer uso de la funcionalidad del Software, e identificamos como usuarios administradores del Software a aquellos que acceden para realizar mantenimiento del Software, administrar seguridades, catálogos, entre otros.*

*Ambos tipos de usuarios dispondrán de un Software cliente web, que se comunicará con un Software de servidor la cual implementará las reglas del negocio y a su vez será la de intermediaria entre el Software cliente web y la base de datos.*



*La interfaz de usuario será un “Client Rich” ya que se requiere en interfaz de usuario interactiva con el usuario, experiencias de usuario óptimo para el manejo de gráficos y alto rendimiento.*

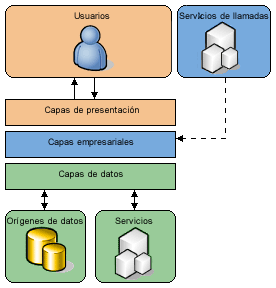
## CAPAS DE LA ARQUITECTURA

Estas secciones deben describir las capas de la arquitectura utilizada.

***Ejemplo:***

*El Software ha sido visionado como un Software web bajo la especificación Java Servlet, con una arquitectura de N-capas. Estará compuesta por las siguientes capas:*

* ***Capa de Presentación:*** *bajo patrón de diseño Modelo-Vista-Controlador:* 
  + *Vistas: Páginas web JSP con tag libraries de SpringMVC que despliegan formularios al usuario.*
  + *Controladores: Controladores de SpringMVC, que reciben y procesan las solicitudes del usuario.*
  + *Modelos: objetos reutilizados de la capa de modelo de dominio.*
* ***Lógica de Negocios (capas empresariales):*** 
  + *Capa de servicios: componentes encargados de ejecutar procesos complejos de la lógica de negocio, que interactuarán con los objetos del modelo de dominio.*
  + *Capa de modelo de dominio: componentes con la estructura conceptual que representa el dominio del Software, en la forma de JavaBeans tradicionales.*
* ***Capa de Datos:*** 
  + *Capa de implementación de acceso a datos: ORM Hibernate.*



## ARQUITECTURA DE INFRAESTRUCTURA

### PLATAFORMA DE SOFTWARE BASE

***Ejemplo:***

***Cliente:***

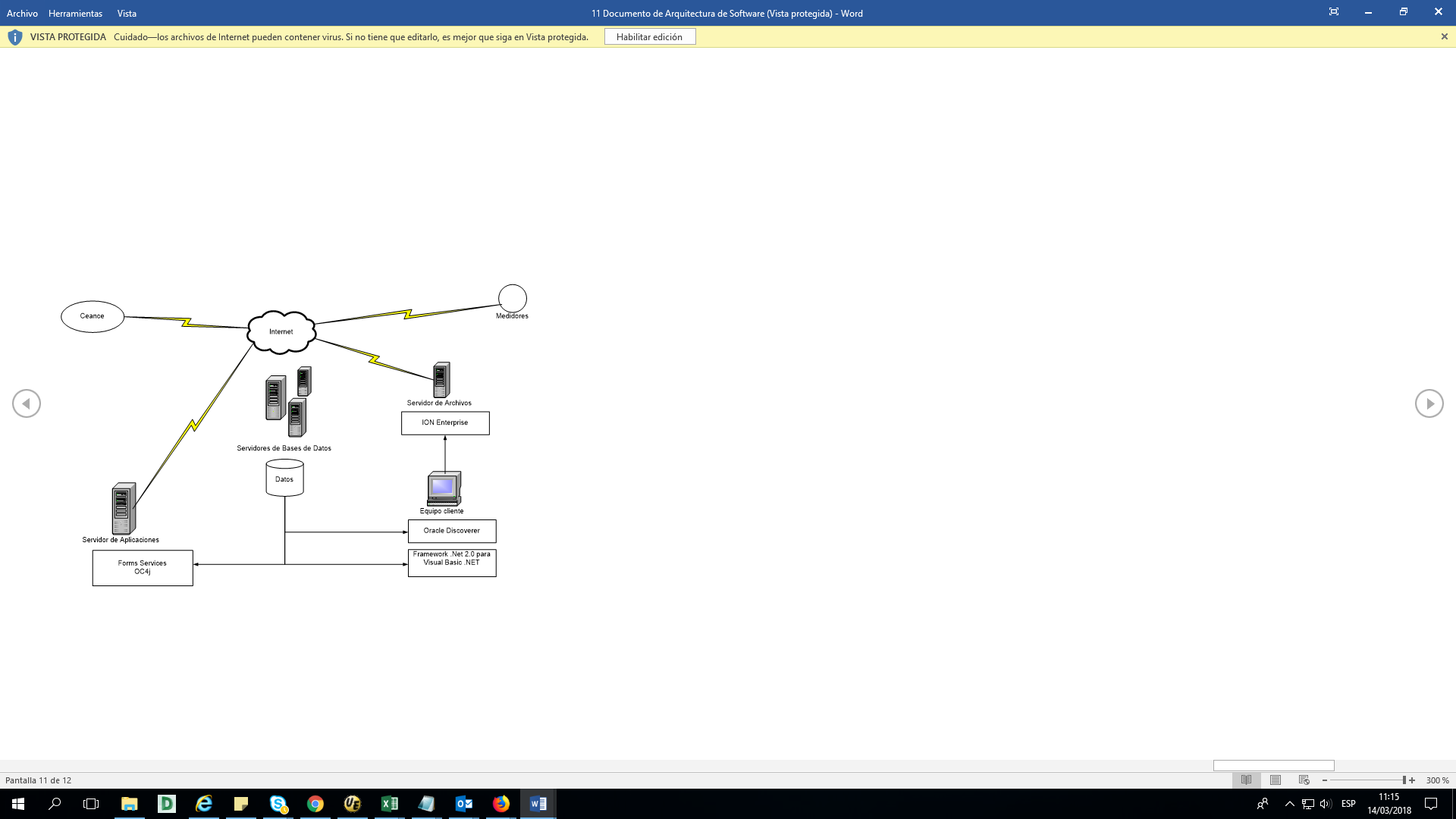
|  |  |
| --- | --- |
| ***Sistema Operativo*** | *Multiplataforma* |
| ***Browser*** | *Internet Explorer, Chrome, Firefox* |
| ***Tecnologías utilizadas*** | *HTML5 ,CSS3, jQuery* |
| ***Lenguajes Utilizados*** | *JavaScript* |

***Servidor de Software:***

|  |  |
| --- | --- |
| ***Sistema Operativo*** | *Multiplataforma* |
| ***Browser*** | *Tomcat* |
| ***Lenguajes Utilizados*** | *Java J2EE* |
| ***Framworks Utilizados*** | *Spring MVC, Hibernate* |

***Servidor de base de datos:***

|  |  |
| --- | --- |
| ***Sistema Operativo*** | *Multiplataforma* |
| ***Base de Datos*** | *mySQL 5.5* |

***.***

# DIAGRAMA DE DESPLIEGUE

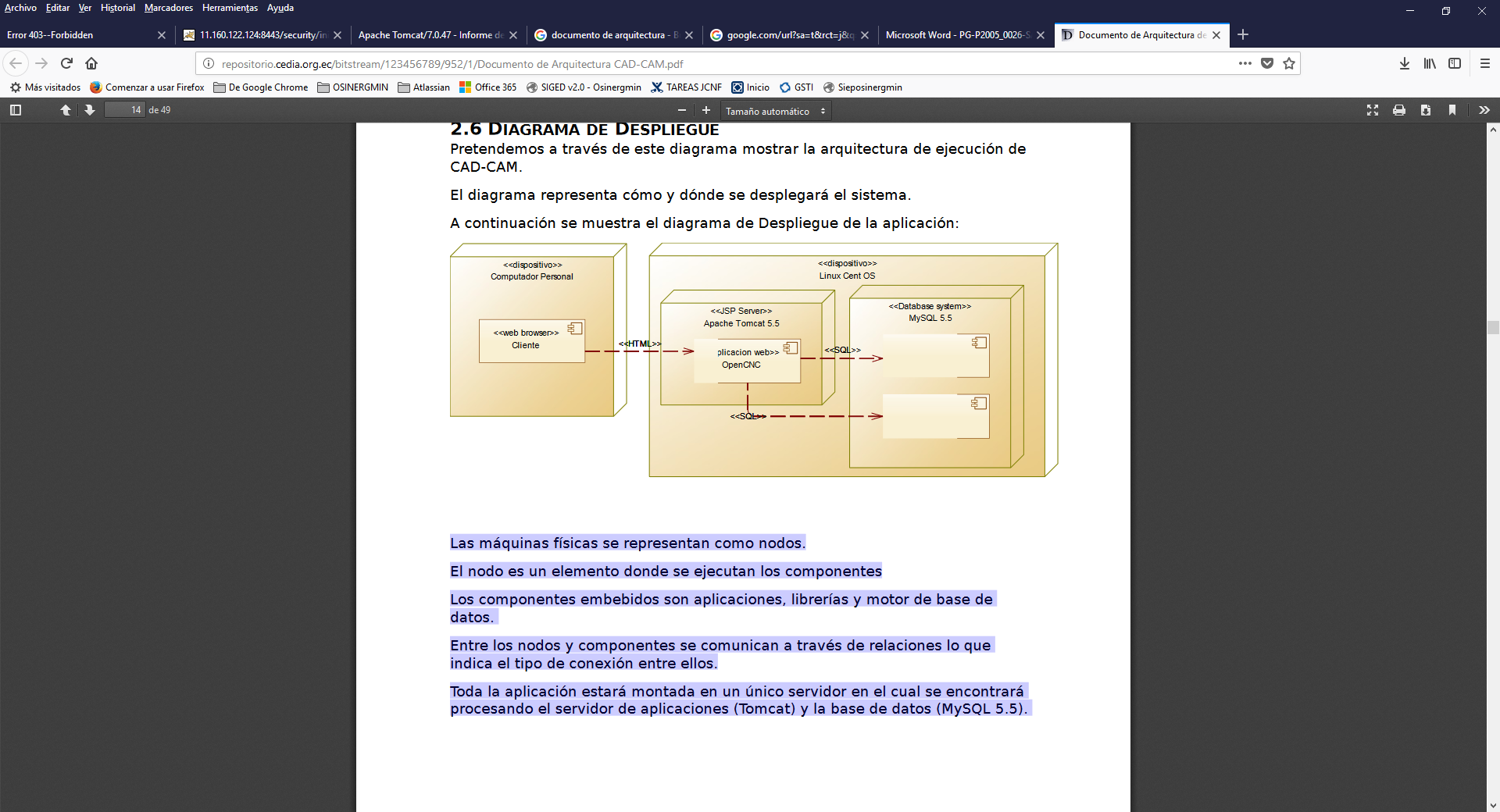
Esta sección colocar el diagrama de despliegue de la solución implementada, describiendo unas o más configuraciones físicas de la red (hardware) en las cuales se despliegue el funcionamiento del software. Es una vista del modelo del despliegue. En un mínimo para cada configuración debe indicar que los nodos físicos (computadoras, CPU) que se ejecuta el software y sus interconexiones (bus, LAN, punto a punto, y así sucesivamente.) también incluyen un mapeo de los procesos de la vista de procesos sobre los nodos físicos.

***Ejemplo****:*

*Pretendemos a través de este diagrama mostrar la arquitectura de ejecución del sistema xx*

*El diagrama representa cómo y dónde se desplegará el sistema.*

*A continuación, se muestra el diagrama de Despliegue del Software:*





*Las máquinas físicas se representan como nodos.*

*El nodo es un elemento donde se ejecutan los componentes*

*Los componentes embebidos son aplicaciones, librerías y motor de base de datos.*

*Entre los nodos y componentes se comunican a través de relaciones lo que indica el tipo de conexión entre ellos.*

*Toda el Software estará montado en un único servidor en el cual se encontrará procesando el servidor de aplicaciones (Tomcat) y la base de datos (MySQL 5.5).*

# DIAGRAMA DE COMPONENTES

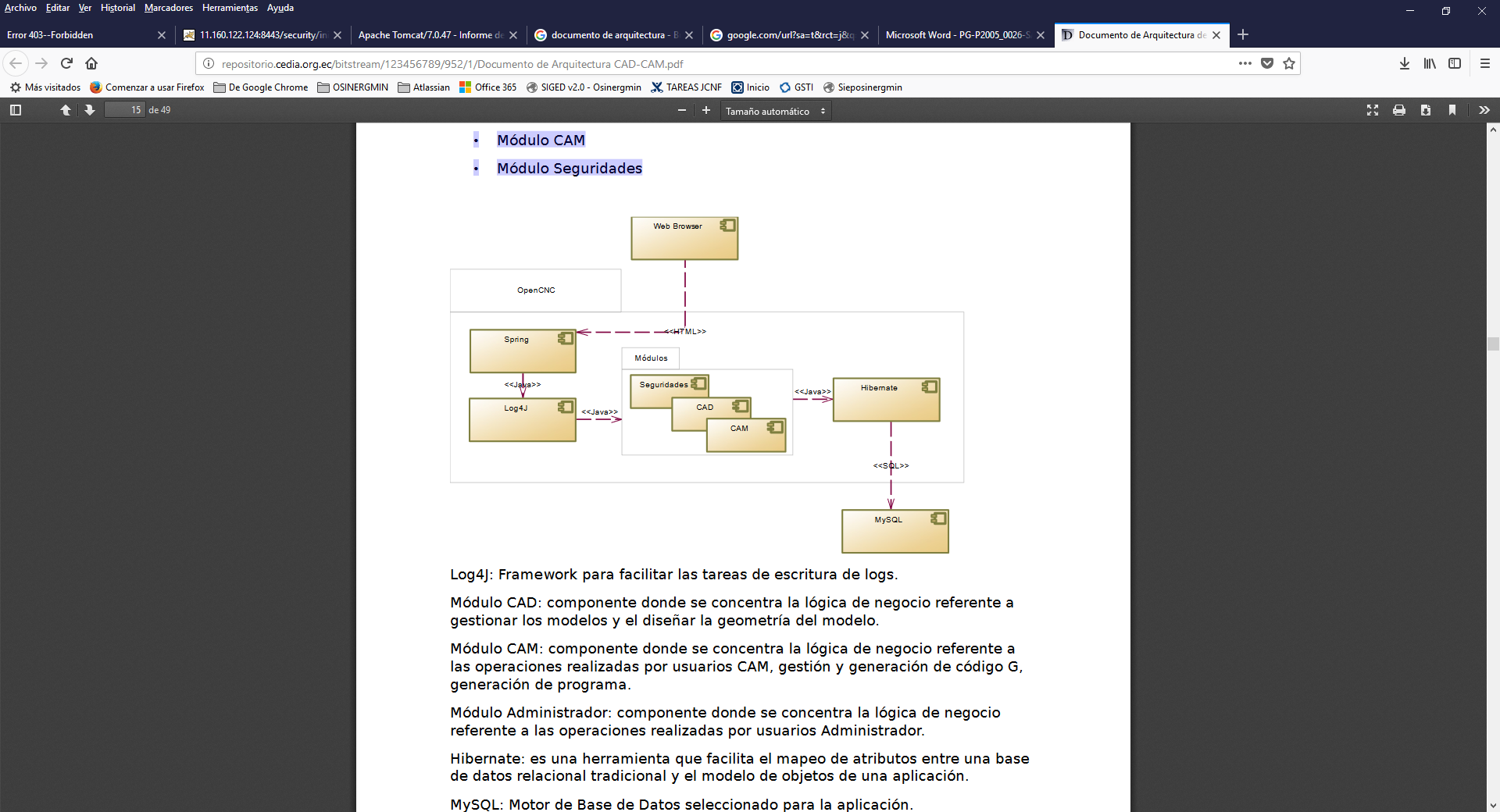
*El diagrama deberá mostrar los módulos en los que está dividida la funcionalidad del sistema*

***Ejemplo:***

*El diagrama de Componentes muestra los módulos en los que estará dividida la funcionalidad del sistema. Los conectores identifican la relación entre los mismos.*

*El Software cuenta con módulos claramente definidos los cuales acceden al modelo de datos único de la solución. Los módulos se dividen en:*

* *Módulo CAD, Módulo CAM, Módulo Seguridades*



*Log4J: Framework para facilitar las tareas de escritura de logs.*

*Módulo CAD: componente donde se concentra la lógica de negocio referente a gestionar los modelos y el diseñar la geometría del modelo.*

*Módulo CAM: componente donde se concentra la lógica de negocio referente a las operaciones realizadas por usuarios CAM, gestión y generación de código G, generación de programa.*

*Módulo Administrador: componente donde se concentra la lógica de negocio referente a las operaciones realizadas por usuarios Administrador.*

*Hibernate: es una herramienta que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de un Software.*

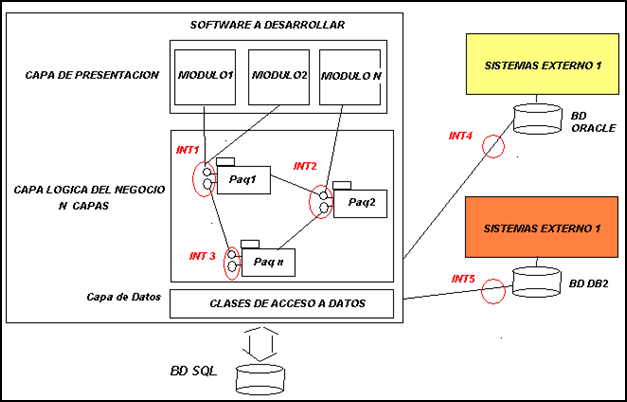
*MySQL: Motor de Base de Datos seleccionado para el Software.*

# VISTA DE INTEGRACIÓN

## DIAGRAMA DE INTEGRACIÓN E INTERFACES

[Incluir una descripción breve que explique rápidamente el grafico.

Por ejemplo:



|  |  |  |
| --- | --- | --- |
| **Interfaz** | **Descripción Breve** | **Tipo de Interfaz** |
| INT1 | La interfaz 1 apoya la integración del Paquete 1 y el Paquete 2, incluye las clases C1, C2, etc…. | Interfaz Interna |
| INT2 | La interfaz 1 apoya la integración del Paquete 1 y el Paquete 2, incluye las clases C1, C2, etc…. | Interfaz Interna |
| INT3 |  | Interfaz Interna |
| INT4 |  | Interfaz Externa |
| INT5 |  | Interfaz Externa |

## CRITERIOS PARA EL DISEÑO Y SELECCIÒN DE INTERFACES

Por ejemplo:

Para la selección y el diseño de las diferentes alternativas de las interfaces se evaluarán los siguientes criterios:

* Facilidad de Construcción
* Tiempo de Respuesta
* otros

## CRITERIOS DE INTEGRACIÒN DEL SOFTWARE

Por ejemplo:

Para la óptima integración del Software se deberán tener que cumplir, considerar y evaluar los siguientes criterios:

* Antes de realizar la integración todos los componentes deberán haber pasado por pruebas unitarias.
* Antes de realizar la integración, todas las incidencias, errores u otras no conformidades encontradas durante las pruebas unitarias deberán estar cerradas.
* Se deberá tener preparado los ambientes y entornos para la integración (Entorno de Desarrollo o Entorno de Integración).
* Deberá haberse inicializado y migrado data consistente previa a la integración.
* Otros Criterios que apoyen a que la integración resulte un éxito.

## SECUENCIA DE INTEGRACIÒN

Por ejemplo:

Para que el Software se integre totalmente se seguirá la siguiente secuencia de integración:

* Realizar las pruebas unitarias a todos los componentes desarrollados (De todos los módulos).
* Levantar todos los errores e incidencias encontradas en las pruebas unitarias (De todos los módulos).
* Realizar revisión de pares al código fuente y levantar las no conformidades.
* Asegurarse que todos los componentes del Software estén completamente corregidos (Realización de nuevas pruebas sobre los errores encontrados).
* Validar que el entorno de integración este listo.
* Validar que la data haya sido migrada satisfactoriamente.
* Iniciar la integración
  + Integrar Modulo 1 y Modulo 2 - Realizar pruebas de integración entre ambos módulos.
  + Integrar Modulo 1 y Modulo 2 y Modulo3 - Realizar pruebas de integración entre módulos.
  + Integrar Modulo 1 y Modulo 2 y Modulo n - Realizar pruebas de integración entre módulos.
* Finalizada la Integración entre módulos, realizar la integración con aplicativos externos al Software en desarrollo.
  + Integrar Software en desarrollo con Software Externo1 (Aplicativo Externo) y Realizar Pruebas.
  + Integrar Software en desarrollo con Software Externo2 (Aplicativo Externo) y Realizar Pruebas.
* Finalmente realizar las pruebas del Software y luego de ellas las Pruebas de Aceptación con los Usuarios Finales.

# CONTROL DE VERSIONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión N°** | **fecha** | **Autor** | **Descripción** |
| 1.0 | 12/01/2018 | Edwin Ramos | Creación del Documento |
|  |  |  |  |