



# Especificación de Requerimientos de Software Versión 2.5

## Historia de revisiones

Fecha	Versión	Descripción	Autor
29/08/2014	1.0	Especificación de requerimientos	Pablo Minetti
31/08/2014	1.1	Se agregan comentarios y/o sugerencias	Ezequiel Sanchez
31/08/2014	1.2	Se agregan pequeños cambios y se especifican ciertos conceptos	Mariangeles Artola
31/08/2014	1.3	Se corrigen algunas oraciones y se agregan requisitos que faltaban. Se agregan comentarios y/o sugerencias.	Gonzalo Arcos
01/09/2014	1.4	Correcciones varias, se juntaron algunos requisitos y se mejoró el formato del documento. Se agregaron varios comentarios y/o sugerencias.	Rodrigo Ferreri, Pablo Minetti
01/09/2014	1.5	Añadidos 3 requisitos más	Gonzalo Arcos
01/09/2014	1.6	Revisión de Documento	Diego Melli
02/09/2014	1.7	Se cambió el formato del	Gonzalo Arcos, Rodrigo Ferreri,

		documento, se agregaron requisitos, se refinó la descripción de los que ya había.	Pablo Minetti
05/09/2014	1.8	Se realizaron correcciones propuestas por el cliente. Se agregaron requisitos faltantes. Se modificaron algunas secciones.	Rodrigo Ferreri
08/09/2014	1.9	Se realizaron más correcciones propuestas por el cliente.	Rodrigo Ferreri
08/09/2014	2.0	Revisión de Documento	Diego Melli
09/09/2014	2.1	Se hacen modificaciones producto del feedback del cliente.	Gonzalo Arcos
11/09/2014	2.2	Se corrigen requerimientos producto de un email del cliente	Gonzalo Arcos, Rodrigo Ferreri
13/09/2014	2.3	Revision	Diego Melli
15/09/2014	2.4	Más correcciones	Rodrigo Ferreri
22/09/2014	2.5	Revision	Diego Melli

# Contenido

## [Introducción](#)

[Propósito](#)

[Alcance](#)

[Definiciones, siglas y abreviaturas.](#)

[Referencias](#)

[Visión general](#)

## [Descripción general](#)

[Perspectiva del producto](#)

[Interfaces de Usuario](#)

[Interfaces con hardware](#)

[Interfaces con software](#)

[Interfaces de comunicación](#)

[Restricciones de memoria](#)

[Restricciones de adecuación al entorno](#)

[Funciones del producto](#)

[Características de los usuarios](#)

[Restricciones de diseño](#)

[Modelo de procesos](#)

[Herramientas de Desarrollo](#)

[Supuestos y dependencias](#)

## [Requerimientos específicos](#)

[Requerimientos Suplementarios](#)

## [Requerimientos de documentación](#)

[Manual de Usuario](#)

[Ayuda en línea](#)

[Guías de instalación, configuración y archivo Léame.](#)

[Etiquetado y empaquetado](#)

## 1. Introducción

El presente documento tiene como objetivo brindar una descripción detallada de los requerimientos del sistema para el sistema de gestión de requisitos y casos de pruebas que se desarrollará para la empresa Genexus Consulting, obtenidos en el relevamiento realizado por parte del equipo de analistas en conjunto con el cliente.

El sistema que se pretende construir tomará las funciones provistas por un sistema actual llamado *Test Sync*, agregando nuevas funcionalidades y mejorando las ya existentes.

### 1.1. Propósito

El propósito de este documento es describir de manera formal el comportamiento del sistema que se pretende construir. A continuación se detallan los requerimientos funcionales, las restricciones y atributos de calidad que deberá cumplir el sistema. El documento está dirigido a:

- El cliente.
- Todos los miembros del equipo de desarrollo.
- El administrador.
- El arquitecto.
- El Director
- SQA y SCM.

### 1.2. Alcance

El sistema se desarrollará con el fin de mejorar el proceso de desarrollo de la empresa, manteniendo un repositorio centralizado de toda la documentación asociada a las etapas de requerimientos y testing del proyecto. El sistema permitirá:

- Dar de alta y baja un requisito, administrando los cambios que estos puedan tener y permitiendo asignarles distintas prioridades.
- Realizar modelado de los requisitos por medio de la técnica de casos de uso, así como de diagramas de flujos de trabajo de negocios.
- Facilitar el trabajo con requisitos dentro de cada organización involucrada con el proyecto (cliente y usuarios, equipo de desarrollo y auditores) facilitando un ambiente controlado, colaborativo y centralizado en el cual trabajar.
- Facilitar y agilizar la comunicación entre el cliente y los usuarios del sistema.
- Registrar la información correspondiente a los casos de prueba, manteniendo tanto un versionado de los mismos como un historial de ejecuciones.
- Realizar estimaciones de tiempo de ejecución y tiempo de creación de planes de pruebas.
- Mantener un histórico de versiones (versionado) tanto de requisitos como de casos de uso y casos de prueba.
- Facilitar la navegabilidad y lograr trazabilidad entre requerimientos, casos de uso, objetos Genexus y los casos de prueba.
- Integrar el sistema con GxTest para permitir la ejecución automatizada de casos de prueba.

- Integrar el sistema con GxServer para permitir conocer los objetos GeneXus asociados a casos de uso.
- Integrar el sistema con Mantis Bug Tracker para permitir el registro de incidentes.
- Crear, consultar y editar requisitos desde un Smart Device.

### 1.3. Definiciones, siglas y abreviaturas.

- RF: Requerimiento funcional.
- RNF: Requerimiento no funcional.
- GAM: Genexus Access Manager.

A lo largo de todo el documento, se utilizan los términos *Requisito* y *Requerimiento* en forma indistinta.

### 1.4. Referencias

- [1] Documento con información a almacenar de requerimientos: PIS 2014 - Requerimientos.docx
- [2] Mantis Bug Tracker: <https://www.mantisbt.org/>
- [3] GxTest: <http://gxserver.com/>
- [4] GxServer: <http://www.abstracta.com.uy/es/gxtest.html>

### 1.5. Visión general

En este documento se detallan características generales del proyecto además de especificaciones sobre requisitos funcionales y no funcionales.

## 2. Descripción general

### 2.1. Perspectiva del producto

El sistema funcionará en un ambiente web, por lo que deberá ser capaz de funcionar en cualquier plataforma con un navegador web instalado, siempre y cuando disponga de conexión al servidor en que va a estar corriendo la aplicación, ya sea por Internet o alguna red privada.

El sistema tendrá además un conjunto reducido de funcionalidades que podrán ser accedidas desde un Smart Device.

#### 2.1.1. Interfaces de Usuario

El sistema tendrá un portal donde cada usuario podrá consultar tanto la parte del módulo de requisitos como la de pruebas.

#### 2.1.2. Interfaces con hardware

No se han identificado interfaces con hardware.

#### 2.1.3. Interfaces con software

No se han identificado interfaces con software.

#### 2.1.4. Interfaces de comunicación

El sistema proveerá interfaces de comunicación con los siguientes sistemas:

- **Mantis Bug Tracker:** Gestor de incidencias de código abierto que permite gestionar tareas e incidentes en un equipo de trabajo, el propósito de la interfaz del producto con el sistema es el de integrar la ejecución de los casos de prueba asociados a los incidentes que se registran.
- **GxServer:** Herramienta que permite facilitar el desarrollo de aplicaciones GeneXus tanto de forma individual como en equipo de trabajo, disponiendo de los objetos de una aplicación en desarrollo públicos para la modificación de todo el equipo, el objetivo de la interfaz es integrar los casos de uso y los casos de prueba con los objetos GeneXus, permitiendo una mejor trazabilidad de los mismos.
- **GxTest:** Herramienta que permite automatizar las pruebas funcionales sobre aplicaciones web GeneXus. Esta aplicación se compone de dos partes, que son GxTest Designer, que permite correr scripts de pruebas individuales en un equipo, y GxTest Manager, que permite planificar un conjunto de pruebas para que estas se ejecuten en determinado equipo. El propósito de la interfaz de esta aplicación con el sistema es el de unificar las pruebas manuales y las pruebas planificadas dentro de un mismo sistema, permitiendo ambas funcionalidades, y a su vez mantener indicadores de la ejecución de dichas pruebas.

#### 2.1.5. Restricciones de memoria

No hay restricciones de memoria conocidas.

#### 2.1.6. Restricciones de adecuación al entorno

No hay restricciones de adecuación al entorno conocidas.

### 2.2. Funciones del producto

Las principales funciones que el sistema debe brindar son:

- **En lo que concierne a la gestión de los requerimientos:**
  - Administrar información referente a los requisitos. La información a almacenar está especificada en el documento [1].
  - Gestionar los cambios, realizando un versionado, y mantener el estado en que se encuentran los requerimientos permitiendo así un seguimiento de los mismos.
  - Asignar distintas prioridades a los requerimientos, manteniendo un histórico de dichas prioridades.
  - Permitir modelar requisitos mediante las técnicas de casos de uso y de diagramas de flujos de trabajo de negocios.
  - Integrar el sistema con GxServer para permitir asociar a cada caso de uso distintos objetos GeneXus.
  - Brindar funcionalidades para exportar los requerimientos a Word y PDF.

- **En lo que concierne a la gestión de casos de prueba:**
  - Registrar la información correspondiente a los casos de prueba, manteniendo un versionado de los mismos. Además, registrar un historial de la ejecución de los mismos.
  - Realizar estimaciones tanto de tiempo de ejecución como de creación de planes de pruebas.
  - Brindar funcionalidades para exportación los casos de prueba a Word y PDF.
  - Integrar el sistema con GXTest Manager.
  - Integrar el sistema con Mantis Bug Tracker.

## 2.3. Características de los usuarios

El producto a construir va dirigido a dos tipos de usuarios principales. En primer lugar, usuarios con conocimientos avanzados en lo referente a sistemas de información, que poseen conocimientos específicos sobre los procesos de ingeniería de requisitos y gestión de pruebas de software. En segundo lugar, está dirigido a clientes de proyectos que en algunos casos no tienen conocimientos sobre el tema.

## 2.4. Restricciones de diseño

### 2.4.1. Modelo de procesos

Se utilizará el modelo de procesos MUM (Modularizado, Unificado y Medible) propuesto por la Facultad de Ingeniería en la materia Proyecto de Ingeniería de Software. Dicho proceso consta de cuatro fases, que son: Inicial, Elaboración, Construcción y Transición. Este modelo está pensado para un equipo de trabajo de entre 10 y 14 personas, por un periodo de 14 semanas.

### 2.4.2. Herramientas de Desarrollo

El sistema a construir será implementado con la herramienta **Genexus Ev2 Upgrade 5** para plataformas Windows. El lenguaje generado deberá ser **Java**. Además, se utilizará un manejador de bases de datos **MySql**.

## 2.5. Supuestos y dependencias

No se identificaron supuestos ni dependencias.

### 3. Requerimientos específicos

A continuación se detallan los requisitos funcionales que el sistema deberá implementar. Todos los requerimientos especificados a continuación están pendientes de aprobación.

#### Web y Smart Devices:

- **RF-01 - Autenticación y autorización de usuarios:**

El sistema deberá permitir a los usuarios iniciar sesión en el mismo. Esto incluye brindar un sistema de soporte a la autenticación y autorización de usuarios, así como de gestión de los mismos. También deberá permitirles cerrar una sesión iniciada previamente.

Se deberán soportar los siguientes roles: analista funcional, desarrollador, tester y cliente. También existirá un administrador del sistema y otro para cada proyecto. Cada uno de estos usuarios tendrá diferentes permisos y visibilidad de recursos del sistema dependiendo del rol que cumplan.

- **RF-02 - Gestión de proyectos**

El sistema deberá permitir gestionar la información correspondiente a un proyecto de software (esto es, realizar el ingreso, consulta y modificación de los mismos). El proyecto es una entidad que agrupa requisitos y casos de prueba, así como usuarios con distintos roles que actúan sobre estos.

Los proyectos están asociados a una organización, que también debe representarse en el sistema. Las organizaciones tienen asociadas áreas de negocio, y también procesos de negocio.

Los proyectos también contienen información sobre cómo realizar la autenticación sobre los servidores de GxServer, GxTest y Mantis Bug Tracker, de forma tal de poder realizar la integración con dichas herramientas.

El sistema deberá permitir que un usuario tenga roles distintos en dos proyectos diferentes.

- **RF-03 - Gestión de la línea base del proyecto**

El sistema deberá permitir gestionar la línea base del proyecto, que incluirá tanto a los requerimientos como casos de prueba si se desea. Una línea base es una versión congelada (snapshot) del sistema en un momento determinado, que contiene a todas las últimas versiones de todos los objetos (requisitos, casos de uso y casos de pruebas). Dichas versiones no son modificables una vez que ingresan a la línea base.

El sistema permitirá tanto crear como consultar líneas base. Al crear una línea base, se ingresa su nombre y su descripción, además de que se registra el responsable de su creación. Las líneas base se identifican con números generados automáticamente en forma ascendente. A su vez, se guardará el histórico de todas las líneas base del proyecto para consultas futuras.

- **RF-04 - Ingreso de requisitos:**

El sistema deberá permitir el ingreso de requisitos por parte de los usuarios. Los requisitos a ingresar podrán ser tanto funcionales como no funcionales, así como también transicionales.

Los datos que se ingresan de un requisito se encuentran en [1].



- **RF-05 - Relaciones entre requisitos:**

El sistema deberá permitir al usuario asociar requisitos que estén relacionados entre sí. Un requisito podrá estar relacionado con cualquier cantidad de otros requisitos. Se desean registrar las relaciones de:

- Depende de
- Bloquea a
- Relacionado con

Las relaciones son entre dos versiones de requisitos distintos. Además, las relaciones Depende de y Bloquea a son complementarias (si un requisito se encuentra en una relación de Depende de con otro, entonces automáticamente el otro requisito debe encontrarse en una relación de Bloquea a con este).

- **RF-06 - Actualización de requisitos:**

El sistema deberá permitir la actualización de cualquier atributo de un requisito previamente ingresado, ya sea por un analista funcional o por el cliente mismo. Esto incluye permitir cambiar el estado de un requisito, y su prioridad. Luego de realizar cualquier cambio, el sistema deberá crear una nueva versión del requisito, manteniendo el historial de todas las versiones anteriores.

- **RF-07 - Consulta de información de requisitos:**

El sistema permitirá a los usuarios consultar la información de cualquier requisito ingresado (sobre el cual tenga permisos). Para cada requisito, se podrán ver toda la información registrada del mismo, que está definida en el documento [1]. Además, se podrán consultar todas las versiones anteriores por las cuales pasó. También se podrán ver los requisitos relacionados al actual.

- **RF-08 - Priorización de requisitos**

El sistema permitirá la priorización de un requisito que haya sido ingresado previamente al sistema. Los cambios de prioridad del requisito deberán ser almacenados para mantener un histórico de los mismos, es decir, se generan nuevas versiones al cambiar la prioridad de un requisito. Se desean cuatro niveles de prioridad:

- 1 - Debe incluirse (Must have)
- 2 - Debería incluirse (Should have)
- 3 - Podría incluirse (Could have)
- 4 - No se incluye ahora (Won't have this time)

También se desea poder agregar prioridades nuevas al sistema en caso de necesitarse.

- **RF-09 - Cambio de estado de requisitos**

El sistema deberá permitir cambiar el estado asociado a los requisitos. Los estados válidos para un requisito son:

- Borrador
- En espera de aprobación

- Aprobado
- Rechazado

Para cada cambio de estado se deben registrar tanto la fecha como al usuario responsable del mismo. El sistema deberá registrar también el histórico de los estados por los que ha pasado un requisito. Para el caso de un cambio a estado Aprobado o Rechazado, todos los que se encuentran registrados como interesados por el requisito, además del usuario solicitante, deben realizar el cambio de estado en forma unánime. Solo es posible realizar cambios de estado desde Borrador a En espera de aprobación, de allí a Aprobado o Rechazado, y luego de cualquiera de estos dos a Borrador. Ante cualquiera de estos cambios, debe generarse una nueva versión del requisito.

- **RF-10 - Versionado de requisitos:**

El sistema deberá almacenar un histórico de las distintas versiones por las que pasa un requisito a lo largo en proyecto. Cada vez que se realiza una modificación en un requisito, se genera una nueva versión, que pasa a ser la última, pero la versión anterior no se descarta sino que se almacena en un historial.

**Web:**

- **RF-11 - Integración con herramienta de modelado de procesos de negocio:**

El sistema deberá permitir realizar el modelado de un requisito mediante una herramienta de modelado de procesos de negocio. Para esto, se deberá realizar una integración con la misma.

- **RF-12 - Modelado de requisitos mediante casos de uso:**

El sistema permitirá modelar requisitos previamente registrados utilizando casos de uso. La plantilla con la información que se desea registrar sobre los casos de uso se encuentra en [1]. Un caso de uso podrá asociarse a más de un requisito, y además cada requisito podrá tener cero o más casos de uso asociados. Los casos de uso también pueden tener más casos de uso relacionados.

- **RF-13 - Ingreso de casos de prueba:**

El sistema permitirá el ingreso de casos de prueba junto con sus datos asociados. Los casos de prueba podrán ser de regresión o no, además de manuales o automáticos. En el caso de que un caso de prueba sea automático, se deberá ofrecer un mecanismo para seleccionar su correspondiente en la herramienta GxTest.

La información a almacenar de los casos de prueba es la misma utilizada en TestSync, pero se desea eliminar la opción de seleccionar la técnica de caso de prueba que corresponde, que en la versión actual de la herramienta está limitada a clases de equivalencia.

Los casos de prueba se podrán agrupar dentro de Suites de casos de prueba, las cuales funcionarán como carpetas contenedoras (que tengan alguna relación lógica que le sea de utilidad al usuario). Un caso de prueba puede pertenecer a más de una Suite.

La información de los casos de prueba existentes deberá poder ser consultada. En particular, el sistema deberá permitir al usuario realizar búsquedas por dos filtros: pruebas de regresión y pruebas por módulo. Los

casos de prueba que resulten de esta búsqueda podrán ser exportados a Word, como en RF-15.

- **RF-14 - Guardar versionado de casos de prueba**

El sistema deberá mantener un versionado (histórico) de los casos de pruebas. Al realizar cualquier modificación sobre un caso de prueba se creará una nueva versión, pero no se eliminarán las anteriores, que se mantendrán en un histórico. El histórico podrá ser consultado por el usuario.

- **RF-15 - Exportación de casos de prueba**

El sistema permitirá exportar uno o un conjunto de casos de prueba al formato docx. También se permitirá exportar un plan de pruebas. Adicionalmente, se podrá exportar al formato pdf.

- **RF-16 - Asignación de casos de prueba a Tester**

El sistema permitirá asignar un caso de prueba (o un conjunto de casos de prueba) a un Tester para su ejecución.

- **RF-17 - Ejecución de casos de prueba**

El sistema permitirá realizar la ejecución de casos de prueba (o planes de prueba), tanto en forma automática como manual. En el caso de la ejecución automática, se realizará una integración con la herramienta GxTest Manager para este propósito. Los resultados de las pruebas deberán ser obtenidos y registrados para su posterior evaluación. En el caso de la ejecución manual, el Tester deberá completar los resultados de la ejecución de los casos de prueba. Los datos a completar son los mismos que en la herramienta TestSync.

- **RF-18 - Estimaciones sobre casos de prueba**

El sistema permitirá ingresar estimaciones a nivel de casos de prueba. Las estimaciones harán referencia al tiempo que se estima que llevará determinada tarea relacionada con los casos de prueba. Las estimaciones se pueden hacer sobre el tiempo que lleva el diseño de un caso de prueba, el tiempo de preparación del mismo y el tiempo de ejecución que llevará.

- **RF-19 - Priorización de los casos de prueba dentro de un plan de prueba**

El sistema permitirá asignar prioridades a los casos de prueba dentro de un plan de ejecución. Se almacenarán los cambios de prioridad realizados (tanto fecha como responsable).

- **RF-20 - Creación de planes de pruebas**

El sistema permitirá la creación de planes de pruebas. Se podrán agrupar distintos casos de prueba en un único plan de pruebas. Los planes de prueba se entienden como un conjunto de casos de pruebas planificados para su ejecución. Los planes de prueba tendrán registrado al responsable del mismo, y podrán ser ejecutados, como en RF-17.

- **RF-21 - Registro de incidentes**

Se deberán registrar los incidentes ocurridos durante la ejecución de los casos de pruebas. Para esto, se deberá realizar una integración con la herramienta Mantis Bug Tracker, registrando allí los incidentes detectados, y almacenando el número que retorna esta herramienta de forma tal de poder consultarlos en forma posterior. El sistema deberá proveer un mecanismo que realice este reporte automáticamente. Se deben de poder registrar más

de un incidente por caso de prueba, y a su vez, asociar un mismo incidente a más de un caso de prueba.

- **RF-22 - Indicadores sobre planes de prueba**

El sistema deberá poder asociar indicadores a planes de prueba. Dichos indicadores pueden referirse a:

- Cantidad de pruebas por plan que se ejecutaron, que pasaron exitosamente, que fallaron, que fueron bloqueadas, y que no se ejecutaron. Interesa conocer el porcentaje de cada uno de estos valores (es decir, cantidad sobre cantidad total de casos de prueba en el plan de pruebas).
- Cantidad de incidentes detectados por plan de pruebas y/o módulo, detallados por criticidad y severidad.
- Cantidad de incidentes en relación al tamaño de la KB. El tamaño de la KB es una métrica que se definió como la cantidad de objetos GeneXus que se tiene.

- **RF-23 - Trazabilidad entre los módulos de requisitos y pruebas**

El sistema deberá proveer un mecanismo para permitir la trazabilidad horizontal entre un requisito, los casos de uso que lo modelan, los objetos GeneXus que implementan al caso de uso y los casos de prueba asociados a este. El sistema debe permitir que esta relación sea visible para los usuarios, sujeto a los permisos que tenga el mismo.

- **RF-24 - Gestión de módulos**

El sistema deberá mantener el concepto de módulos para un proyecto determinado, permitiendo crear, modificar y consultar información sobre ellos. En particular, se deberá poder definir módulos, permitiendo posteriormente incluir dentro del mismo requisitos, así como también casos de uso y casos de prueba.

- **RF-25 - Exportación de requisitos**

El sistema permitirá exportar uno o un conjunto de requisitos al formato docx. Adicionalmente, se podrá exportar al formato pdf.

- **RF-26 - Exportación de casos de uso**

El sistema permitirá exportar uno o un conjunto de casos de uso al formato docx. Adicionalmente, se podrá exportar al formato pdf.

## 1.1. Requerimientos Suplementarios

A continuación se detallan los requerimientos no funcionales para el sistema.

- **RNF-01 - Interfaz del sistema**

La interfaz del sistema debe estar implementada como una aplicación web, a excepción de la extensión para Smart Devices. Se pretende además la existencia de un portal de requisitos.

- **RNF-02 - Usabilidad**

El sistema deberá proveer una interfaz amigable al usuario, que sea fácil de aprender y de usar. Se pretende que la navegación entre las distintas funcionalidades del sistema sea realizada en forma sencilla y en pocos pasos. La interfaz deberá ofrecer funcionalidades para realizar tareas repetitivas en forma automática.

- **RNF-03 - Utilización de K2B Tools**

Se deberá utilizar el patrón K2B Tools para estandarizar la interfaz gráfica.

- **RNF-04 - Utilización de GAM**

Se deberá utilizar Genexus Access Manager para toda la gestión de usuarios y control de acceso; autenticación y autorización.

## **4. Requerimientos de documentación**

### **4.1. Manual de Usuario**

Se deberá realizar un manual que especifique y ejemplifique las principales funcionalidades del sistema. Este manual será un medio por el cual los usuarios podrán aprender a utilizar el sistema, por lo que deberá ser realizado con este propósito.

### **4.2. Ayuda en línea**

El cliente manifestó que no desea ningún tipo de ayuda en línea.

### **4.3. Guías de instalación, configuración y archivo Léame.**

[En esta sección especifique si el sistema de software contendrá instrucciones para instalación y configuración. Además si se incluirá el típico archivo Léame, que puede incluir las Novedades de la versión, discusión de compatibilidad con versiones anteriores, documentación de errores conocidos y soluciones alternativas.]

### **4.4. Etiquetado y empaquetado**

[El estado del arte de las aplicaciones de hoy proporciona un aspecto consistente que comienza con el paquete del producto y se manifiesta a través de los menús de la instalación, las pantallas del sistema, los sistemas de ayuda, los diálogos con el usuario, etc. Esta sección define las necesidades y tipos de etiquetas a para ser incorporado en el código, por ejemplo, derechos de propiedad literaria y avisos patentes, logotipos corporativos, iconos estandarizados y otros elementos gráficos, etc.]