



## **Descripción de la Arquitectura**

### **Versión 1.2**

### **Historia de revisiones**

Fecha	Versión	Descripción	Autor
01/09/2014	0.1	Versión inicial (borrador)	Rodrigo Ferreri
13/09/2014	1.0	Primera versión completa del documento	Rodrigo Ferreri
19/09/2014	1.1	Se agregó el modelo de transacciones	Rodrigo Ferreri
22/09/2014	1.2	Revisión	Diego Melli

# Contenido

<b>1.INTRODUCCIÓN .....</b>	<b>4</b>
1.1.PROPÓSITO.....	4
1.2.ALCANCE.....	4
1.3.DEFINICIONES, SIGLAS Y ABREVIATURAS.....	4
1.4.REFERENCIAS.....	4
1.5.VISIÓN GENERAL .....	4
<b>2.VISTA DEL MODELO DE CASOS DE USO.....</b>	<b>5</b>
2.1.DIAGRAMA DE CASOS DE USO RELEVANTES A LA ARQUITECTURA.....	5
2.2.CASOS DE USO RELEVANTES A LA ARQUITECTURA .....	5
<b>3.TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO .....</b>	<b>8</b>
3.1.1.Iniciar sesión.....	8
3.1.2.Registrar nuevo usuario.....	8
3.1.3.Crear proyecto.....	9
3.1.4.Seleccionar proyecto actual.....	9
3.1.5.Crear línea base.....	10
3.1.6.Crear requisito.....	10
3.1.7.Cambiar estado de un requisito .....	12
3.1.8.Crear caso de uso .....	12
3.1.9.Ingresar caso de prueba .....	13
3.1.10.Ejecutar caso de prueba automático.....	13
3.1.11.Ejecutar caso de prueba manual.....	14
3.1.12.Crear plan de pruebas .....	14
3.1.13.Registrar incidente.....	15
<b>4.VISTA DEL MODELO DE DISEÑO .....</b>	<b>16</b>
4.1.DESCOMPOSICIÓN EN SUBSISTEMAS .....	16
4.1.1.Web UI.....	16
4.1.2.Smart Devices UI .....	16
4.1.3.Módulo de autenticación y usuarios .....	16
4.1.4.Genexus Access Manager .....	17
4.1.5.Módulo de proyectos.....	17
4.1.6.Módulo de requisitos.....	17
4.1.7.Módulo de pruebas.....	17
4.1.8.Módulo de integración externa.....	17
4.1.9.Módulo de persistencia (Genexus).....	17
4.2.DISEÑO DE TRANSACCIONES.....	18
Módulo de autenticación y usuarios .....	18
4.2.1.User.....	18
4.2.2.RelatedTo.....	18
4.2.3.Module .....	18
4.2.4.Baseline.....	19
4.2.5.RequirementHead.....	19
4.2.6.RequirementVersion.....	19
4.2.7.RelatedTo.....	19
4.2.8.UseCaseHead.....	19
4.2.9.UseCaseVersion .....	19
4.2.10.State .....	19
4.2.11.GxObject .....	19
4.2.12.TestSuite.....	19
4.2.13.TestCaseHead .....	19
4.2.14.TestCaseVersion.....	19
4.2.15.TestPlan .....	19
4.2.16.TestCaseExecution .....	19
4.2.17.TestPlanExecution .....	20
4.2.18.Issue .....	20

4.3.DISEÑO DE CASOS DE USO.....	20
<b>5.TRAZABILIDAD DESDE EL MODELO DE DISEÑO AL MODELO DE IMPLEMENTACIÓN.....</b>	<b>21</b>
<b>6.VISTA DEL MODELO DE IMPLEMENTACIÓN .....</b>	<b>21</b>
6.1.SUBSISTEMAS .....	21
6.1.1.[Subsistema 1] .....	21
6.2.COMPONENTES .....	21
6.2.1. [Componente 1] .....	21
6.3.INTERFASES .....	21
6.3.1.[Interfase 1] .....	21
<b>7.VISTA DEL MODELO DE DISTRIBUCIÓN .....</b>	<b>22</b>
7.1.DIAGRAMA DE DISTRIBUCIÓN.....	22
7.2.NODOS.....	22
7.2.1.Cliente web.....	22
7.2.2.Cliente móvil (Smart Device).....	22
7.2.3.Apache Tomcat.....	22
7.2.4.BD GAM MySQL .....	23
7.2.5.BD Rexus MySQL.....	23
7.2.6.GxServer .....	23
7.2.7.GxTest Manager.....	23
7.2.8.Mantis Bug Tracker .....	23
7.3.CONEXIONES.....	23
7.3.1.Conexión https/internet.....	23
7.3.2.Conexión por Web Services .....	23
7.3.3.Conexión por JDBC.....	24

## **1. Introducción**

En este documento se presenta una descripción a nivel global de la arquitectura del sistema a construir.

### **1.1. Propósito**

Este documento proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

Está dirigido a los diseñadores e implementadores del sistema, incluyendo al Arquitecto y al Asistente de Arquitecto.

### **1.2. Alcance**

Este documento pretende describir al sistema ReXus, contemplando todos los subsistemas que lo componen y aquellos casos de uso críticos para que realice su trabajo. Este documento se basa en el documento de especificación de requerimientos para definir la arquitectura candidata, y por lo tanto será actualizado a medida que el proyecto avance y los requisitos del sistema se establezcan.

### **1.3. Definiciones, siglas y abreviaturas.**

Todas las definiciones, siglas y abreviaturas se encuentran en el glosario del proyecto [1].

### **1.4. Referencias**

[1] RQGLOG9v1.0.doc: Glosario

[2] RQDRQG9v2.4.doc: Especificación de Requerimientos.

[3] RQMODG9v2.0.doc: Modelo de casos de uso.

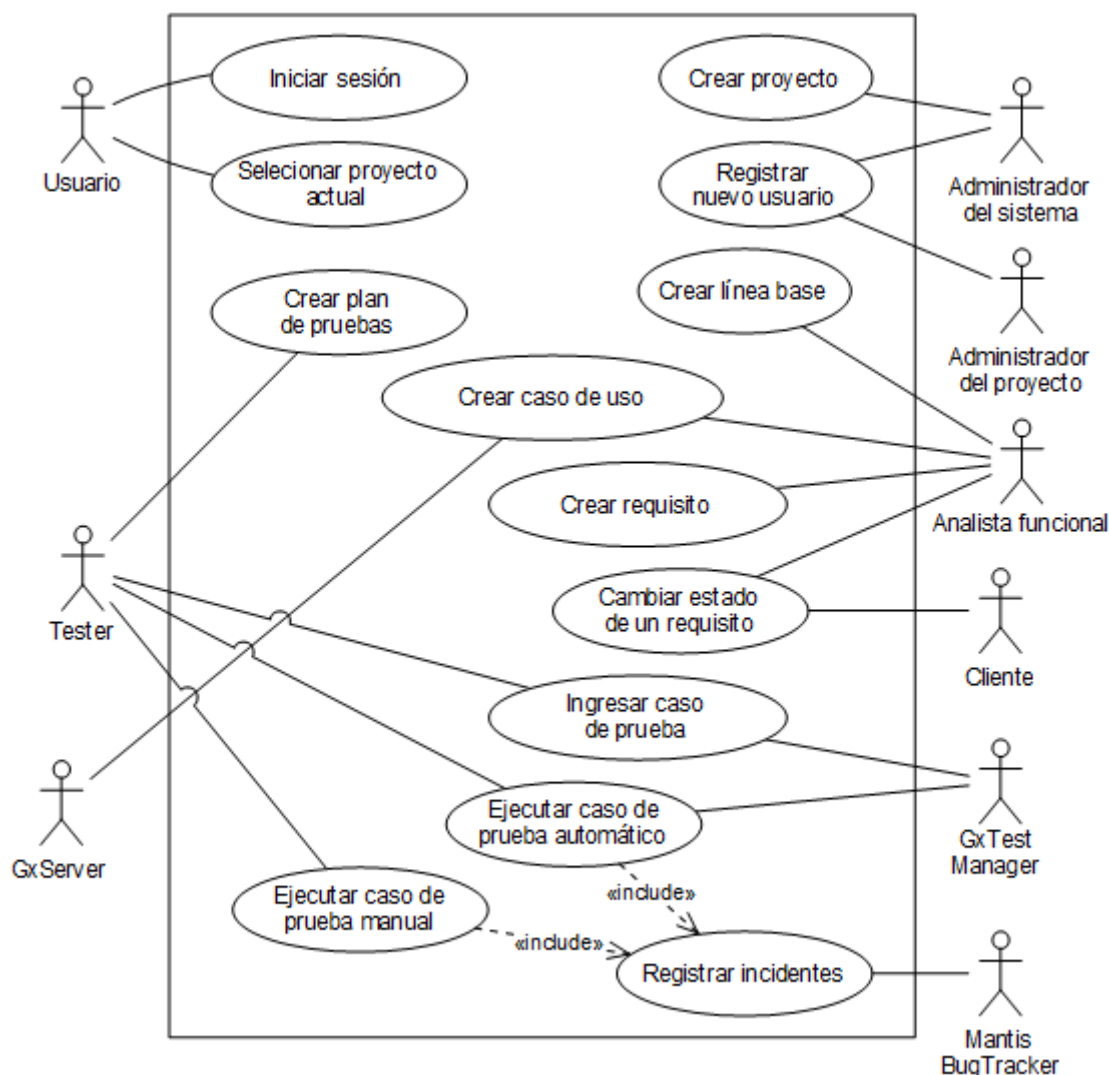
### **1.5. Visión general**

El resto de este documento contiene la descripción de la arquitectura del sistema en base a distintos modelos. La sección 2 presenta la vista de casos de uso, que contendrá a aquellos que sean críticos para la definición y elaboración del sistema. La sección 3 asocia los casos de uso definidos en el apartado anterior con el modelo de diseño que los implementa, mientras que la sección 4 describe el Modelo de Diseño, detallando los subsistemas y componentes que definen la solución planteada.

La sección 5 define los componentes de implementación asociados a los subsistemas presentados en el apartado anterior. La sección 6 muestra la vista del modelo de implementación. Por último, la sección 7 presenta la vista del modelo de distribución del sistema.

## 2. Vista del Modelo de Casos de Uso

### 2.1. Diagrama de Casos de Uso relevantes a la Arquitectura



### 2.2. Casos de Uso relevantes a la Arquitectura

Se especifica a continuación la descripción de los casos de uso relevantes a la arquitectura. La versión completa de los mismos se encuentra en [3].

#### 2.2.1. Iniciar sesión

El caso de uso comienza cuando un usuario desea iniciar sesión en el sistema. Para esto, el usuario ingresa un nombre de usuario y contraseña al sistema. En caso de existir el usuario ingresado y de ser correcta la contraseña para ese usuario; el usuario queda autenticado en el sistema.

#### 2.2.2. Registrar nuevo usuario

El administrador indica que desea dar de alta un usuario. Luego, el administrador completa el formulario con la información personal del usuario a crear. Esta es: el nombre de usuario, nombre completo, teléfono de contacto, teléfono móvil, dirección, fecha de nacimiento, dirección de correo electrónico, organización/es a la/s cual/es pertenece. Además, se puede indicar el o los proyectos en los cuales está involucrado el usuario a

crear. Finalmente el sistema da de alta al usuario y le envía a la dirección de correo electrónico asociada a su cuenta su contraseña de inicio de sesión.

### 2.2.3. Crear proyecto

El caso de uso comienza cuando el actor indica que desea crear un proyecto. Luego el actor ingresa el nombre del proyecto, una descripción. El actor también indica todos los usuarios del sistema que se relacionan con el proyecto que se está creando, indicando además, cual es el rol que cumplen éstos dentro del proyecto.

### 2.2.4. Seleccionar proyecto actual

El caso de uso comienza cuando un usuario desea seleccionar un proyecto para poder trabajar con él. Para eso, el sistema lista los proyectos en los cuales el usuario tiene permisos y el usuario selecciona uno. Luego, el sistema le indica al usuario que se ha seleccionado dicho proyecto como proyecto actual.

### 2.2.5. Crear línea base

El caso de uso comienza cuando el usuario desea crear una nueva línea base para un proyecto. Para esto, ingresa el nombre y la descripción de la nueva línea base, y el sistema crea una nueva instancia de la misma y le asocia un identificador (auto numerado) en forma creciente. Luego el sistema le asocia una copia no modificable de todas las instancias de las últimas versiones existentes de los requisitos, casos de uso y casos de prueba registrados en el sistema.

### 2.2.6. Crear requisito

El caso de uso comienza cuando el usuario desea crear un requisito en el proyecto actual. Para ello, ingresa todos los datos asociados al requisito, y el sistema realiza la validación de los mismos y crea una nueva instancia del requisito con dichos datos.

### 2.2.7. Cambiar estado de un requisito

El caso de uso comienza cuando un usuario desea cambiar el estado de un requisito. Para eso, el sistema lista todos los requisitos registrados para el proyecto actual y el usuario selecciona uno. Luego, el sistema muestra el estado actual del requisito y el usuario puede cambiarlo por otro. Los estados posibles para cambiar un requisito son **En borrador**, **En espera de aprobación**, **Aprobado** y **Rechazado**. Si el usuario confirma el cambio, el sistema crea una nueva versión del requisito con el estado nuevo y registra al usuario que realizó el cambio y a la fecha actual del sistema. Los cambios al estado **Aprobado** o **Rechazado** deben ser realizados en forma unánime por todos los usuarios registrados como interesados en el requisito (incluyendo al solicitante del mismo). Mientras esto no se cumpla, el requisito no cambiará su estado.

### 2.2.8. Crear caso de uso

El caso de uso comienza cuando el usuario desea crear un caso de uso en el sistema. El usuario ingresa los datos correspondientes al caso de uso y el sistema crea una nueva instancia del mismo asociándole un identificador, uno o más requisitos y cero o más objetos GX.

### 2.2.9. Ingresar caso de prueba

El caso de uso comienza cuando el actor indica que desea ingresar un caso de prueba. Luego, el actor ingresa un nombre para la prueba, así como un resumen y una descripción de la misma, precondiciones, postcondiciones, una prioridad, pasos y resultados esperados (todos estos campos son texto libre). El caso de prueba se asocia al proyecto seleccionado actualmente. Además se indica si es un caso de prueba de regresión o no, y si es automático o manual. En el caso de ser automático se lo relaciona con un caso de prueba en GxTest que lo representa. En el caso que no exista un proyecto actual seleccionado se le solicita al usuario que elija uno. Además se eligen los Casos de Uso que estarán relacionados con la prueba. Finalmente, se puede asociar la prueba a uno o más

planes de pruebas ya existentes en el sistema. Luego, se crea una nueva instancia del caso de prueba.

#### **2.2.10. Ejecutar caso de prueba automático**

El caso de uso comienza cuando el usuario desea ejecutar un caso de prueba automático, el usuario selecciona uno de los casos de prueba que desea ejecutar y el sistema abre la interfaz con la herramienta GXTest. Finalmente al concluir la ejecución del caso de prueba, la información obtenida de la ejecución es registrada en el sistema.

#### **2.2.11. Ejecutar caso de prueba manual**

El caso de uso comienza cuando el usuario desea ejecutar un caso de prueba, el usuario selecciona uno de los casos de prueba que desea ejecutar. A continuación el sistema permite al actor ingresar información asociada con la ejecución del caso de prueba, como por ejemplo: tiempo insumido, si paso o fallo, e incidentes registrados.

#### **2.2.12. Crear plan de pruebas**

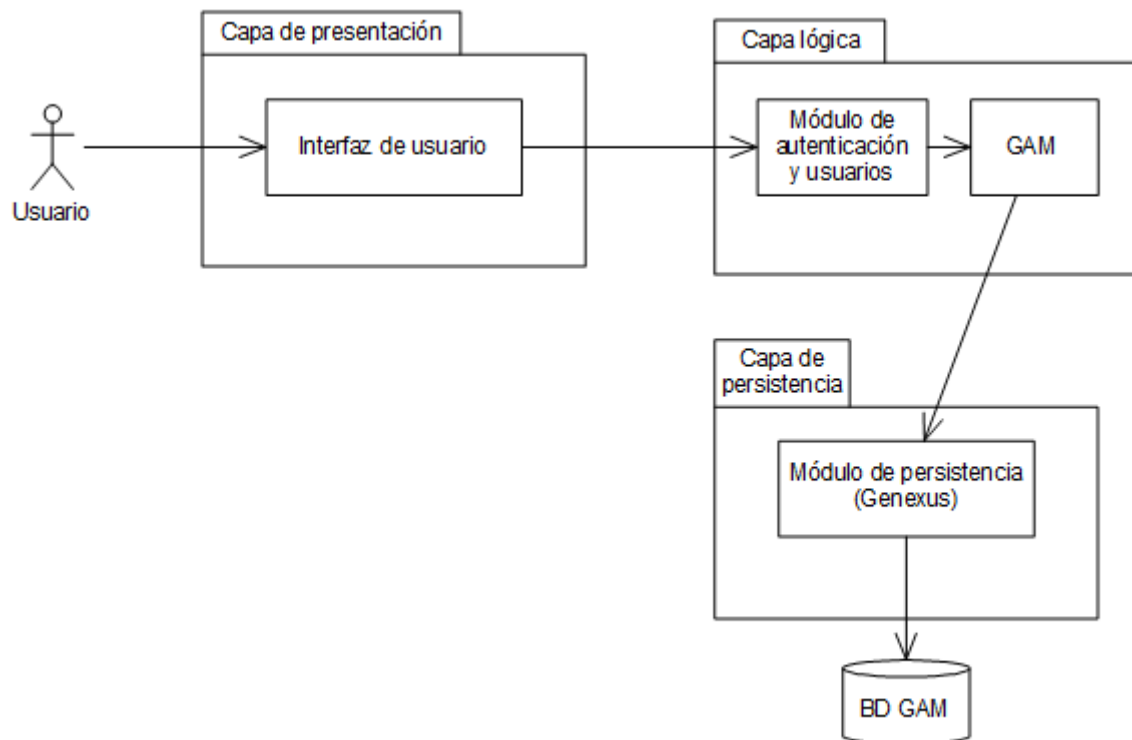
El usuario completa los campos del formulario para crear un plan de pruebas, indicando los datos del mismo, los casos de prueba que van a pertenecer a él, así como su responsable.

#### **2.2.13. Registrar incidente**

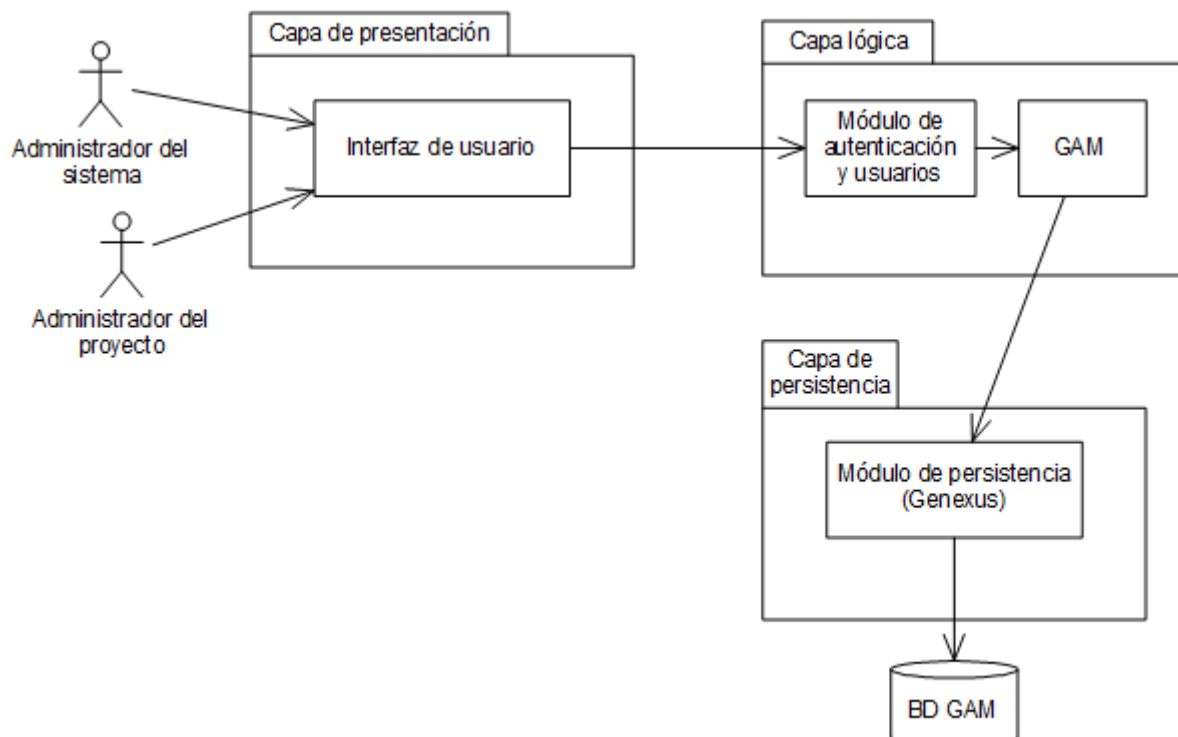
El usuario registra uno o más incidentes ocurridos durante la ejecución del caso de prueba.

### 3. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño

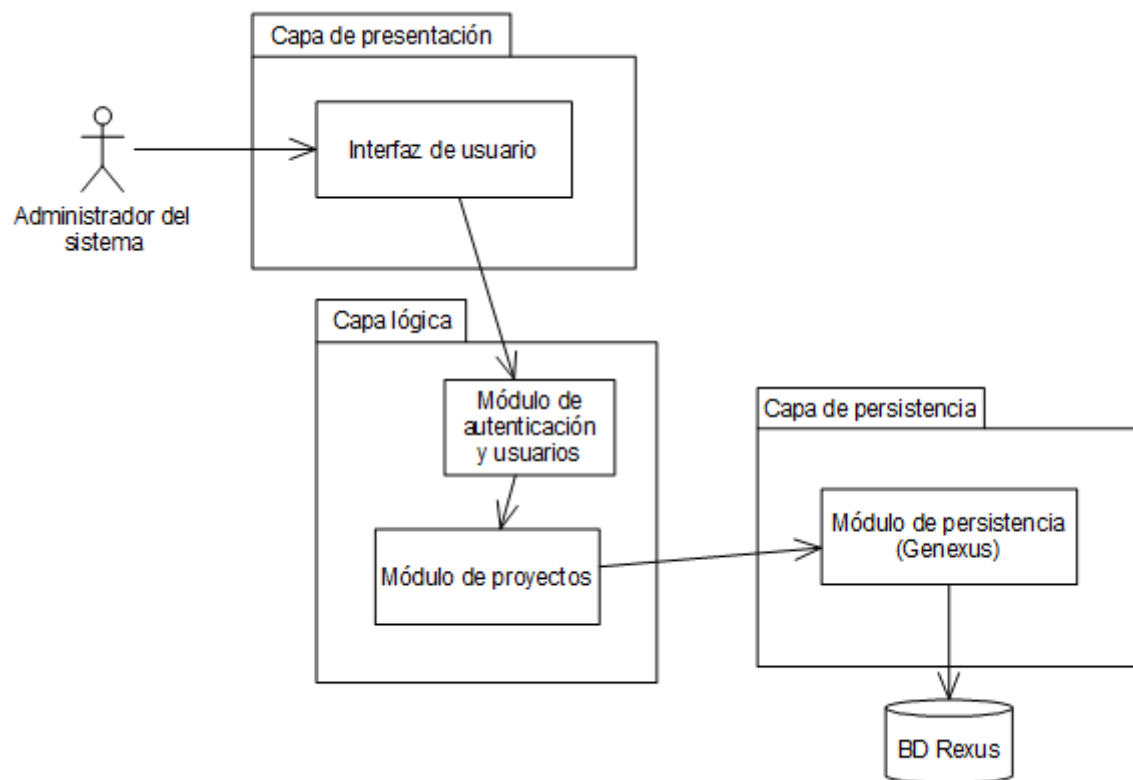
#### 3.1.1. Iniciar sesión

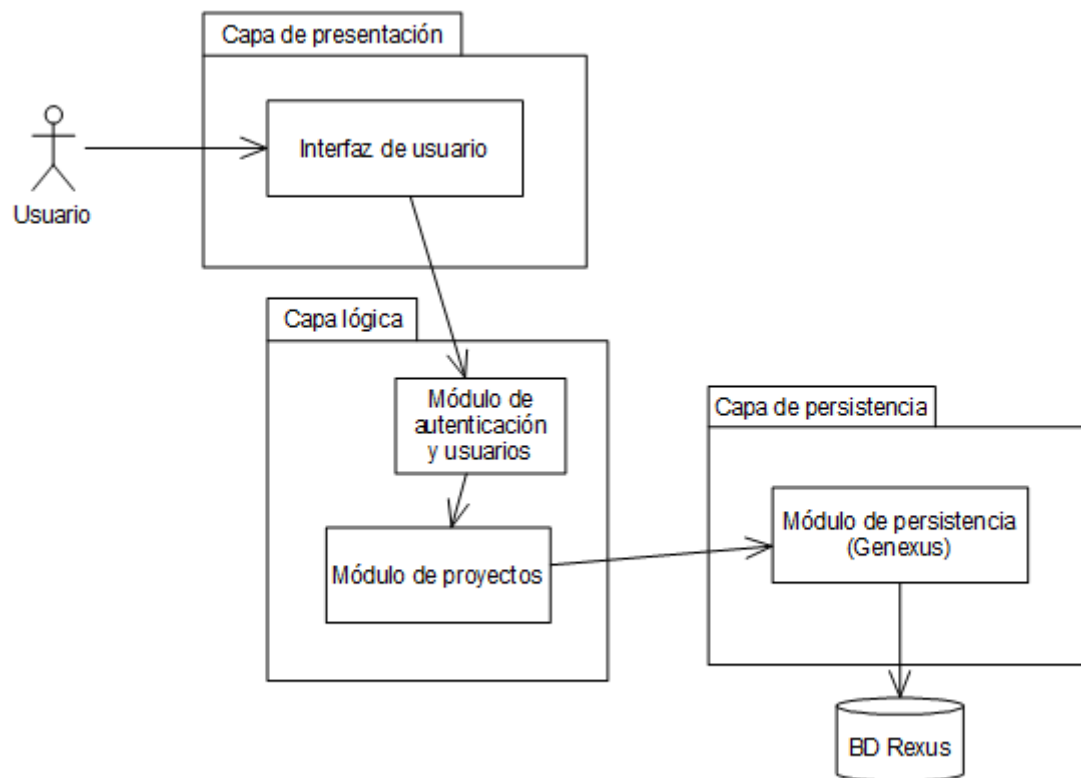


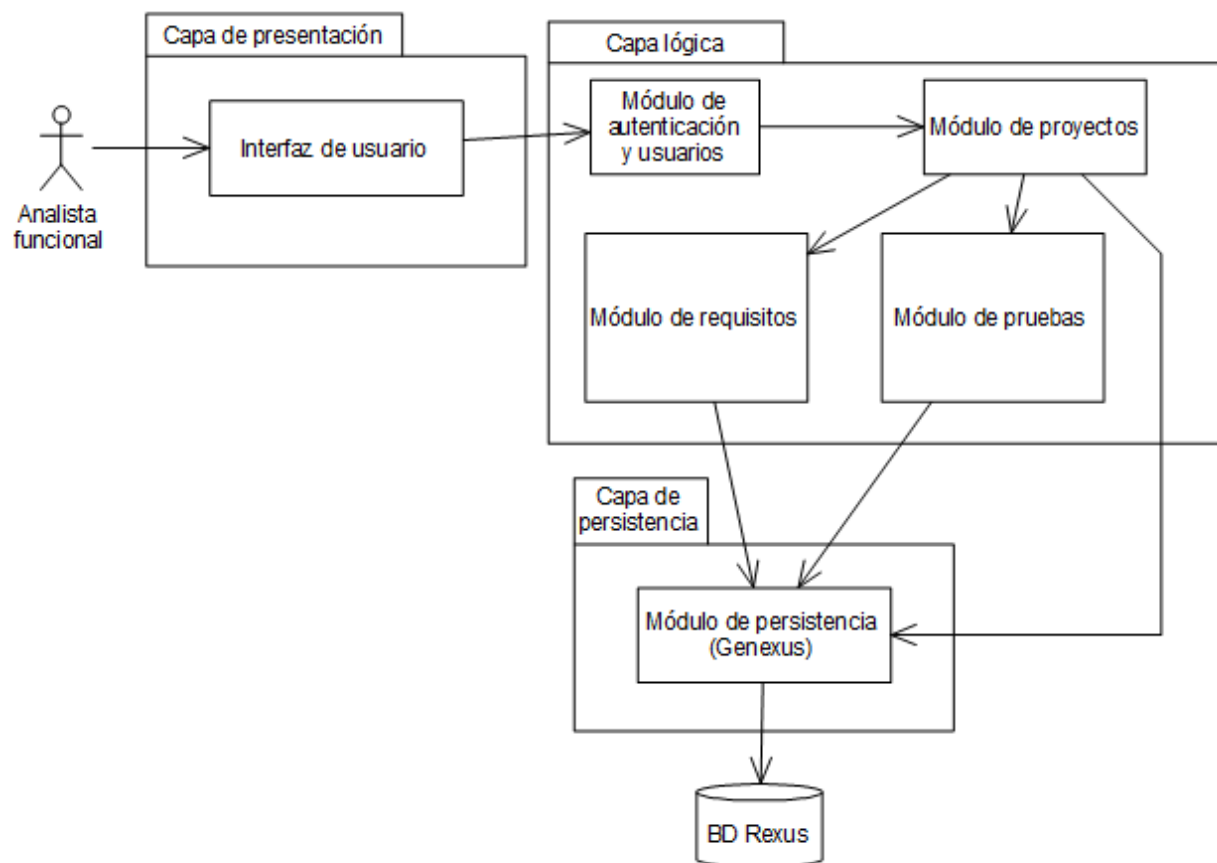
#### 3.1.2. Registrar nuevo usuario

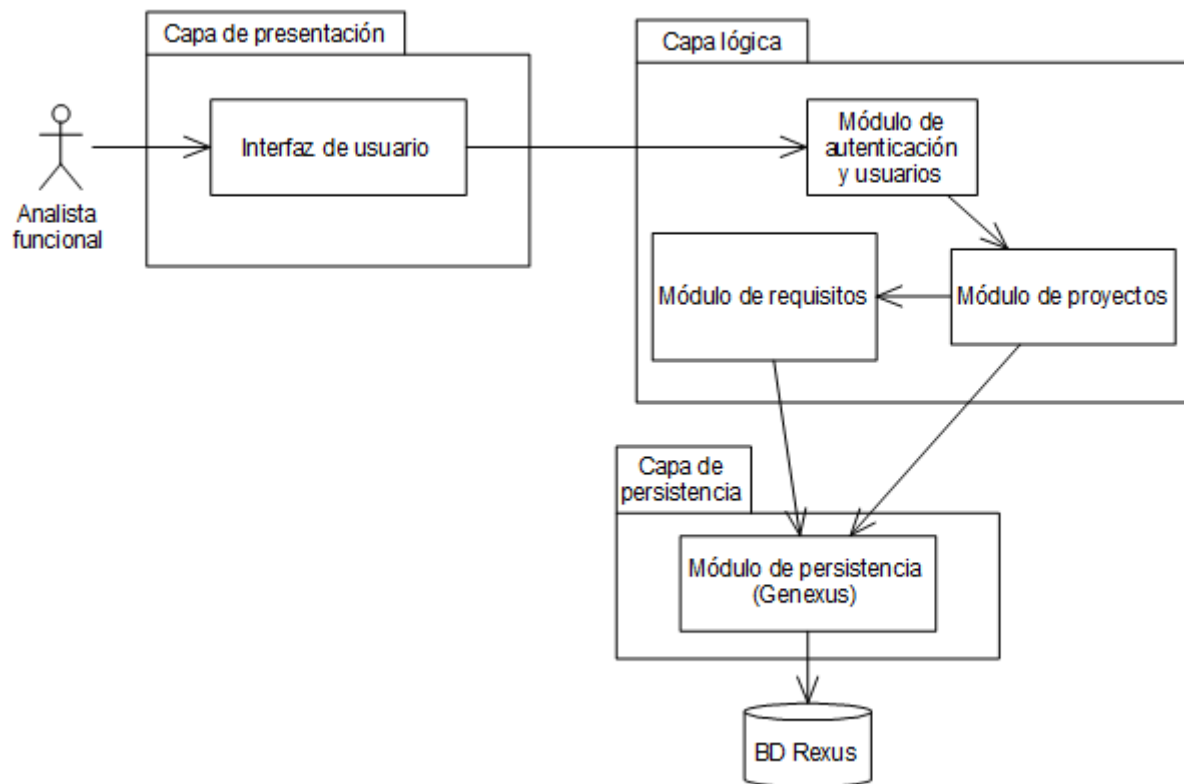




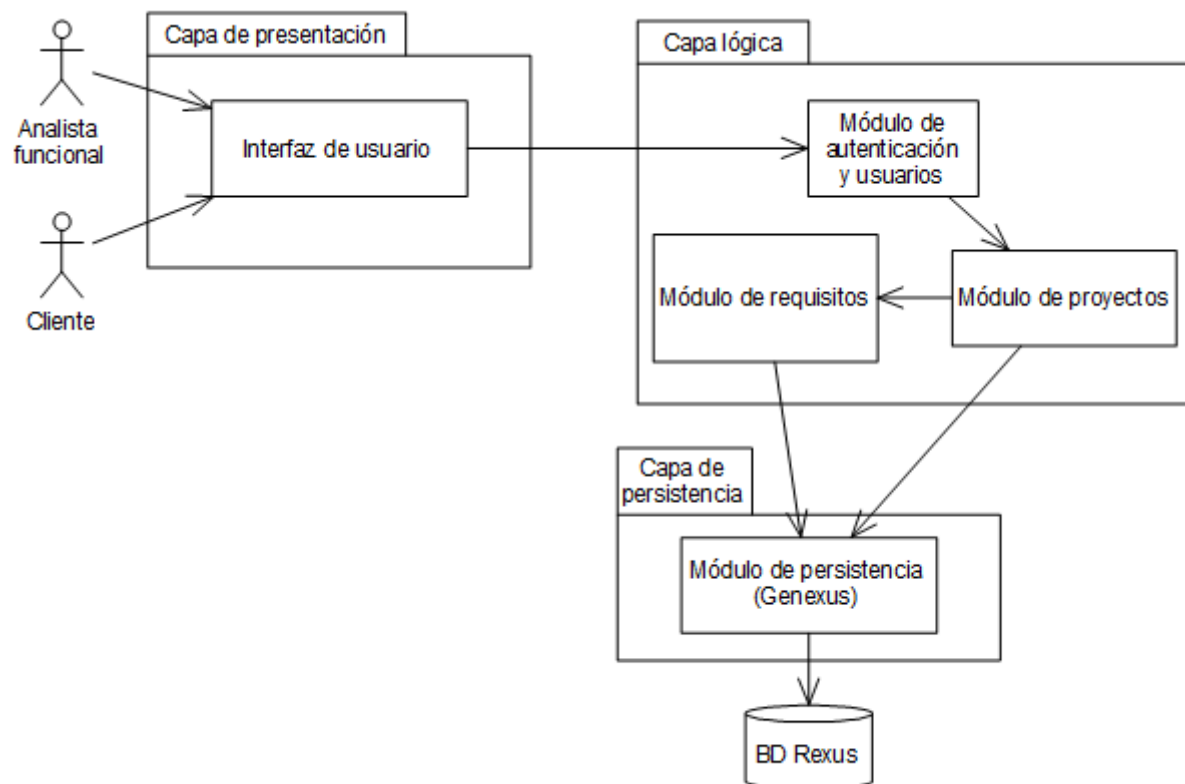
**3.1.3. Crear proyecto****3.1.4. Seleccionar proyecto actual**



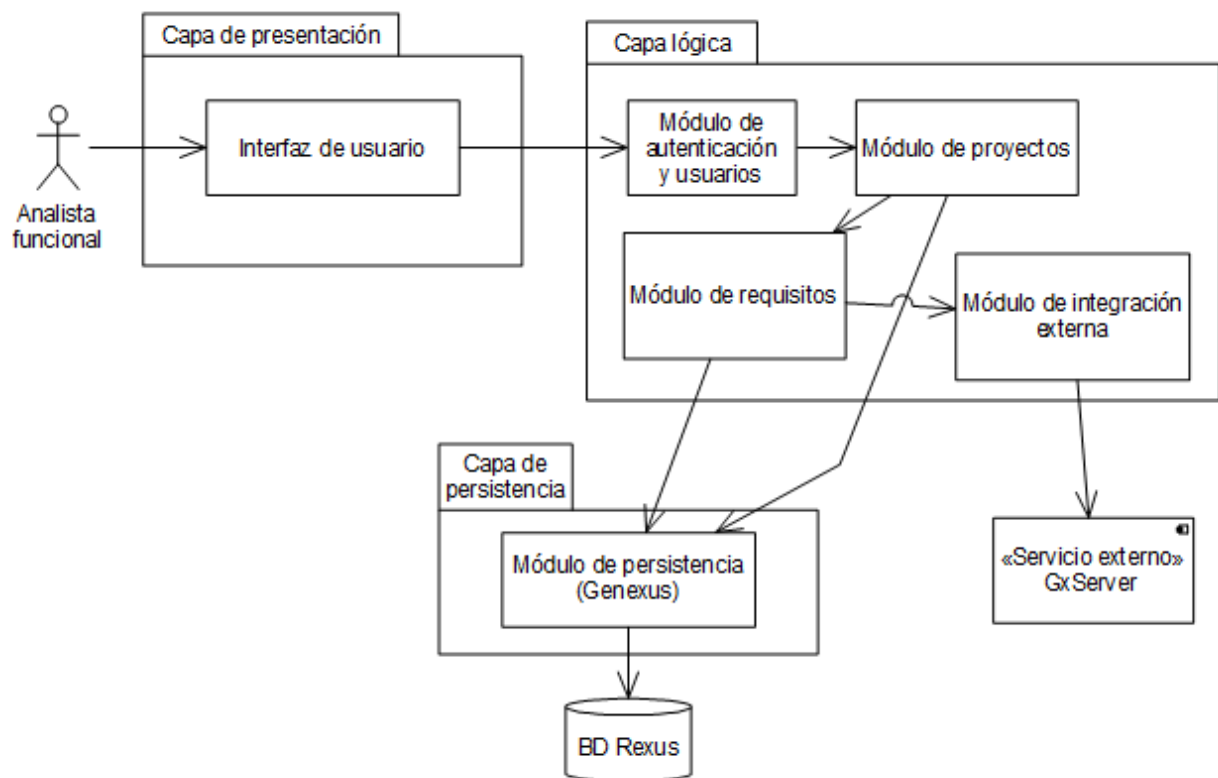
**3.1.5. Crear línea base****3.1.6. Crear requisito**



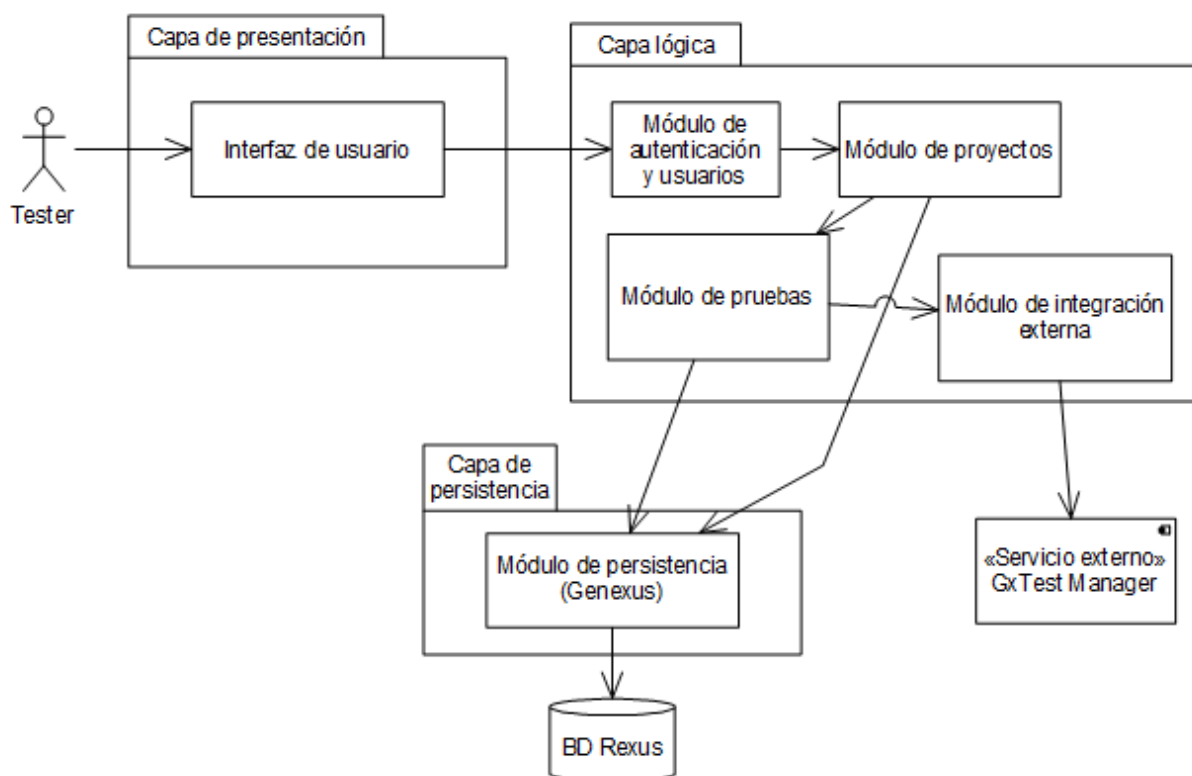
### 3.1.7. Cambiar estado de un requisito



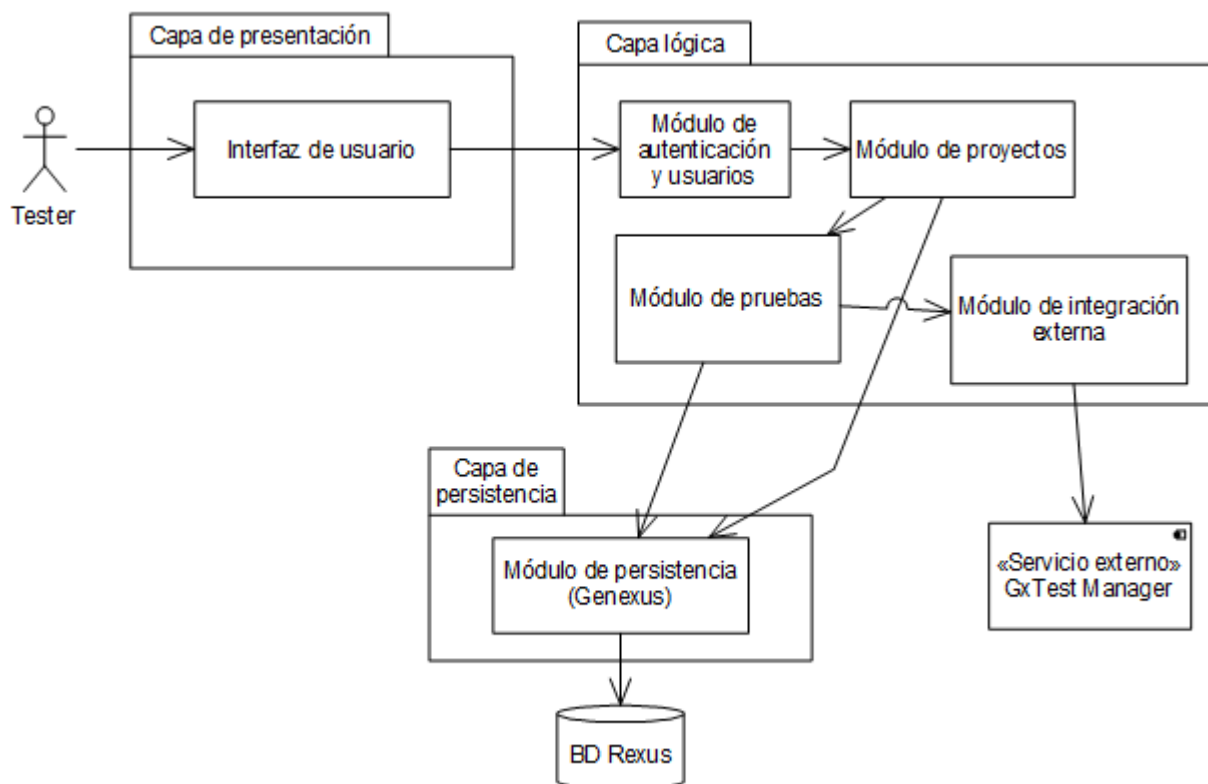
### 3.1.8. Crear caso de uso

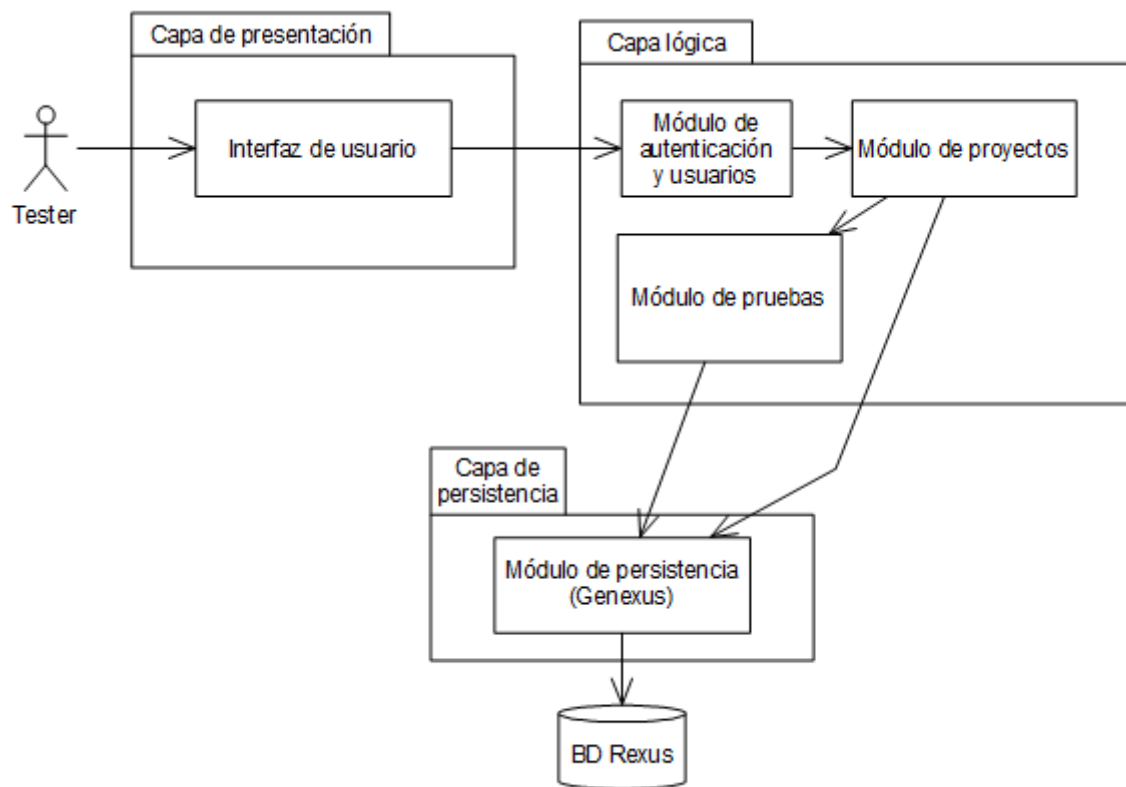


### 3.1.9. Ingresar caso de prueba

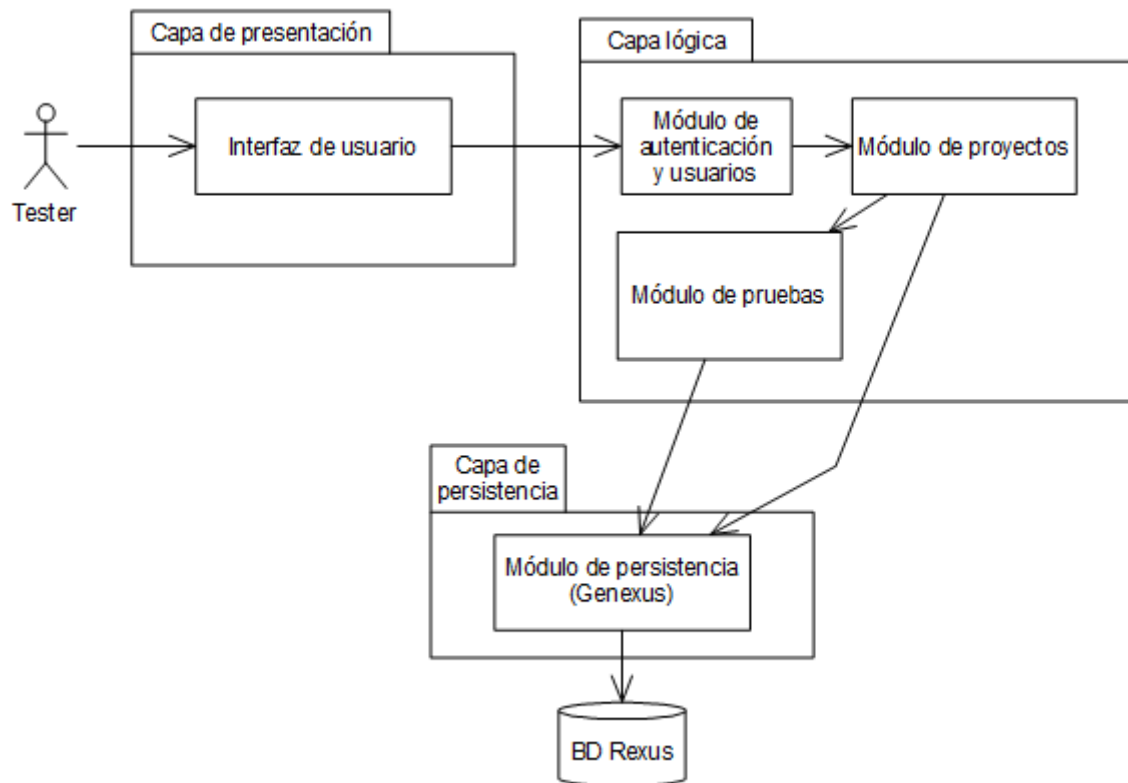


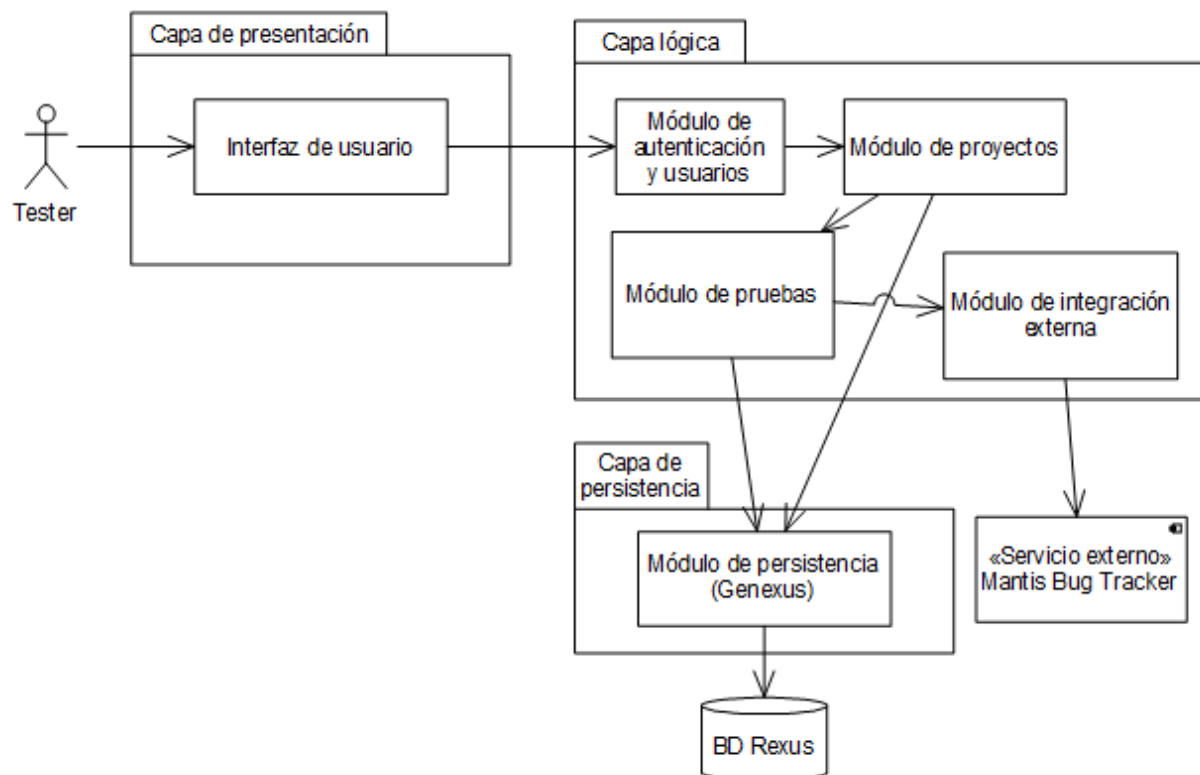
### 3.1.10. Ejecutar caso de prueba automático



**3.1.11. Ejecutar caso de prueba manual****3.1.12. Crear plan de pruebas**

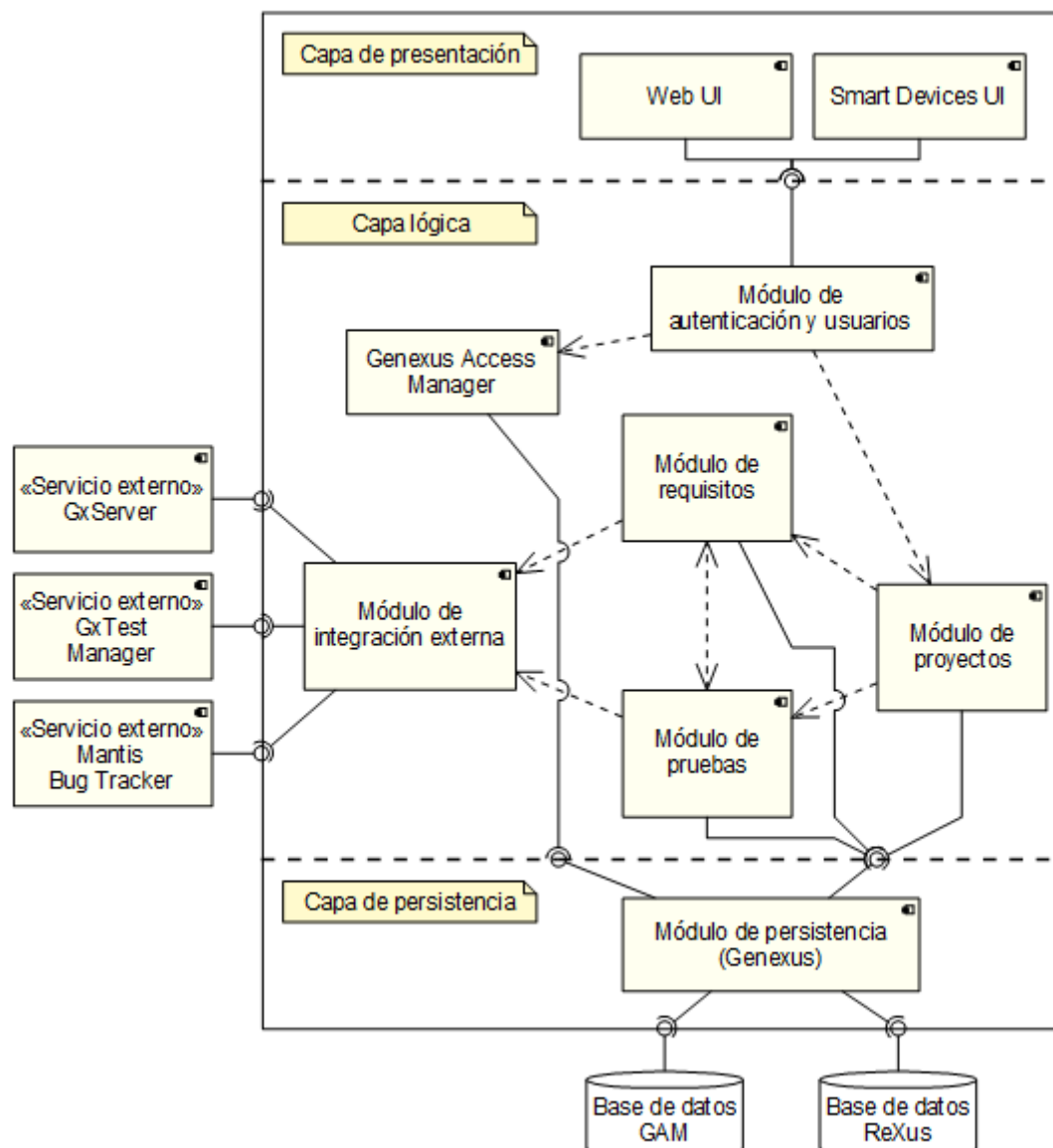




**3.1.13. Registrar incidente**

## 4. Vista del Modelo de Diseño

### 4.1. Descomposición en Subsistemas



#### 4.1.1. Web UI

Es la interfaz de usuario que será provista por medio de un navegador web. Deberá ser capaz de proveer todas las funcionalidades del sistema.

#### 4.1.2. Smart Devices UI

Es la interfaz de usuario que será provista en dispositivos móviles (Smart Devices). Dichos dispositivos móviles podrán ser Android o iOS. La interfaz deberá proveer acceso al portal de requisitos únicamente.

#### 4.1.3. Módulo de autenticación y usuarios

Este subsistema se encarga de realizar la autenticación y autorización de usuarios, manteniendo la seguridad del sistema. Debe conocer qué usuario pertenece a qué proyecto y con qué rol, funcionalidad que deberá ser provista integrando los servicios del Genexus Access Manager (GAM) y del módulo de proyectos.

#### **4.1.4. Genexus Access Manager**

El Genexus Access Manager (GAM) es el subsistema encargado de gestionar los usuarios del sistema, así como de los permisos y roles de cada uno de estos.

#### **4.1.5. Módulo de proyectos**

El módulo de proyectos contiene información sobre los proyectos existentes en el sistema. De cada proyecto, interactúa con el módulo de requisitos y con el de pruebas para obtener los requisitos y los casos de prueba asociados al proyecto en cuestión.

#### **4.1.6. Módulo de requisitos**

El módulo de requisitos contiene toda la funcionalidad que corresponde al manejo de requisitos y casos de uso que hay en el sistema, además de la que permite mantener la trazabilidad entre ambos. También interactúa con el módulo de pruebas para mantener la trazabilidad entre un caso de uso y sus casos de prueba asociados, y con el módulo de integración externa para comunicarse con el sistema externo GxServer.

#### **4.1.7. Módulo de pruebas**

El módulo de pruebas contiene toda la funcionalidad que corresponde al manejo de casos de prueba que hay en el sistema. Interactúa con el módulo de requisitos para mantener la trazabilidad entre casos de prueba y casos de uso, y con el módulo de integración externa para comunicarse con los sistemas externos Mantis Bug Tracker y GxTest Manager.

#### **4.1.8. Módulo de integración externa**

El módulo de integración externa provee servicios para comunicarse con sistemas externos, entre los cuales se encuentran GxServer, GxTest Manager y Mantis Bug Tracker.

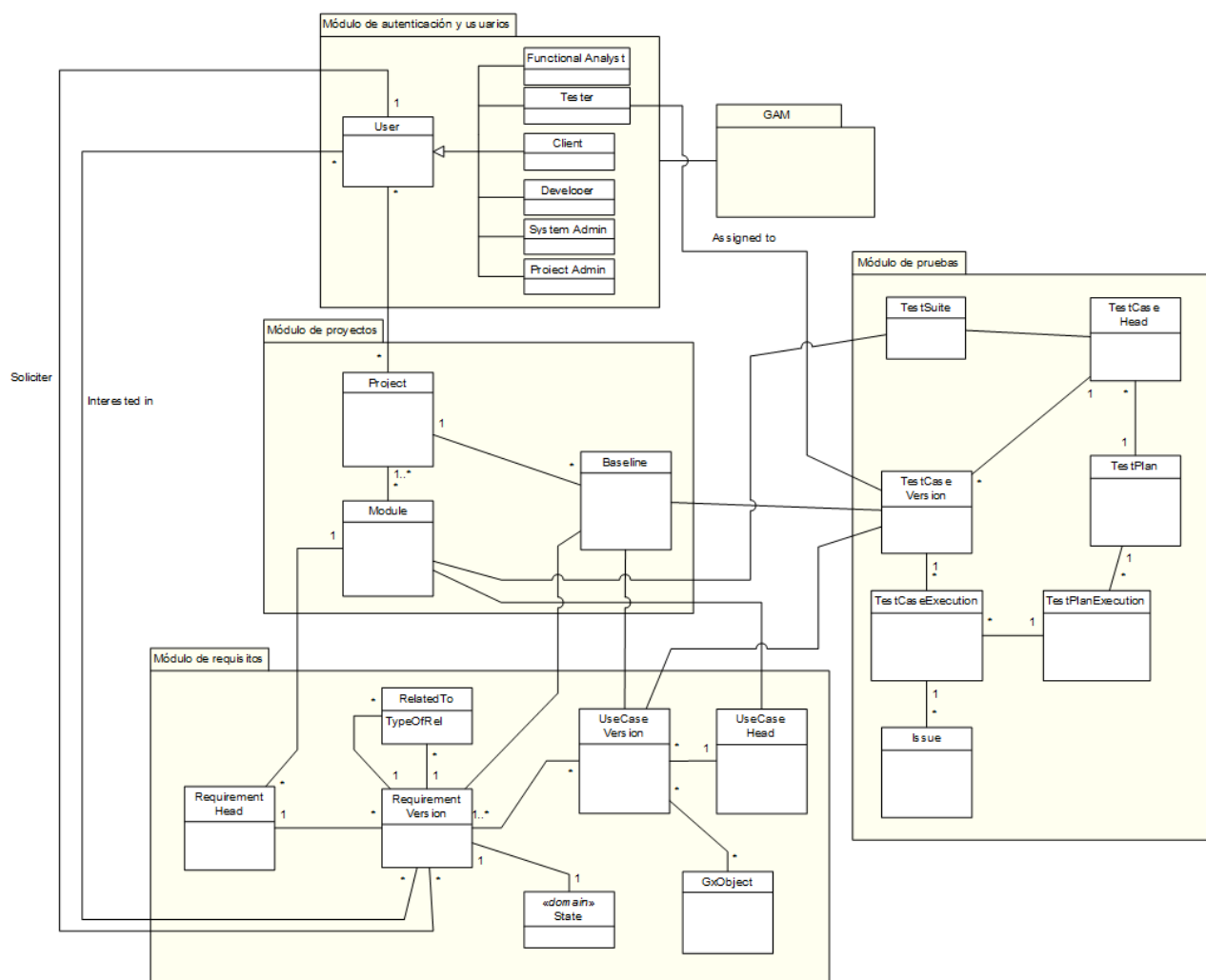
#### **4.1.9. Módulo de persistencia (Genexus)**

El módulo de persistencia de Genexus se encarga de interactuar con las bases de datos para persistir los datos del sistema. Este módulo no será implementado en la práctica puesto que forma parte de las funcionalidades provistas por Genexus.

## 4.2. Diseño de Transacciones

Se presenta a continuación un diagrama de las transacciones más relevantes a nivel de arquitectura. El mismo se encuentra con mayor detalle en el Modelo de Diseño.

Aclaración: la agrupación en paquetes lógicos es artificial, y se perderá cuando se realice el mapeo a transacciones en Genexus. Se provee para facilitar la comprensión del diagrama.



### Módulo de autenticación y usuarios

#### 4.2.1. User

Aquí están los usuarios dados de alta en el sistema.

### Módulo de proyecto

#### 4.2.2. RelatedTo

Se corresponde con el concepto de proyecto. Un proyecto agrupa requisitos, casos de uso, y casos de prueba de forma tal de mantener la trazabilidad entre ellos, además de mantener un histórico de los mismos (versionado).

#### 4.2.3. Module

Los módulos son unidades de agrupación de requisitos, casos de uso y casos de prueba dentro de un proyecto. Permiten al usuario organizar de manera lógica los elementos del proyecto, creando grupos de unidades similares.

#### **4.2.4. Baseline**

Una línea base es un congelado de todas las versiones de requisitos, casos de uso y casos de prueba existentes en un momento dado en el sistema. Los elementos de una línea base no pueden ser modificados.

### **Módulo de requisitos**

#### **4.2.5. RequirementHead**

Esta transacción modela el concepto de un requisito sin importar sus versiones. Contiene toda la información que no cambia del mismo. Está pensada para facilitar el diseño del versionado, y no tanto como una forma de no repetir información, puesto que la inmensa mayoría de los campos de un requisito pueden cambiar de una versión a la siguiente.

#### **4.2.6. RequirementVersion**

Es la versión de un requisito, que contiene todos sus campos y atributos en un momento dado.

#### **4.2.7. RelatedTo**

Es el concepto de que una versión de un requisito está relacionada con una versión de otro requisito. Contiene el tipo de esta relación (que puede ser Bloquea a, Depende de o Relacionado con).

#### **4.2.8. UseCaseHead**

Ídem que RequirementHead pero para casos de uso.

#### **4.2.9. UseCaseVersion**

Ídem que RequirementVersion pero para casos de uso.

#### **4.2.10. State**

Es el concepto de Estado de un Requisito. Los estados de un requisito pueden ser Borrador, En espera de aprobación, Aprobado y Rechazado. Más información en [3].

#### **4.2.11. GxObject**

Es el concepto que representa a los Objetos Genexus.

### **Módulo de pruebas**

#### **4.2.12. TestSuite**

Se corresponde con el concepto de Suite de pruebas, que es una agrupación de casos de prueba, de forma tal de facilitar el manejo de los mismos dentro de un módulo.

#### **4.2.13. TestCaseHead**

Ídem que RequirementHead pero para casos de prueba.

#### **4.2.14. TestCaseVersion**

Ídem que RequirementVersion pero para casos de prueba.

#### **4.2.15. TestPlan**

Se corresponde con el concepto de Plan de Pruebas, que es un conjunto priorizado de casos de prueba planificados para ejecutar posteriormente.

#### **4.2.16. TestCaseExecution**

Esta es la transacción que se ejecuta cuando se realiza una Ejecución de un Caso de Prueba. Aquí se almacenan todos los resultados obtenidos de la ejecución. Se manejan tanto ejecuciones automáticas como manuales.

#### **4.2.17. TestPlanExecution**

Esta es la transacción que se ejecuta cuando se realiza una Ejecución de un Plan de Pruebas, almacenando también todos los resultados obtenidos de la ejecución global.

#### **4.2.18. Issue**

Se corresponde con el concepto de Incidente, que contiene toda la información que se almacena en el caso en que se produzca un incidente. Esta transacción dispara la conexión remota con la herramienta Mantis Bug Tracker, y luego almacena información que permite rastrear y ubicar el incidente allí.

### **4.3. Diseño de Casos de Uso**

El diseño de los casos de uso se encuentra en el modelo de diseño.

## **5. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación**

[Indicar la trazabilidad entre las entidades de diseño del Modelo de Diseño y los componentes del Modelo de implementación.

Indicar para cada entidad de diseño qué elemento en el ambiente de implementación le corresponde, por ejemplo para cada subsistema el módulo que lo implementa; para una funcionalidad importante cual es el conjunto de objetos que la implementan, a qué módulo pertenece.]

## **6. Vista del Modelo de Implementación**

El sistema se encuentra dividido en los siguientes subsistemas:

Módulo de usuarios

- Genexus Access Manager (GAM)
- Módulo de proyectos
- Módulo de requisitos
- Módulo de pruebas
- Módulo de integración externa

Para simplificar el desarrollo, todos los componentes serán implementados en una misma base de conocimiento Genexus.

Esta sección será extendida a medida que se realicen el modelo de diseño y el prototipo del núcleo.

[En esta sección se describe la estructura general del modelo de implementación y la descomposición del software en módulos. Utilice los subtítulos a continuación para esta descripción.]

### **6.1. Subsistemas**

#### **6.1.1. [Subsistema 1]**

[Se detalla brevemente y se muestra la correspondencia con el Modelo de Diseño del Subsistema 1 relevante a la Arquitectura.]

### **6.2. Componentes**

#### **6.2.1. [Componente 1]**

[Se especifica la correspondencia con el Modelo de Diseño del Componente 1 implementado relevante a la Arquitectura. También se debe especificar cómo están organizados los componentes de acuerdo a la estructura y mecanismos de modularización disponibles en el ambiente de implementación y en el(los) lenguaje(s) de programación, y cuales son las dependencias entre los distintos componentes. Pueden existir dependencias de compilación entre componentes, denotando cuales componentes son requeridos para la compilación de un componente específico.]

### **6.3. Interfases**

#### **6.3.1. [Interfase 1]**

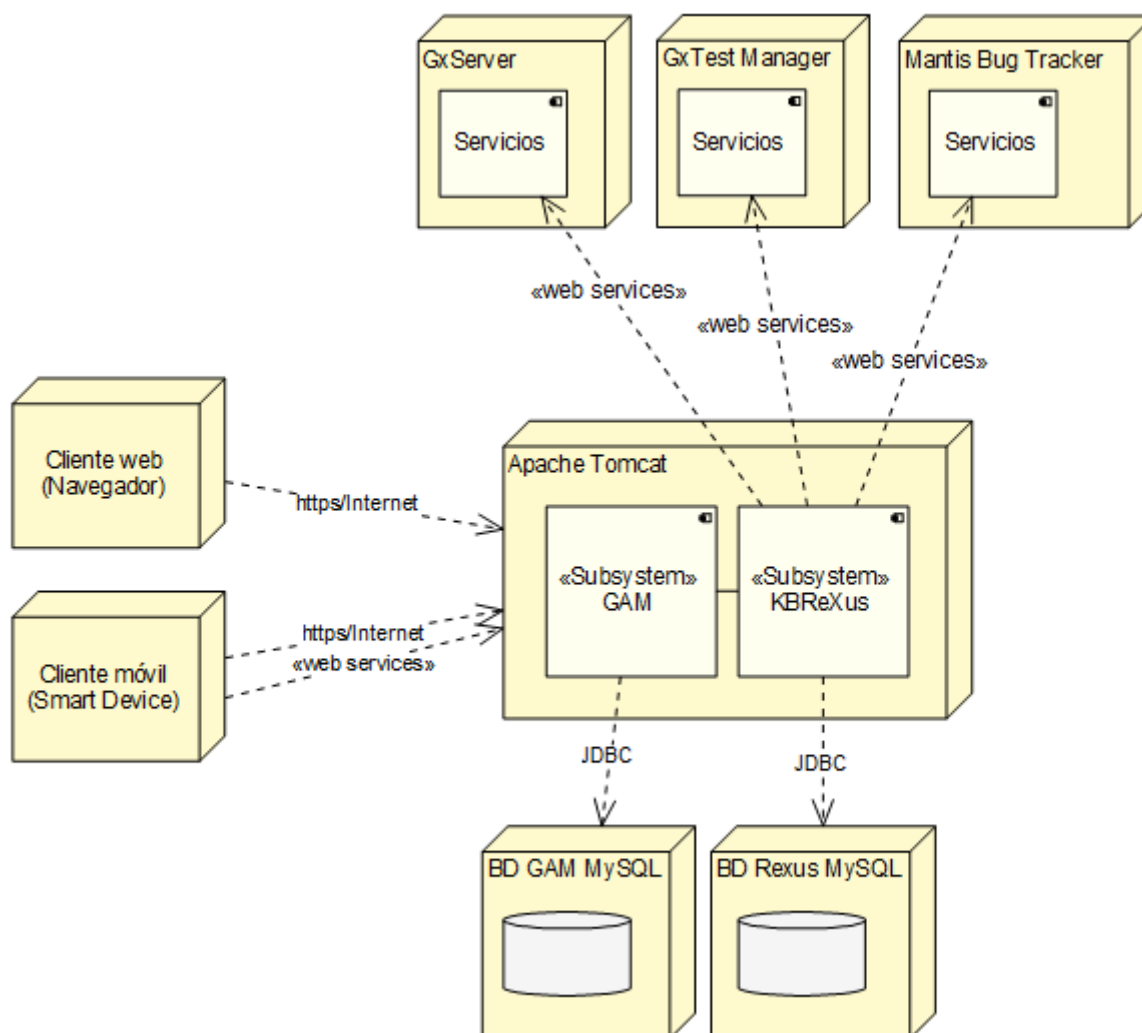
[Se detalla brevemente y se especifica la correspondencia con el Modelo de Diseño de la Interfase 1 relevante a la Arquitectura.]



## 7. Vista del Modelo de Distribución

### 7.1. Diagrama de Distribución

El modelo de distribución describe la distribución física del sistema en términos de como se distribuye la funcionalidad entre los nodos computacionales.



### 7.2. Nodos

En esta sección se da más detalle de los nodos relevantes a la arquitectura del sistema.

#### 7.2.1. Cliente web

En este nodo se ejecuta la interfaz gráfica web que utilizarán todos los usuarios del sistema (Web UI). Es el nodo de uso más común del sistema.

Se corresponde con el módulo Web UI de la vista del modelo de diseño.

#### 7.2.2. Cliente móvil (Smart Device)

En este nodo se ejecuta la interfaz gráfica móvil. A diferencia de la interfaz web, se encuentra restringida, puesto que solamente se podrá acceder a funcionalidades del módulo de requisitos del sistema.

Se corresponde con el módulo Smart Devices UI de la vista del modelo de diseño.

#### 7.2.3. Apache Tomcat

En este nodo se encuentra el servidor web Apache Tomcat que aloja al programa principal Java. Los siguientes módulos de la vista del modelo de diseño se encuentran en este nodo:

**De la capa de autenticación:**

- Módulo de autenticación y usuarios
- Genexus Access Manager

**De la capa lógica:**

- Módulo de proyectos
- Módulo de requisitos
- Módulo de pruebas
- Módulo de integración externa

**De la capa de persistencia:**

- Módulo de persistencia (Genexus)

**7.2.4. BD GAM MySQL**

En este nodo se encuentra la base de datos MySQL que es utilizada por el módulo Genexus Access Manager para almacenar a los usuarios y a los roles. Este nodo está representado directamente en la vista del modelo de diseño.

**7.2.5. BD Rexus MySQL**

En este nodo se encuentra la base de datos MySQL que es utilizada por el sistema Rexus en su conjunto para almacenar toda la información registrada por los usuarios, incluyendo los proyectos, requisitos, casos de uso y casos de prueba, junto a la trazabilidad que se requiere entre todos los elementos además del histórico de versiones de cada uno. Este nodo está representado directamente en la vista del modelo de diseño.

**7.2.6. GxServer**

Sistema externo que almacena bases de conocimiento, y que provee servicios para acceder a ellas y consultar los objetos Genexus que la componen.

Se corresponde con el servicio externo GxServer de la vista del modelo de diseño.

**7.2.7. GxTest Manager**

Sistema externo que almacena casos de prueba automatizados en GxTest y que provee servicios para acceder a ellos y ejecutarlos de manera remota.

Se corresponde con el servicio externo GxTest Manager de la vista del modelo de diseño.

**7.2.8. Mantis Bug Tracker**

Sistema externo que permite crear proyectos y registrar incidentes asociados a ellos, además de brindar servicios para crear nuevos y accederlos de manera remota.

Se corresponde con el servicio externo Mantis Bug Tracker de la vista del modelo de diseño.

**7.3. Conexiones****7.3.1. Conexión https/internet**

Esta conexión se realiza a través de Internet por medio del protocolo https (que garantiza que las conexiones se encuentran encriptadas para mayor seguridad).

La disponibilidad de la conexión depende de las características de la conexión a internet que se disponga. En general, debería ser muy buena (cualquier cliente conectado a internet debería poder acceder sin problemas).

**7.3.2. Conexión por Web Services**

La conexión por Web Services hace uso de alguna tecnología de Web Services por Java, como pueden ser JAX-WS (Java API for XML Web Services) o JAX-RS (Java API for RESTful Web Services). Los sistemas externos ofrecen diversas funcionalidades por este medio, que permiten realizar integraciones con ellos que sean de tipo automática, sin intervención del usuario. La conexión con la versión para Smart Devices de la aplicación utiliza esta conexión también.

### **7.3.3. Conexión por JDBC**

La conexión por JDBC (Java Database Connectivity) consiste en un API que permite la ejecución de operaciones sobre una base de datos desde el lenguaje Java, sin importar de la plataforma desde donde se encuentre. Hace uso del dialecto de SQL propio de la base con que se comunicará (en este caso la base de datos es MySQL). La conexión permite que el servidor web se comuniquen con los servidores que contengan las bases de datos de Rexus y de GAM.