

Diseño e implementación de un repositorio de componentes software para soportar el desarrollo de software empresarial – caso: División de Servicios de Información de la Universidad Industrial de Santander

Fernando Antonio Rojas Morales
M. Sc. en Ciencias Computacionales,
Universidad Industrial de Santander
Docente Tiempo Completo, Investigador Grupo GIB,
Universidad Industrial de Santander UIS
Bucaramanga, Colombia
frojas@uis.edu.co

Robinson Delgado Rojas
M. Sc.(c) Ingeniería de Sistemas e Informática,
Universidad Industrial de Santander
Ing. Diseño y Desarrollo,
Universidad Industrial de Santander UIS
Bucaramanga, Colombia
robdel@uis.edu.co

Fredy Humberto Vera Rivera
M. Sc. Ingeniería de Sistemas e Informática,
Universidad Industrial de Santander
Docente Tiempo Completo, Universidad Santo Tomás,
Investigador Grupo UNITEL, Universidad Santo Tomás USTA
Bucaramanga, Colombia
Freve9@hotmail.com

Resumen— El modelo de desarrollo basado en componentes es un nuevo paradigma de desarrollo que permite la implementación de sistemas rápidamente, a partir de componentes de software previamente desarrollados. El repositorio de componentes es la pieza principal de este modelo de desarrollo ya que permite la administración, selección, identificación, cualificación y ensamblaje de componentes de software.

En este artículo se explica el modelo de desarrollo, y cómo se diseñará un repositorio de componentes para dar soporte al proceso de desarrollo de software empresarial. Para el diseño se tendrán en cuenta todos los requerimientos funcionales y no funcionales que debe contemplar un repositorio de componentes, que soporte el modelo de desarrollo basado en componentes.

La implementación de este repositorio se aplicará a la División de Servicios de Información DSI, de la Universidad Industrial de Santander UIS, como mejora al modelo de desarrollo allí implantado, para optimizar este modelo de desarrollo por componentes, ya que con el repositorio se permite disminuir los tiempos de búsqueda, selección e identificación de los componentes. Además de mejorar los tiempos de respuesta de atención a las solicitudes de mantenimiento y desarrollo que allí se generan, ya que las demoras se disminuyen al máximo al obtener e identificar de forma rápida los componentes que deben ser revisados o integrados en los sistemas solicitados. También se elimina el desarrollo de componentes de forma repetida y descontrolada por la falta de la herramienta que permita monitorear y centralizar los componentes desarrollados.

Palabras clave— Componentes, Ingeniería de Software basada en componentes, Repositorio de componentes.

Abstract— The component-based development model is a new paradigm of development that allows the implementation of systems quickly, starting of software components developed previously. The component's repository is the main tool of this development model since it allows the administration, selection, identification and assembly of software components.

In this paper the development model will be explained, as well as the design of a component's repository to give support to the development process of business software. For this design all the functional and non-functional requirements having component's repository and giving support to components-based development model will be taken into account.

The deployment of this repository will be applied to Information Services Division of Universidad Industrial de Santander UIS in order to improve the development model implanted here, optimizing this development components model. This repository can decrease times of searching, selection, and identification of components. Besides, it shortens response times of attention to the maintenance requirements and the development that these generate. This is due to the fact that the delays decrease to the maximum by getting and identifying quickly the components that it would be verified or integrated in the requested systems. Also, it eliminates the development of repeated and not controlled components due to the lack of a tool that allows overlook and centralizes the components developed.

Keywords— Component, Component – Based Software Engineering, Component's Repository.

I. INTRODUCCIÓN

El modelo de desarrollo basado en componentes es una clase de enfoque que permite el desarrollo de software mediante la existencia de componentes de software preexistentes. Este enfoque conlleva grandes cambios en el ciclo de vida [1]. Entre las ventajas del modelo de desarrollo basado en componentes se encuentran la mejora en la producción de software y la disminución del costo de desarrollo, además de disminuir el tiempo de mantenimiento y pruebas [2].

Ya que los componentes son piezas de software independientes, con una lógica de negocios e interfaces bien definidas para el acceso a sus funcionalidades, se puede pensar en el desarrollo de software como la creación y ensamble de estos componentes, pero para ello es indispensable contar con una herramienta que permita almacenar, catalogar, seleccionar y ensamblar los componentes de interés para el desarrollador [3].

En la División de Servicios de Información se cuenta con un grupo considerable de componentes desarrollados pero no se tiene una herramienta que permita la adecuada implementación del modelo de desarrollo. Por ello se plantea la construcción de un repositorio de componentes que optimice el modelo de desarrollo de la DSI.

Para la implementación de este repositorio se han llevado a cabo varias etapas de carácter obligatorio que permiten identificar las funcionalidades requeridas por la DSI en el repositorio. Etapas como la obtención de los requerimientos de la DSI para implantación del repositorio, funcionalidades que debe prestar el repositorio en el modelo de desarrollo, además de contar con algunas restricciones propias del modelo de desarrollo de la DSI. Para ello se debe identificar el modelo del dominio, capturar los requerimientos de dicho modelo, elegir la arquitectura de software, construcción y población del repositorio entre otras [4].

A continuación se explicará brevemente la metodología utilizada durante el desarrollo de la propuesta.

II. METODOLOGÍA

Para el diseño e implementación de un repositorio de componentes que permita dar soporte al modelo de Desarrollo Basado en Componentes

CBD (Component-Based Development), es necesario conocer plenamente los requerimientos de este paradigma de desarrollo. Para ello se ha realizado un estudio del modelo de desarrollo y su funcionamiento dentro de la División de Servicios de Información, ya que éste será implementado para dar soporte a esta empresa de desarrollo de software.

Como primera medida se ha recolectado la mayor información posible acerca de los procedimientos y falencias del modelo implementado allí, de este estudio particular y el conocimiento acerca del modelo de desarrollo basado en componentes nace un problema que se debe considerar.

A. Problema de investigación

Debido a que el modelo de desarrollo de software de la División de Servicios de información es el CBD y no cuenta con una herramienta que permita la administración de los componentes desarrollados que pueden ser utilizados en el proceso, nace un problema que debe ser resuelto y es el siguiente:

¿Cómo optimizar la disposición de componentes implementados para mejorar el proceso de desarrollo de software en la División de Servicios de Información?

Como parte de la solución de este problema surge la necesidad de un sistema que permita, además de la centralización de los componentes desarrollados, acciones como consultar, seleccionar e identificar componentes que puedan ser integrados en el desarrollo de un nuevo sistema software.

Para determinar la forma de satisfacer las necesidades y/o falencias del modelo de desarrollo de software implementado en la DSI, se debe realizar algunas preguntas que permitan desglosar el problema planteado.

B. ¿Cómo identificar los componentes que cumplen con los requerimientos del desarrollador para su reutilización?

Para identificar un componente debidamente es necesario definir las características relevantes de este, y así crear su ficha de especificación para que el desarrollador pueda utilizarla para la plena identificación del componente requerido.

C. ¿Qué alternativas de arquitectura se pueden utilizar para el desarrollo del repositorio de componentes reutilizables?

Es necesario definir la estructura que se utilizará para la construcción del repositorio, ya que de ella depende la relación entre los componentes registrados y el sistema repositorio.

D. ¿Qué estructura de datos se debe crear con el fin de implementar un repositorio de componentes funcional y completo?

Para implementar un repositorio que cumpla las expectativas y requerimientos de la División, se debe diseñar una estructura de datos que contemple todos estos requerimientos y funcionalidades exigidas, de tal forma que pueda integrarse al modelo de desarrollo de la DSI y permitir la solución de la problemática allí evidenciada.

E. ¿Cómo se debe especificar un componente de software para permitir su plena identificación?

Para especificar un componente de software se debe luego de definir un procedimiento de especificación, registrar su ficha en el repositorio y permitir la consulta de dicha ficha por el desarrollador con el fin de facilitar la identificación o rechazo del componente.

A continuación se definirán algunos conceptos básicos acerca del modelo de desarrollo CBD y su herramienta principal.

III. REPOSITORIO DE COMPONENTES – GENERALIDADES

A continuación se detallará en el modelo de desarrollo, la importancia del repositorio de componentes, sus funcionalidades y beneficios.

A. El Proceso de Desarrollo Basado en Componentes (CBD)

El proceso de desarrollo basado en componentes permite elevar la eficiencia del proceso mismo de desarrollo de componentes. Este proceso puede dividirse en 2 grandes áreas [5], enfocadas básicamente a la producción de componentes, y a la utilización de los mismos dentro del repositorio para la integración de nuevos sistemas.

B. El Repositorio de componentes

El repositorio de componentes es herramienta que permite buscar, hallar y administrar los componentes de software que se utilizarán para el desarrollo de una nueva aplicación de negocios [5]. Un repositorio de componentes debe soportar algunas actividades principales del modelo de desarrollo CBD, tales como: búsqueda, selección, verificación y almacenamiento [5].

En la actualidad existen varios repositorios de componentes, algunos de carácter gubernamental y otros comerciales, como el CRECOR, GIRO y +1Reuse Repository, entre otros [2] [6] [7]. Estos ofrecen algunas funcionalidades, de las que se esperan de un repositorio de componentes, pero no cuentan con algunas funcionalidades que pueden ser importantes para empresas que desarrollan y utilizan dichos componentes, a diferencia de la mayoría de empresas que desarrollan componentes y son creados en su mayoría para utilización o integración por terceros. Entre algunas de estas funcionalidades se pueden observar la búsqueda o identificación por medio de la estructura de datos o tablas que toca el componente en la base de datos, otra funcionalidad importante es la de poder observar un componente en funcionamiento desde el repositorio mismo, ya que éste permite eliminar cualquier duda acerca del componente elegido.

En la construcción de un repositorio se debe tener en cuenta definir la forma de búsqueda de componentes en el repositorio, es decir, definir el sistema de congruencia entre los parámetros de búsqueda y la metadata del componente. También definir las funciones que se van a permitir ejecutar desde el repositorio, además de información que se va a visualizar y grabar en el momento del registro del componente.

C. Funcionalidades Principales de un Repositorio de componentes.

Algunas principales funcionalidades que debe contemplar un repositorio de componentes son:

1) Búsqueda

Esta actividad hace referencia a encontrar dentro de un espacio o un conjunto de componentes aquel que provea las funcionalidades requeridas por el sistema que se desarrolla.

2) Selección.

Los componentes candidatos son hallados, de ellos se toma el que mejor se acomode a los requerimientos y funcionalidades.

3) Verificación

Esta actividad podría ser parte de la selección del componente, ya que contempla actividades de pruebas de funcionalidad del componente separado del sistema.

4) Almacenamiento.

Cuando un componente es terminado y está debidamente probado, este debe poder registrarse en el repositorio, para hacer parte de él, para ello debe registrarse también la metadata del componente junto con su ficha de especificación.

D. Elección de la Arquitectura

En la actualidad existen varias arquitecturas que pueden adoptarse para el diseño de un repositorio de componentes, entre ellas se tienen: modelo-vista-controlador, Arquitectura por capas y Arquitectura orientada a servicios (SOA) [8]. De la elección de la arquitectura depende el funcionamiento y la forma de interacción entre los diferentes subsistemas del repositorio, además de la interacción del usuario y el repositorio en sí.

E. Beneficios

Entre los principales beneficios del modelo de desarrollo basado en componentes se tienen la economía en el tiempo de desarrollo, tiempo para pruebas, tiempo de mantenimiento y la mejora en la calidad del software desarrollado, todas estas debido al desarrollo anticipado y reutilización de software que permite probar y utilizar software una sola vez y ser implementado en varios sistemas con la plena seguridad de su correcto funcionamiento.

Se explicará detalladamente cómo se implementó el repositorio de componentes para la División de Servicios de Información de la Universidad Industrial de Santander.

IV. DISEÑO E IMPLEMENTACIÓN DEL REPOSITORIO DE COMPONENTES UIS

Se explicará en detalle los pasos y requerimientos que se tuvieron en cuenta para el diseño y desarrollo del repositorio de componentes requerido

por la División de Servicios de Información de la Universidad Industrial de Santander UIS. Sección encargada del mantenimiento y desarrollo de todos los sistemas informáticos académicos y administrativos institucionales.

A. Adquisición de requerimientos

Para la División de Servicios de Información de la Universidad Industrial de Santander es de vital importancia que el repositorio de componentes cumpla con una serie de funcionalidades que permitan optimizar el proceso de desarrollo basado en componentes allí implementado, para ello es importante definir algunos requerimientos que este repositorio debe contemplar. Para obtener estos requerimientos es importante tener en cuenta la experticia del grupo de desarrolladores de software de la DSI con el fin de obtener información acerca de qué y cómo debe hacerse para obtener información del componente y qué datos son realmente relevantes en el momento del registro. Adicionalmente, se tiene en cuenta algunos parámetros que si bien no permiten la flexibilidad del sistema, si agilizan el procedimiento de registro y búsquedas. Entre ellos se tienen la utilización de atributos de tipo string en vez de tablas de descripciones, y el uso de campos redundantes para consultas por personas con diferente conocimiento de la programación (tablas – entidades).

Entre ellas están las funcionalidades propias de un repositorio de componentes: almacenamiento, búsqueda, selección e identificación de componentes, pero adicionalmente se desean algunas funcionalidades y adaptaciones propias de la DSI, que no permiten la implementación de un repositorio del mercado, por ello, se recomienda el desarrollo e implantación de un repositorio propio que implemente las funcionalidades requeridas y que adicionalmente proporcione estas funcionalidades particulares propias del modelo de desarrollo de la DSI. Uno de los aspectos que se debe tener en cuenta en el desarrollo del repositorio es la posibilidad de ver en funcionamiento el componente al momento de su selección. Además, por política de la DSI, no se registrarán en el repositorio componentes de diferentes casas de desarrollo, sólo se almacenarán componentes desarrollados por la DSI.

B. Funcionalidades y/o servicios

Dado que es un buen sistema, uno que sea útil, aprovechable, fiable, flexible, accesible y disponible [11], y que el repositorio de componentes de software de la DSI posea estas características, se definieron las funcionalidades y servicios del repositorio de componentes para la División de Servicios de Información servicios las que se consideran propias de un repositorio de componentes típico: registro, almacenamiento, búsqueda y selección de componentes [1], [9]. Adicionalmente se contemplan otras funcionalidades que hacen de este repositorio la herramienta ideal para ser utilizada por los desarrolladores de la DSI, entre las principales se encuentra la opción de observar el componente en funcionamiento desde el repositorio con el fin de disipar cualquier duda acerca de su funcionamiento. También se desea almacenar información acerca de cómo implementar el componente o como se hace el llamado desde el sistema en desarrollo. Para permitir la fiabilidad, accesibilidad y disponibilidad del repositorio, se han seguido algunos estándares de programación y desarrollo establecidos por la DSI como la elaboración de prototipos de desarrollo para plasmar

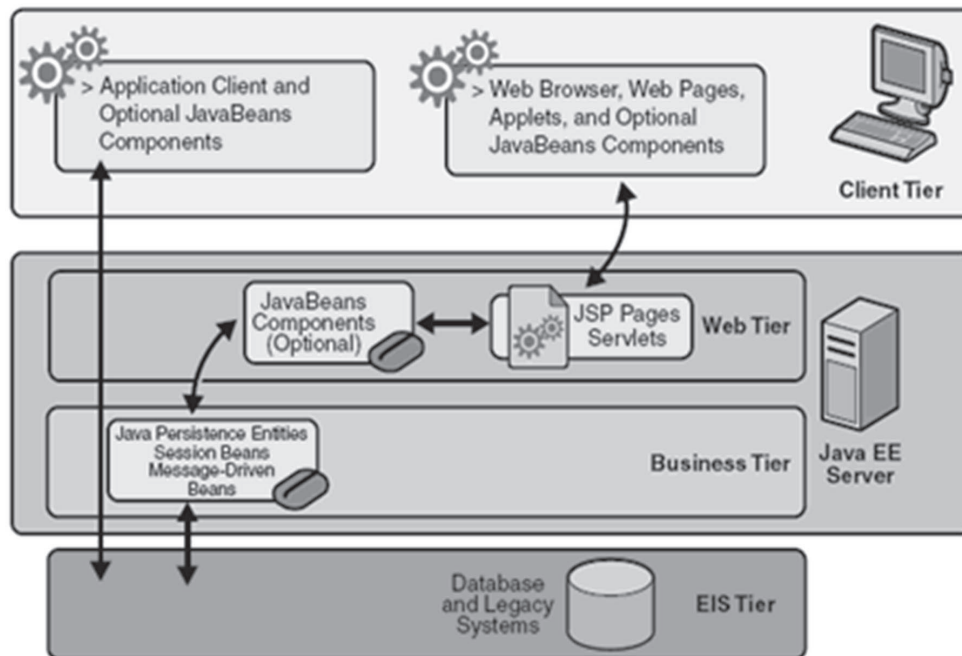
los requerimientos en un modelo y ser validados por el personal encargado de la DSI.

C. Definición de la arquitectura

La selección de la arquitectura es importante, ya que se define cómo funcionará el repositorio internamente, es decir, cómo permitirá interactuar al usuario con sus funcionalidades y la estructura de la base de datos. La Ingeniería de Software Basada en Componentes (CBSE: Component-Based Software Engineering), define varias arquitecturas que pueden utilizarse en aplicaciones de desarrollo de software. Para el caso particular de la DSI, se tomará el mismo modelo arquitectónico implementado para el desarrollo de los componentes, es la arquitectura por capas [8], ya que esta estructura es la básica del desarrollo de aplicación en JAVA EE5, tecnología adoptada por la División de Servicios de Información, estándar en el cual se planteará el desarrollo del repositorio de componentes.

Las aplicaciones Java EE5 constan de tres capas principales: La cliente, la del servidor Java EE, que se subdivide en dos: la web y de lógica del negocio y, la de información empresarial, como se muestra en la Fig. 1.

FIG. 1. ARQUITECTURA POR CAPAS



D. Beneficios del repositorio

El repositorio de componentes es pieza fundamental en la ingeniería de software basada en componentes, ya que permite la centralización, unificación de los componentes y optimización en los tiempos de búsqueda, selección y pruebas de los componentes.

En el proceso de desarrollo de software implementado en la DSI no se cuenta con un repositorio de componentes, razón por la cual la búsqueda e identificación de componentes requería de una buena cantidad de tiempo y no garantizaba la identificación ni existencia real de un componente, ya sea por la falta de documentación o por el desarrollo descontrolado de componentes causado por la necesidad del componente sin desarrollar aún.

Los beneficios que traerá el desarrollo de componentes en el modelo de desarrollo de DSI son: Mejora en los tiempos de búsqueda, selección e identificación de componentes, mejora en los tiempos de desarrollo, ya que no se tendrán que desarrollar nuevos componentes por no encontrarse en un sitio centralizado, mejora en los tiempos de pruebas, ya que las pruebas dedicadas a los componentes no deben hacerse nuevamente, ya que están debidamente probados, mejora la calidad de software, ya que los desarrollados son implementados por las personas que saben de la

lógica del negocio, se optimiza el tiempo de mantenimiento, ya que se mejora la documentación de los componentes y se unifica la utilización de componentes debido a la unicidad de los mismos.

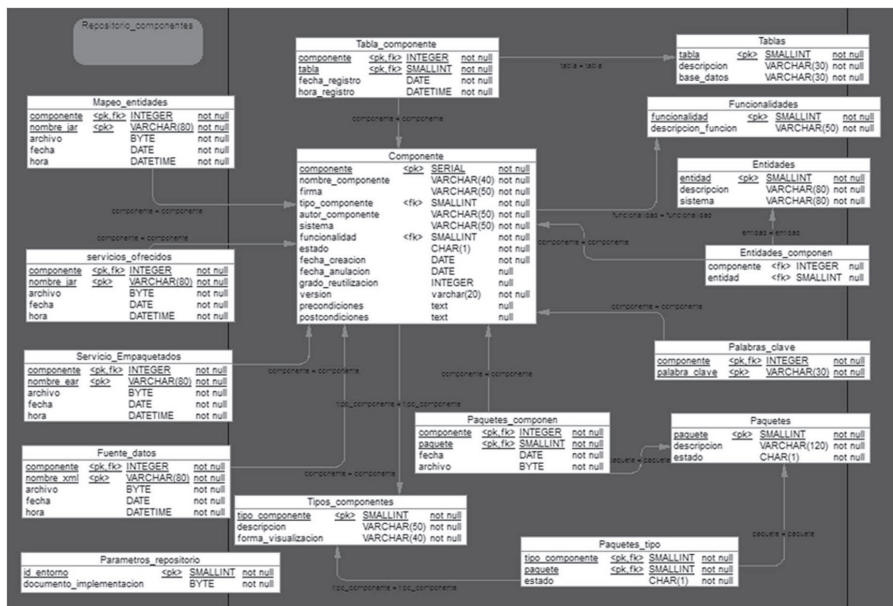
E. Estructura de datos

Para el desarrollo de un repositorio de componentes se deben tener en cuenta las especificaciones y requerimientos definidos para él. El modelo de datos o estructura diseñada debe soportar las funcionalidades definidas y permitir la administración de los componentes. Para el desarrollo del repositorio de componentes de la DSI se diseñó el modelo de datos que muestra la Fig. 2.

El prototipo de repositorio para la División de Servicios de Información contempla las funciones para las que fue concebido, entre las restricciones establecidas figura el almacenamiento de componentes desarrollados sólo por la DSI, el desarrollo de componentes en el estándar JAVA EE5 y la documentación requerida del componente que se contempla en la metadata o en la información para búsqueda e identificación del componente.

Dentro del prototipo se contemplan búsquedas por nombre de la tabla en el modelo de datos, entidades relacionadas, funcionalidad y autor entre otras. Adicionalmente se registrará información relacionada con la implementación del componente.

FIG. 2. ESTRUCTURA DE DATOS DEL REPOSITORIO DE COMPONENTES DSI



Fuente: Autor del proyecto

F. Especificación del componente

La especificación de componente es una parte importante en el modelo de desarrollo, ya que permite la plena identificación del componente, además de guardar la información relevante del mismo dentro del repositorio de componentes.

Esta información permitirá su plena identificación e integración por parte del desarrollador que lo consulte. Los atributos o datos de la especificación se definen según los parámetros definidos para búsqueda, identificación y selección del componente, además de otra información de importancia que se desee guardar desde el mismo momento de registrarlo en el repositorio de componentes.

TABLA I
 ficha de especificación definida para los componentes desarrollados por la división de servicios de información de la UIS

FICHA DE ESPECIFICACIÓN DEL COMPONENTE					
NOMBRE	Nombre del componente	FECHA:	Fecha de creación	VERSIÓN	1.0
DESCRIPCIÓN	Descripción breve del componente, qué hace, las funcionalidades que maneja, dominio de aplicación al cual pertenece.				
ESPECIFICACIÓN DE IMPLEMENTACIÓN					
TIPO VISUALIZACIÓN	Tecnología en la cual fue implementado el componente, junto con sus versiones.				
INTERFAZ	Nombre de la interfaz que define los servicios del componente.				
SERVICIOS	Define la signatura de los servicios que presta el componente.				
SERVICIOS EAR	Define la signatura de los servicios que presta el componente en un ear.				
FUENTE DE DATOS	Define el archivo donde se encuentra el data source del componente.				
PRECONDICIONES	Definen los requisitos que se deben cumplir antes de entrar a utilizar el componente.				
POSTCONDICIONES	Definen los requisitos que se deben cumplir después de utilizar el componente.				
ESPECIFICACIÓN NO FUNCIONAL					
GRADO DE REUTILIZACIÓN	Es el puntaje obtenido por el componente en el análisis de reutilización realizado en el modelo de selección de componentes.				
DISTRIBUCIÓN	Define la forma como se debe instalar y utilizar el componente en las plataformas soportadas.				
ESPECIFICACIÓN DE IDENTIFICACIÓN					
TABLAS	Define las estructuras de datos que están involucradas en el proceso.				
ENTIDADES	Define las entidades relacionadas directamente con la lógica del componente.				
PALABRAS CLAVE	Define o relaciona las palabras que identificarán el componente con los criterios de búsqueda.				
SISTEMA	Sistema bajo el cual el componente fue desarrollado, y por lo cual pertenece su lógica de negocio.				
FUNCIONALIDAD	Define las funcionalidades realizadas por el componente (INSERT, UPDATE).				
AUTOR	Nombre de la persona que desarrolló de componente.				

Fuente: Autor del proyecto

En una ficha de especificación pueden distinguirse varios tipos de atributos, cada uno puede agruparse según su función dentro del proceso de desarrollo, es así que pueden observarse atributos de identificación del componente como nombre, firma, versión y descripción general, entre otros. También se distinguen atributos orientados a la búsqueda y selección, atributos como palabras clave, autor, funcionalidades, entidades u objetos relacionados con el componente entre otros.

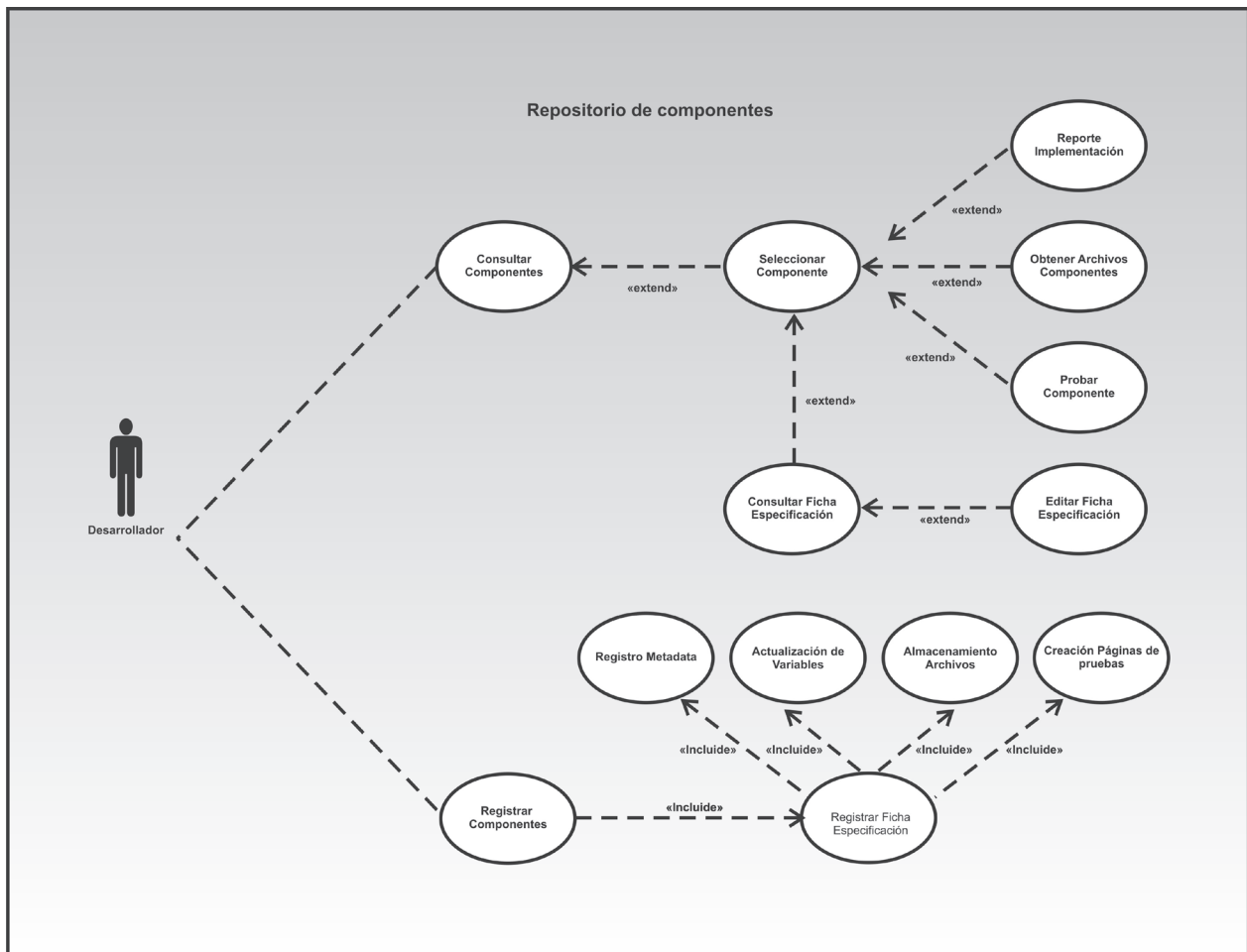
Además existen otros atributos que sirven para documentación complementaria del componente, como precondiciones y postcondiciones del componente o arquitectura de desarrollo del mismo.

Cabe anotar que para algunos desarrolladores ciertos atributos pueden pertenecer a varios de estos tipos o por el contrario pertenecer a otro tipo del que se especificó en este documento.

G. Implementación del repositorio

Para la implementación de este repositorio, se debe tener en cuenta los pasos o actividades anteriores que permiten plasmar las funcionalidades y requerimientos demandados por la División de Servicios de Información. Estas funcionalidades o actividades que debe realizar el desarrollador o programador en el repositorio de componentes se ilustran mejor en el modelo de caso de uso que se muestra en la Fig. 4.

Fig. 4. MODELO DE CASO DE USO DEL REPOSITORIO DE COMPONENTES DSI



Fuente: Autor del proyecto

Para la implementación del repositorio se definen algunos EJB´s que permitirán agrupar las funcionalidades y/o actividades que se podrán ejecutar en el repositorio, entre los que se encuentran: Registro, búsqueda, selección y reportes entre otros, cada una de estos EJB posee funcionalidades propias de su lógica, por ejemplo el EJB de búsqueda cuenta con métodos o funciones de consulta, comparación, consistencia, visualización, inspección y recuperación.

Existe diferencia entre inspeccionar y recuperar, ya que recuperar consiste en identificar y extraer según el criterio de concordancia, e inspeccionar no requiere criterio predefinido [6].

Para las funciones de registro se decide agrupar los componentes en 2 grandes familias que son componentes con interfaz y Componentes de servicios o de recuperación de información.

H. Interface de usuario

La interfaz de usuario permite la interacción con el repositorio de componentes, que habilita las opciones implementadas según el rol del usuario.

En esta interfaz el usuario podrá acceder a sus funciones principales como el registro del

componente, situación en la cual deberá diligenciar la ficha de especificación del componente diseñada especialmente para identificación de cada componente. Adicionalmente podrá consultar los componentes, al acceder a ellos inicialmente por medio de una consulta de criterios de selección, entonces, podrá observar las propiedades y funcionalidades de cada componente mostrado como resultado de la consulta. El repositorio de componentes permite que el implementador observe toda la información de cada uno de los componentes mostrados por la consulta con el fin de permitir al desarrollador identificar plenamente el componente adecuado para su sistema en desarrollo.

Entre la información que el implementador o desarrollador puede visualizar de un componente se encuentran los datos de implementación, como interfaces, servicios y tipos de componentes, además se pueden observar datos no funcionales y de selección e identificación como entidades, funcionalidad, tablas y autor, entre otros.

Ahora se presenta en la Fig. 5, la interface principal de consulta, donde el desarrollador podrá elegir los criterios de búsqueda para realizar la consulta en el repositorio de componentes.

Fig. 5. INTERFAZ DE USUARIO PARA CONSULTA DE COMPONENTES



Fuente: Autor del proyecto

I. Registro de componentes

Antes del registro del componente se debe determinar qué componentes deben ir en el repositorio, esto se hace por medio de los criterios de reusabilidad que nos entregan una calificación del componente [4][8], la cual nos permite determinar si el componente como tal cuenta con un alto grado de reusabilidad. Para el registro y almacenamiento de cada componente en el repositorio es indispensable contar con los datos básicos de la ficha de especificación exigida por el repositorio para el registro, ya que esta información será utilizada para su posterior identificación, además se debe contar con los archivos fundamentales del componente, estos archivos para el caso

particular de los componentes seleccionados por la División de Servicios de Información (EJB), estos archivos son:

1) *JAR de Entidades*

Archivo que contiene el mapeo de las entidades – estructura de datos.

2) *JAR de Servicios*

Archivo donde se implementa o escriben los métodos o funciones que se ofrecen al usuario, estas funciones deben estar claramente documentadas.

3) *JAR de interfaz de usuario*

En este archivo de distribución, se implementan las diferentes interfaces de usuario que el componente mostrará al usuario en el momento de su ejecución, es de notar que no todos los componentes requieren interfaces, ya que algunos de ellos ofrecen el retorno de información no visible para el usuario final, es decir, retornan objetos con información que será tomada por otro proceso o componente.

4) *EAR de servicios*

Los EAR o “Enterprise Archives” son básicamente los JAR servicios empaquetados con archivos META-INF, esto debido a la estructura del Java Application Server” que requiere dicha estructura para su reconocimiento.

5) *Datasource.xml Verificación*

Los archivos de este tipo definen la conexión con la base de datos, es decir, contienen la información del usuario, servidor y base de datos para establecer la conexión.

J. Búsqueda y selección de componentes

Para la consulta dentro del repositorio se contemplan varias áreas de búsqueda, para definir estas áreas, se tuvo en cuenta la distribución y conocimiento de las personas involucradas directamente en el proceso de desarrollo, es así como para algunas personas es importante la funcionalidad misma del componente, para otras es de mayor relevancia las entidades u objetos que se tocan en la lógica del negocio del componente. Para el personal de la DSI la información puede variar según su función dentro del modelo, por ello algunas personas tienen cierto conocimiento con el cual deben hacer las búsquedas, como conocimiento de la estructura de la base de datos, pero no de la forma en que fueron denominados dichos objetos en el proceso de mapeo. Igualmente sucede con las funcionalidades y palabras clave de búsquedas definidas para cada componente.

Según todos estos requisitos, se definieron varios patrones de búsqueda que pueden ser seleccionados por el desarrollador en el momento de consultar el repositorio en busca de algún componente que cumpla con las funcionalidades deseadas, entre los patrones de búsqueda se encuentran palabras clave, funcionalidad, entidades, tablas, autor, sistema y tipo de componente, entre otros. Todos estos contemplados en el diseño del repositorio.

Luego que el desarrollador selecciona los patrones de búsqueda y ejecuta la consulta, el repositorio visualiza al desarrollador la lista de componentes que cumplen con uno o más criterios seleccionados. De esta lista el desarrollador podrá acceder a la información completa del componente con el fin de identificar completamente las funcionalidades del mismo y así elegir o descartar dicho componente. Si el desarrollador desea podrá hacer uso de la funcionalidad de prueba del componente, con la cual podrá ver el componente en funcionamiento, con el fin de descartar cualquier duda acerca del funcionamiento.

V. CONCLUSIONES

De este trabajo se obtuvo el diseño de un repositorio de componentes que permite dar soporte al modelo de desarrollo basado en componentes

implementado por la División de Servicios de Información de la Universidad Industrial de Santander, este trabajo fue elaborado por miembros del Grupo de Investigación en Ingeniería Biomédica de la Universidad - GIIB. El desarrollo de este repositorio se basó prácticamente en suplir las necesidades del modelo de desarrollo de la DSI, y de mejorar sustancialmente algunas falencias del modelo consecuencia de la falta del repositorio de componentes.

La estructura de almacenamiento fue implementada en el motor de base de datos Informix Dynamic Server 11.5, y sus interfaces al usuario en ambiente gráfico para facilitar su utilización y navegabilidad.

Como trabajos futuros se recomienda revisar los procedimientos de la DSI relacionados directamente con el modelo de desarrollo en donde se defina la utilización obligatoria del repositorio en la implementación de los sistemas allí desarrollados, y el desarrollo de los componentes por las áreas de la DSI que conozcan la lógica del negocio sobre la cual debe funcionar el componente, áreas como académica, recursos humanos y financiera, entre otras.

REFERENCIAS

- [1] Ivica Crnkovic, Michael Chaudron, Stig Larson. Component-Based Development Process and Component Lifecycle. Sweden. 2006.
- [2] Jihyun Lee, Jinsam Kim, and Gyu-Sang Shin. Facilitating Reuse of software components using repository Technology. Republic of Korea. 2003.
- [3] Clemens Szyperski, Component Software: Beyond Object-Oriented Programming. Addison-Wesley. 1998
- [4] James Petro, Michael E. Fotta and David B. Weisman. Model-Based Reuse Repository – Concepts and Experience. 1995.
- [5] Jung Eun Cha, Young-Jung Yang, Mun-Sub Song and Hang Gon Kim. Design and Implementation of component repository for supporting the component based development process. Software Engineering Department, Electronic and Telecommunications Research Institute and Dept. of Computer Engineering, Catholic University of Deagu. Korea 2001.
- [6] Hernández Díez Carmen, Laguna Serrano Miguel A, García Peñalvo Francisco J. La biblioteca de Reutilización GIRO. España
- [7] Jiang Guo. Research Associate. Monterey - Luqi. Department of computer Science. Monterey. A Survey of Software Reuse Repositories. USA.
- [8] Vera R, Freddy H. Rojas, Fernando. Artículo. Propuesta de un proceso de desarrollo de componentes software reutilizables. Revista Gerencia Tecnológica Informática. Volumen 7 – Numero 19. Diciembre de 2008.
- [9] Minder Chen, Edgar H. Sibley. Using a CASE Based Repository for Systems Integration. George Mason University. IEEE 1991.
- [10] SUN MICROSYSTEM, The Java EE5 tutorial. 2008.
- [11] STEVENS Perdita, POOLEY Rob. Utilización de UML en ingeniería del Software con Objetos y Componentes. PERSON Addison Wesley. 2ª. ed. Madrid, 2007.