Revista Tecnología y Ciencia
ISSN 1666-6933 DOI: https://doi.org/10.33414/rtyc.33.29-44.2018

Arquitectura y Operatoria de un Sistema de Corrección de Exámenes Automatizado, Utilizando Grafos Dirigidos

María Alejandra Paz Menvielle
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
pazmalejandra@gmail.com

Mario Alberto Groppo
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
proyale@groppo.com.ar

Marcelo Martín Marciszack
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
marciszack@gmail.com

Analía Guzmán
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
analia.guzman@the-group.com.ar

Karina Ligorria
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
karinaligorria@hotmail.com

Martín Cassatti
CIDS-Centro de Investigación, Transferencia y Desarrollo de Sistemas de Información, Departamento de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional, Argentina
mcasatti@gmail.com

Revista Tecnología y Ciencia
ISSN 1666-6933

Presentación 18/11/2016
Aprobación 20/07/2017

Resumen

El presente trabajo describe la arquitectura diseñada y la operatoria implementada para analizar las respuestas escritas por alumnos en forma de texto redactado en lenguaje natural, a preguntas de un examen.

Se muestran las técnicas que permitirán asignar valores a los conceptos y relaciones a fin de poder ponderar la respuesta suministrada por el alumno y compararla con la ponderación de la respuesta base elaborada por un docente.

Estas técnicas consideran todos los casos, incluyendo los distintos grados de acierto que pueda tener la respuesta del alumno, exponiendo los mecanismos con los que se deben analizar los conceptos y las relaciones para obtener la ponderación de la respuesta provista.

Palabras clave: análisis de textos; grafos; detección de patrones; detección de rutas, arquitectura.

Abstract

This paper describes the designed architecture and the implemented operation to analyze the student answers written in natural language text format, to exam questions.

It shows the techniques that will allow assigning values to the concepts and the relationships in order to analyze the response provided by the student and compare it with the weighting of the basic response made by a teacher.

These techniques consider all the cases, including the different degrees of success that the student's response can have, detailing the mechanisms that must be used to analyze concepts and relationships to obtain the weighting of the provided answer.

Keywords: text analysis; graphs; pattern detection; route detection, architecture

Introducción

El presente trabajo forma parte del proyecto de investigación y desarrollo homologado por la Secretaría de Investigación, Desarrollo y Posgrado de la Universidad Tecnológica Nacional, desarrollado en el ámbito del CIDS – Centro de Investigación, Desarrollo y Transferencia en Sistemas de Información, dentro del Departamento de Ingeniería en Sistemas de Información de la Facultad Regional Córdoba de la Universidad Tecnológica Nacional.

El dominio de aplicación seleccionado para la validación de la presente propuesta, se corresponde con los contenidos mínimos fijados para la asignatura

Paradigmas de Programación (Marciszack et al, 2016), pertenecen al bloque de tecnologías básicas dentro del área programación y están principalmente referidos a los paradigmas lógicos, funcional y de orientación a objetos.

La implementación de un sistema de corrección automatizada de exámenes tiene, no solo una complejidad teórica importante, sino que también su implementación práctica es compleja ya que los elementos constitutivos de dicho sistema cuentan con un conjunto de características particulares que no pueden ni deben dejarse de lado.

Una arquitectura correctamente desarro-llada es el elemento fundamental mediante el cual es posible gestionar la complejidad del sistema mencionado y a la vez dejar previstos los fundamentos para su evolución futura.

La arquitectura y operatoria presentadas en este trabajo han sido diseñadas basándose en los grafos conceptuales (Sowa, 1976) y están en proceso de implementación, con el objetivo de lograr la aplicación de las técnicas de grafos, la utilización de bases de datos y herramientas de software actualizadas. Estas técnicas, bases de datos y herramientas facilitan y hacen posible conocer si las respuestas a las preguntas de examen, que requieran respuestas en forma textual, sean o no correctas, a la vez que permite mantener actualizado el dominio de conocimiento de la materia, seleccionada como material de prueba, previendo además el crecimiento de la misma y del sistema y su aplicación a otros ámbitos diferentes.

Desarrollo

Arquitectura

Se ha optado por una arquitectura en tres capas horizontales como se muestra en la figura 1, que modelan niveles crecientes de abstracción y que utilizan interfaces de software definidas para aislar a los distintos módulos de efectos no deseados al realizar modificaciones.

El patrón de diseño utilizado se denomina MVC (Modelo/Vista/Controlador) y es adecuado para realizar la separación de responsabilidades y lograr una mayor flexibilidad a la hora de realizar modificaciones o mantenimiento del sistema.

El patrón MVC establece tres componentes fundamentales:

El Modelo: es el encargado de gestionar el almacenamiento de la información y su accesibilidad para los componentes de capas superiores. En este caso está representado por el componente que gestiona la Base de datos orientada a grafos (GraphDB) y los diccionarios general y personalizado.

La Vista: es la encargada de la presentación de la información a los usuarios y de la recepción de los comandos que se utilizan para interactuar con el sistema. Está representada por los componentes Aplicación y el Graficador de Grafos Dirigidos.

El Controlador: es el contenedor de toda la lógica de negocios o también llamada lógica del dominio. Es el responsable de coordinar la interacción entre los demás módulos, recibir comandos de la Vista, gestionar los datos del Modelo y presentar los resultados, nuevamente en la Vista. Incluye además la lógica de validaciones y la funcionalidad específica del sistema. Está formado por los componentes

Revista Tecnología y Ciencia
ISSN 1666-6933

ManejoLang, LanguageTool (Herramienta para el Lenguaje, librería para análisis ortográfico y gramatical de texto), Gestión de Conceptos, GraphAPI (Interfaces de Programación de Aplicaciones para Gestión de Grafos) y GraphStream (Flujo de Grafos, librería para manipular grafos dinámicos).

En el nivel de Controlador se cuenta con dos niveles de abstracción diferentes según se trate de entidades de bajo nivel, por ejemplo, palabras en un diccionario, o nodos y arcos. Si se está trabajando con entidades de nivel superior, pueden ser términos del dominio, conceptos, relaciones, preguntas o respuestas.

Si bien la separación de responsabilidades se resuelve utilizando el patrón MVC existe una relación entre los componentes del sistema que permite agruparlos lógicamente a través de su funcionalidad, se puede apreciar en forma de grupos verticales en la vista de arquitectura, véase figura 1.

De esta manera es posible relacionar la arquitectura con el diseño y definirlos Módulos Ortografía (Language Tools y Manejo Lang), Gestión de Conocimiento (GraphAPI y Gestión de Conceptos) y Visualización (GraphStream y GraphView).

Esta distribución brinda una gran flexibilidad en la implementación de las mejores soluciones en cada uno de los módulos componentes, manteniendo un bajo nivel de acoplamiento y minimizando los efectos laterales no deseados.

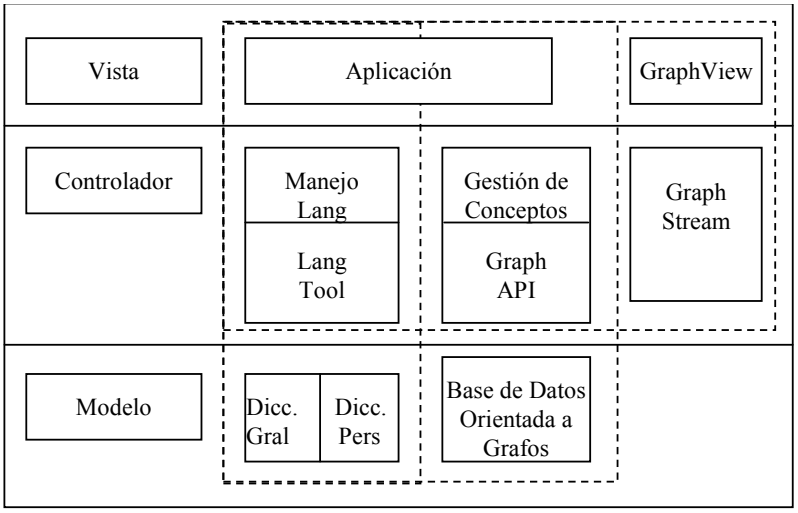

Figura 1 - Arquitectura en tres capas.

Entidades Importantes en la Arquitectura

El sistema de corrección automatizada define algunas entidades que son de uso general en todos los módulos involucrados y que sirven para modelar toda la operatoria del mismo. Estas entidades son:

Término: es el equivalente a una palabra en el texto de la respuesta. Contiene información de gestión asociada al modelo de dominio, por ejemplo: si tiene equivalencias, el tipo de entidad de que se trata en la base de conocimientos (es decir si es concepto o si es relación, tanto simples como compuestas), las sugerencias de corrección en caso de que tenga errores ortográficos, la ubicación en que se encuentra en la oración, entre otros.

Un término contiene, una representación de su información interna en

formato XML, siglas en inglés de eXtensible Markup Language (Lenguaje de Marcado Extensible), para su intercambio con aplicaciones de terceros, que se utilizan en los casos que las mismas estén implementadas en tecnologías diferentes a la del sistema de corrección automatizado.

Todos los términos están compuestos por una y sólo una palabra.

Concepto: es una palabra o conjunto de palabras almacenado en la base de conocimiento y que tiene una equivalencia directa con uno o más conceptos de la materia Paradigmas de Programación.

Concepto Compuesto: está formado por más de una palabra o término. A los fines de la evaluación el concepto compuesto se trata como una unidad indivisible.

Relación: es un arco etiquetado que une dos conceptos. El conjunto Concepto-Relación-Concepto representa la mínima cantidad de información textual que se debe analizar. Una relación tiene sentido únicamente entre dos conceptos, aun cuando un concepto tiene sentido por sí mismo, aunque no participe de ninguna relación.

Relación Compuesta: es una relación que está formada por más de una palabra. Aun así, se procesa como una unidad indivisible. La relación compuesta se utiliza de forma equivalente a la relación.

Estructura: es la mínima unidad que se debe procesar para una evaluación. Está compuesta de dos conceptos unidos por medio de una relación. Hay que destacar que una relación o una relación compuesta, puede unir dos conceptos simples o compuestos, en cualquier orden (Figura 2).

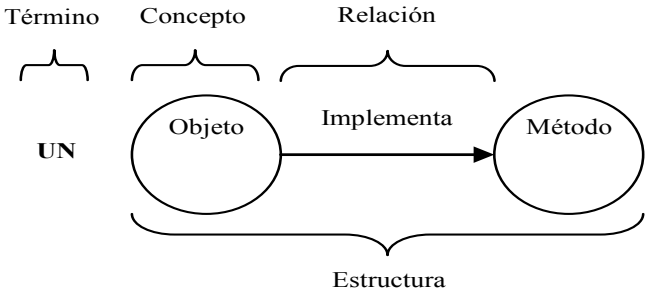

Fig. 2 - Estructura

Flujos de Trabajo

Existen dos flujos de trabajo principales en el sistema de corrección automatizada que plantea el presente trabajo y pueden ser apreciados en la figura3.

El primer flujo es el encargado de procesar una respuesta para asegurarse de que tiene la estructura correcta para ser procesada por el mecanismo de corrección.

Este flujo comienza con la redacción de la respuesta en formato libre, en idioma castellano, dando lugar a un texto que luego se envía al módulo de corrección ortográfica. En caso de que la corrección ortográfica indique que el texto contiene errores, los mismos se reportan y son corregidos por el usuario. El proceso se repite hasta completar la redacción de la respuesta. Una vez validada la ortografía se interactúa con el Módulo Gestión de Conceptos, para la detección de los conceptos existentes en el texto. En este punto se pueden detectar qué conceptos de los

vertidos en el texto de la respuesta se encuentran en la base de datos de conocimientos y cuáles no.

Si se trata de una respuesta ingresada por un docente, para los conceptos que el sistema detectó como faltantes, el docente tiene la posibilidad de realizar consultas sobre los conceptos existentes y sus equivalentes, y/o realizar el alta de los mismos.

Por el contrario, si se trata de la respuesta de examen escrita por un estudiante, los conceptos faltantes serán puntuados con valor 0 (cero).

Para poder construir una ruta que permita evaluar la respuesta, los conceptos deben estar correctamente encadenados, siguiendo la secuencia Concepto-Relación-Concepto.

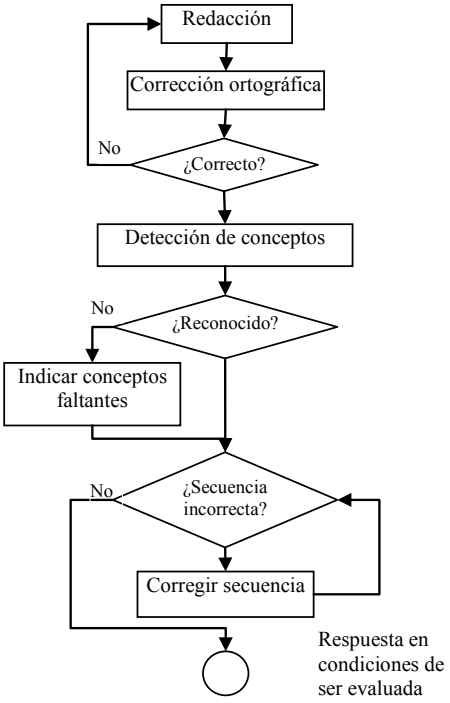

Fig. 3 - Flujos de trabajo.

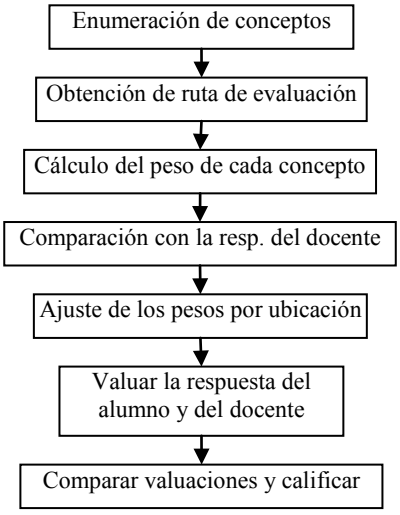

Fig. 4 -Flujo de la respuesta del alumno

Esta secuencia es analizada en el siguiente paso. En caso de detectar un mal encadenamiento se detiene el procesamiento hasta tanto todas las secuencias involucradas sean válidas.

Una vez finalizada esta etapa la respuesta está en condiciones de ser enviada al siguiente flujo de trabajo: evaluación y calificación.

El segundo flujo de trabajo es el encargado de realizar la evaluación y la calificación de la respuesta del alumno (Figura 4), que se realizan a través de los siguientes pasos:

En el primer paso se realiza una enumeración de los conceptos contenidos en la respuesta del alumno. La lista de conceptos se envía al módulo Gestión de Conceptos, que es el encargado de determinar si existe alguna ruta de conceptos y relaciones que incluya los conceptos vertidos por el alumno en su respuesta.

Una vez obtenida la ruta, los conceptos y relaciones se evalúan para obtener el peso de cada uno de ellos, teniendo en cuenta si es un concepto o relación exacta o si es una equivalencia, las cuales poseen menor peso Hopcrof et al, 2008).

A continuación, la respuesta del alumno se compara con la respuesta del sistema y proporcionada por el docente, la que también se encuentra en la base de conocimientos, y se procede a realizar un análisis de la estructura de la respuesta. Esto sirve para ponderar los conceptos teniendo en cuenta el grado de semejanza entre la respuesta candidata (escrita por el alumno) y la respuesta base (dada por el docente), utilizando para ello la ubicación de los conceptos y relaciones, dentro del texto. Aquellos conceptos y/o relaciones cuyas ubicaciones en la respuesta del alumno no coincidieran con la ubicación que tienen en la respuesta base, ven sus pesos calculados de acuerdo a la distancia que hay entre su ubicación en la respuesta candidata con respecto a la respuesta base elaborada por el docente.

Una vez ajustados los pesos, se procede a calcular el valor de cada una de las respuestas (candidata y base). Estos valores se comparan entre sí y el resultado

Revista Tecnología y Ciencia
ISSN 1666-6933

nos indica el grado de acercamiento de la respuesta candidata a la respuesta base, siendo este cociente una forma de calificación que se muestra en la Ecuación 1.

$$C_r = \frac{V_c}{V_b} \quad (1)$$

Siendo Cr la calificación relativa, Vb el valor de la respuesta base y Vc el valor de la respuesta candidata.

Componentes del Sistema

El sistema se compone de cuatro bloques fundamentales que interactúan para obtener el resultado esperado, es decir, la evaluación de una respuesta a una pregunta de examen redactada en texto libre.

Cada uno de los bloques tiene responsabilidades bien definidas que posibilitan un alto grado de especialización y brindan la capacidad, existente en todo sistema de información modularizado de reemplazar un módulo por otro, implementado de manera diferente pero que cumpla con los mismos requisitos de funcionamiento.

Esto último es de fundamental importancia ya que el estado actual del proyecto, como investigación en curso, no está exento de cambios que requieran modificar la implementación o la tecnología subyacente, y frente a este posible escenario es necesario aislar las demás partes del sistema para protegerlas del impacto de estas posibles modificaciones (Joyner et al, 2013).

Los módulos que componen el sistema son:

Módulo Gestión Ortográfica: es responsable de analizar la respuesta escrita, informar los errores ortográficos y sugerir las correcciones necesarias. En este módulo se utiliza una librería de código abierto (Open Source, software distribuido y desarrollado libremente) como base, pero se implementan mecanismos que permiten que la misma reconozca terminología técnica específica de la materia Paradigmas de Programación y considere dicha terminología durante el análisis ortográfico.

Módulo Gestión de Conceptos: realiza la administración de la base de conocimientos y sirve de capa de abstracción a las librerías de gestión de la base de datos de grafos en la que se implementa la base de conocimientos. Utiliza como almacenamiento una base de datos de grafos sobre la que están presentes la persistencia y los mecanismos de consulta necesarios para la evaluación.

Módulo Graficador de Grafos Dirigidos: permite visualizar un conjunto de conceptos y relaciones como un grafo dirigido. Utiliza una librería de código abierto para realizar los gráficos, pero implementa mecanismos propios para establecer el formato visual del gráfico y la forma de reflejar en el mismo las particularidades de la base de conocimiento de la materia y el proceso de corrección.

Módulo Aplicación: ofrece un conjunto de servicios de alto nivel que actualmente se utilizan como una aplicación en sí misma, pero en futuros desarrollos se incluirá una Interfaz de Programación de Aplicaciones para el desarrollo de aplicaciones externas. Este esquema dual de trabajo busca proveer a los docentes con una

herramienta de corrección intuitiva y de fácil uso, además propone un esquema de Software como Servicio (SaaS -Software as a Service) en el cual la infraestructura se hace disponible a través de un conjunto de API's (Interfaces de Programación de Aplicaciones) de alto nivel que posibilitan el desarrollo de aplicaciones cliente que utilicen los algoritmos y datos de la base de conocimientos.

A continuación, se describen las funciones propias de cada módulo así como las interacciones con los demás componentes del sistema.

Módulo Corrección Ortográfica

Un pre-requisito importante para la evaluación automatizada de texto libre es que el mismo esté escrito correctamente según las reglas ortográficas y sintácticas del idioma, para evitar que los mecanismos automáticos malgasten tiempo y ciclos de cómputo intentando analizar términos que no tienen sentido según la lengua castellana (Brookshear, 1993).

El módulo ortográfico es el encargado de efectuar dicha evaluación.

El módulo utiliza una librería de análisis ortográfico implementada en Java, de código abierto, llamada Language Tool, que realiza una revisión general del texto de la respuesta a corregir. Esta librería está extensivamente probada y avalada por la comunidad de usuarios ya que se utiliza, entre otros proyectos de envergadura, como corrector ortográfico y gramatical asociado a los productos de código abierto Mozilla Firefox y Libre Office.

Sobre esta librería se realizan algunas adecuaciones que permiten utilizar terminología técnica concreta, propias del dominio del problema (la materia Paradigmas de Programación), que no forman parte de la lengua castellana pero que deben ser tomadas como correctas por el corrector ortográfico.

La terminología específica de la materia Paradigmas de Programación se almacena en un diccionario personalizado que ampliará el vocabulario pre-existente con un conjunto de términos propios de la asignatura, el que irá creciendo dinámicamente a medida que los docentes vayan incorporando nuevos términos a través de la creación de preguntas y respuestas de examen.

El módulo de corrección ortográfica crea, a partir de las palabras que conforman el texto ingresado, una lista de términos, es decir de las palabras detectadas en el mismo orden en el que se las encontró.

Un	~~ovjeto~~	contiene	~~atrivutos~~
	Objeto		atributos
	Objetó		

Fig. 5 - Lista de palabras sugeridas

Luego se procesan las palabras en busca de errores ortográficos. En caso de encontrarlos se construye una lista de palabras sugeridas como corrección y se la adjunta al término erróneo en la lista. Se puede apreciar un ejemplo en la Figura 5.

Módulo Gestión de Conceptos

El elemento fundamental para el correcto análisis y eventual corrección de una respuesta de examen es la posibilidad de poder detectar los conceptos y relaciones

Revista Tecnología y Ciencia
ISSN 1666-6933

que componen una unidad de información, es decir, una respuesta.

La principal responsabilidad en cuanto al mantenimiento de la base de conocimientos y las consultas necesarias para analizar una respuesta recae en el módulo Gestión de Conceptos.

El mismo sirve como capa de abstracción para las librerías que permiten el acceso a la base de datos subyacente, que canalizan las consultas a la misma, y que además proveen mecanismos de más alto nivel para realizar operaciones sobre entidades de dominio.

El módulo Gestión de Conceptos implementa las siguientes funcionalidades:

Detección de Tipo de Término: el módulo de Gestión de Conceptos es el encargado de recibir la lista de términos que es entregada por el módulo ortográfico, detectar qué tipo de entidad es la representada por cada uno de ellos y anexar dicha información a cada término de la lista de términos. Esto es realizado consultando la base de datos de grafo e indicando para cada término si se trata de un concepto (simple o compuesto), de una relación (simple o compuesta), o si el término no es reconocido como una entidad válida del dominio.

Consulta de Complejos: identifica y marca los términos como compuestos. Para ello se realiza una consulta a la base de datos que obtiene todos los conceptos y relaciones compuestos por más de una palabra. Una vez marcado un término como compuesto, ya sea relación o concepto, es tratado como una unidad indivisible.

Administración de Conceptos y Relaciones: el módulo de gestión de conceptos es el responsable de permitir las operaciones básicas de administración de la base de conocimientos, tales como agregar, modificar y eliminar conceptos, agregar y modificar relaciones, y establecer y remover relaciones entre dos conceptos. El sistema implementa mecanismos de seguridad, por ejemplo, en la eliminación de conceptos, que impiden eliminar un concepto que está siendo utilizado en una relación ya establecida.

Existencia de estructuras: consideran-do que la estructura Concepto-Relación-Concepto es central para el análisis de las respuestas, el módulo Gestión de Conceptos incorpora mecanismos que verifican la existencia o no de una estructura determinada. De esta manera, se le suministran al módulo dos conceptos unidos por una relación, y es consultada la base de conocimientos completa en búsqueda de la estructura suministrada. El usuario será informado por el sistema si dicha estructura ya se existe. Es importante mencionar que el cambio en el orden de los conceptos modifica la estructura en sí, por lo que la consulta de C_1-R-C_2 puede, potencialmente, devolver valores distintos a la consulta de C_2-R-C_1.

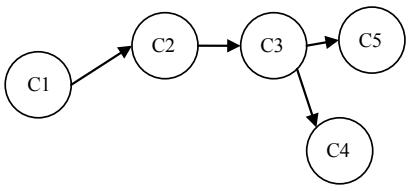

Fig. 6 - Consulta por concepto

Construcción y ejecución de consultas: el módulo Gestión de Conceptos

incorpora una funcionalidad para generar un "query" (o consulta) que obtenga los resultados definidos según un conjunto de criterios, si la misma se ejecuta sobre la base de dato de grafos subyacente. Actualmente hay dos generadores implementados que se están utilizando en la aplicación, ellos son consulta por conjunto de conceptos y consulta por concepto y profundidad.

La consulta por conjunto de conceptos (Figura 6) toma un concepto inicial C_i y un conjunto de conceptos relacionados C_{r1}, C_{r2}, ..., C_{rn} y construye la consulta que al ser ejecutada dará por resultado todas las rutas que, iniciando en el concepto C_i contiene alguno de los conceptos C_{r1}, C_{r2}, ..., C_{rn}

Como se muestra en la Figura 7, la consulta por concepto y profundidad parte de un concepto C_i inicial y recorre todas las rutas que tienen por origen a C_i, y las recorre hasta el nivel de profundidad d indicado.

En el segundo caso no es importante que las rutas sean más profundas que lo especificado, el proceso se detiene al alcanzar la profundidad solicitada.

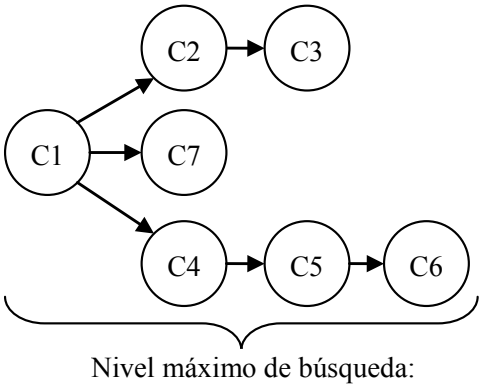

Fig. 7 - Consulta por concepto y profundidad.

Una vez generada la consulta de acuerdo a la sintaxis correcta es posible ejecutarlas sobre la base de datos de grafos.

Ejecución de consultas: las consultas mencionadas anteriormente pueden ser ejecutadas sobre la base de datos de grafos, al igual que otras consultas más simples, utilizadas generalmente para operaciones de administración. El resultado de la ejecución de las consultas mencionadas es una lista de conceptos y relaciones denominados Vértices (Vertex), los cuales conforman una ruta. Si se recorre la lista de vértices se obtiene la ruta que modela la consulta en la base de conocimientos. Hay que mencionar que el módulo encargado de la visualización de los grafos trabaja con listas de vértices.

Módulo Graficador de Grafos Dirigidos

Es el encargado de visualizar una lista de nodos y arcos, incluyendo la información asociada a cada uno de los elementos tal como se obtiene de la base de conocimientos, por medio del módulo Gestión de Conceptos.

C_1	$R_{1,2}$	C_2	$R_{2,3}$	...	C_{n-1}	$R_{n-1,n}$	C_n

Revista Tecnología y Ciencia
ISSN 1666-6933

Fig. 8 - Lista de conceptos y relaciones

El módulo de visualización recibe de los demás módulos, la información necesaria con el formato de una lista de elementos que contiene tanto conceptos (C_1 a C_n) como relaciones (R_1 a R_n), como se muestra en la Figura 8.

Luego lo grafica como se puede apreciar en la Figura 9.

Se utiliza una librería de código abierto, implementada en Java, llamada GraphStream, que provee toda la funcionalidad de visualización y trazado de rutas con una gran flexibilidad y facilidad de uso.

El módulo hace un uso intensivo de las propiedades de ruteo disponibles en la librería GraphView de forma tal que la visualización sea clara y con la menor cantidad de cruces de líneas entre los nodos.

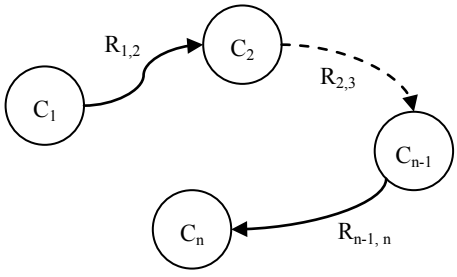

Fig. 9 - Gráfico de conceptos y relaciones

Un detalle interesante es que el algoritmo de distribución (layout) de la librería GraphStream es un módulo reemplazable (plugin) por lo que se puede definir un algoritmo distinto, de acuerdo a las necesidades del sistema, y reemplazarlo por la distribución por defecto.

Se ha ampliado la funcionalidad de GraphStream para permitir que las distintas consultas tengan estilos visuales claramente definidos y que sean adecuados a la visualización que en cada momento se requiera.

Actualmente se realizan tres visualizaciones principales:

Visualización de ruta: se establecen los criterios necesarios, tal como se mencionó en el módulo Gestor de Conceptos y se ejecuta la consulta. El resultado se presenta como una ruta, cuyo origen se encuentra en el nodo Ci, que es resaltado visualmente y se grafican las relaciones y conceptos relacionados, diferenciándolos del concepto inicial (Sowa, 1992).

Visualización de conceptos: se considera un concepto inicial y una profundidad determinada, luego se ejecuta la consulta por concepto y profundidad. Una vez obtenido el resultado, el nodo inicial es graficado y resaltado, utilizando propiedades de texto y color. El resto de los nodos, hasta el nivel de profundidad deseado "d", se grafican utilizando una distribución de tipo "estrella" (Figura 10) que facilita la visualización de los conceptos relacionados.

Visualización de relación: se ejecuta una consulta que obtiene todas las estructuras C1-R-C2, existentes en la base de conocimientos, que contengan la relación R

especificada. El resultado no es un único grafo conexo sino un conjunto de grafos independientes, cada uno conteniendo R (relación buscada). El módulo de visualización grafica esta situación como varios grafos, de pequeño tamaño, y los distribuye de manera equidistante en el espacio de visualización.

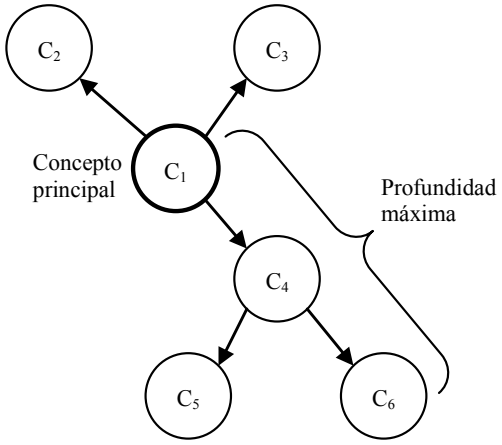

Fig. 10 - Grafico de nodos nivel de profundidad "d"

Aplicación

La capa de aplicación hace uso de las funciones publicadas por las capas de niveles inferiores para unificar todo el proceso en una aplicación que interactúe con el usuario de manera integrada y uniforme.

Esta capa es la responsable de presentar los datos al usuario de manera adecuada. La visualización se realiza a través de Formularios SWI con Java Swing (modelo de ventanas y componentes gráficos de usuario). Aplicando el patrón de diseño MVC se ha logrado desacoplar la capa de aplicación para permitir futuros cambios, como puede ser el desarrollo de vistas web.

La capa de aplicación tiene las siguientes tareas centrales:

Administración de la base de conocimientos, permitiendo a los docentes incorporar nuevas preguntas que posteriormente pueden ser utilizadas para confeccionar un examen.

Adjuntar, a las preguntas mencionadas anteriormente, una o más respuestas "respuestas base" ya que serán consideradas las respuestas correctas al momento de la corrección de un examen.

Es de destacar que cada pregunta puede tener una o más respuestas correctas, redactadas de diferente manera y que involucren diferentes conceptos, a fin de considerar la capacidad expresiva con que cuenta el alumno a la hora de responder.

El docente no debe registrar todas las respuestas posibles que pudiera dar un alumno. Para cada respuesta ideal es posible navegar por diferentes caminos y los

Revista Tecnología y Ciencia
ISSN 1666-6933

términos que la conforman pueden tener diferentes términos equivalentes haciendo posible de esta manera, contemplar las diferentes respuestas del alumno. Caso contrario se trataría de un sistema de "selección múltiple" que no es el objetivo del presente trabajo.

El módulo también tiene como responsabilidad analizar la base de conocimientos y contrastarla contra las respuestas base para determinar si todos los conceptos que intervienen en las mismas ya existen para, en caso contrario, agregar los nuevos conceptos y relaciones. Dicha información se la provee al módulo de gestión de conceptos.

De esta forma el proceso de redacción de respuestas permite validar la consistencia de la base de conocimientos, a la vez que facilita su actualización permanente sin que sea necesario un mantenimiento formal de la misma.

Resultados

La tecnología descripta en el presente trabajo tuvo un papel central en la elección de la arquitectura ya que presenta desafíos interesantes que involucran temas tan diversos como el almacenamiento de una base de conocimientos, la capacidad de consultar de forma rápida el contenido de la misma, la flexibilidad expresiva que esa consulta debe tener, la disponibilidad de herramientas de desarrollo, así como los conocimientos y experiencias de los miembros del equipo de investigación para proponer soluciones factibles y de calidad.

Además, se priorizó el uso de una tecnología que pudiera evolucionar en el tiempo y no quedar obsoleta rápidamente, con el riesgo de perder el conocimiento ya sistematizado almacenado en ella.

Se realizó la implementación en lenguaje Java ya que permite implementar la arquitectura multicapa propuesta y por la fácil intercambiabilidad de componentes que ofrece. A esto debe agregarse la perfecta integración que tiene con la base de datos de grafos, la librería de corrección ortográfica y la de visualización, todas desarrolladas en este mismo lenguaje.

La característica multi plataforma de Java no es secundaria en un proyecto de investigación relacionado con la educación universitaria, en donde es muy factible que distintas unidades académicas cuenten con distintas infraestructuras de hardware y software para implementar una solución de este tipo.

Conclusiones

Se ha explicado la arquitectura de un sistema de corrección automática de exámenes, las preguntas que componen dichos exámenes son el tipo de preguntas que requieren respuestas desarrolladas en formato de texto por el estudiante y escritas en lenguaje natural, es por ello que es necesario analizar la ortografía y la redacción de la respuesta dada.

La arquitectura que se ha presentado implementa la utilización de modernas herramientas de bases de datos, como lo es OrientDB, y se expuso el análisis realizado y las investigaciones que se llevaron a cabo para su adaptación e implementación.

La mencionada arquitectura en tres capas permite trabajar con las distintas partes del sistema logrando la transferencia de información entre capas de manera ordenada y modularizada, con vistas a futuras implementaciones y al crecimiento del sistema, ya que con otro tipo de arquitectura sería muy compleja su implementación.

Se está trabajando y sobre el núcleo de la aplicación en el análisis y desarrollo de una API REST, una interfaz que permitirá interactuar con otras aplicaciones desarrolladas de manera externa.

Se ha elegido utilizar REST porque se ha convertido en un estándar de facto para la comunicación entre aplicaciones en entornos diversos. REST presenta numerosas ventajas entre las que se pueden destacar: es de fácil comprensión, está disponible en casi todos los lenguajes de programación y su tecnología ha sido probada y es eficiente principalmente en entornos de alta latencia como son los existentes en aplicaciones Web ya que precisamente para ellos fue diseñado REST.

La división en tres capas y la incorporación de REST permitirán al sistema ofrecer un conjunto de "servicios" ágiles y ligeros tanto para uso de la cátedra Paradigmas de Programación como para terceros, a la vez que se mantendrá compacto y eficiente el núcleo que contiene la lógica de corrección y la base de conocimientos.

Revista Tecnología y Ciencia
ISSN 1666-6933

Referencias

Marciszack, M., Ligorria, L., Guzmán, A., Corso, C., Tymoschuck, J., Ligorria, K., Castillo, J., Colacioppo, N., Romani, G. (2016). "Material de estudio de la Cátedra Paradigmas de Programación", carrera de Ingeniería en Sistemas de Información, Facultad Regional Córdoba, Universidad Tecnológica Nacional. 6 Unidades temáticas que abordan la totalidad de los contenidos de la asignatura. Disponible para todos los alumnos en el sitio de Universidad Virtual de la Facultad Regional Córdoba, 1-307.

Sowa, J. F. (1976). "Conceptual Graphs for a Data Base Interface", *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Co., como un volumen de IBM Systems Programming Series.

Hopcroft, J., Motwani, R., Ullman, J. (2008). *Teoría de autómatas, lenguajes y computación.* Tercera Edición. Madrid: Pearson Addison-Wesley.

Joyner, D., Van Nguyen, M., Phillips, D. (2013). *Algorithmic Graph Theory and Sage.* Disponible online.

Brookshear, J. G. (1993). *"Teoría de la Computación".* México: Pearson Educación.

Sowa, J. F. (1992) "Conceptual graph summary", en *Conceptual Structures: Current Research and Practice.* New York London Toronto: Ellis Horwood, 3-66.

Sowa, J. F. (2008) "Conceptual Graphs", en *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter (eds.), Elsevier, pp. 213-237.

Bondy, J.A., Murty, U.S.R. (1976). *"Graph Theory with Applications"*. Norh-Holland, 1-36.