Modelo de sistema de recomendación de Objetos de Aprendizaje en dispositivos móviles, caso: Desarrollo del pensamiento computacional

Julio Vera
Universidad Nacional de San Agustín
Arequipa- Perú
jveras.au@gmail.com

Alexander Ocsa Mamani
Universidad Nacional de San Agustín
Arequipa- Perú
aocsa.cc@gmail.com

Klinge Villalba
Universidad Nacional de San Agustín
Arequipa- Perú
kiovic@gmail.com

ABSCTRAC

The information and communications technology favor universal access to education, equality in the exercise of teaching, learning quality and professional development of people and among them are introductory courses in computer science, these are very valuable resources for students from various disciplines. While often the end products of the students observed to judge their competence and not an analysis of how learning, which is the programming process occurs is made, probably due to the difficulty of the track development programming skills of a student. This paper proposes a recommendation system Learning Objects for improved problem-solving skills of novice programmers.

RESUMEN

Las tecnologías de la información y la comunicación favorecen el acceso universal a la educación, la igualdad en el ejercicio de la enseñanza, el aprendizaje de calidad y el desarrollo profesional de los personas y dentro de ellos están los cursos introductorios a ciencias de la computación, estos son recursos muy valiosos para los estudiantes de varias disciplinas. Mientras que a menudo se observa los productos finales de los estudiantes para juzgar su competencia y no se realiza un análisis de cómo se da el aprendizaje, que es el proceso de programación, probablemente debido a lo difícil que es el realizar un seguimiento del desarrollo de las habilidades de programación de un estudiante. Este trabajo propone un sistema de recomendación de Objetos de Aprendizaje para la mejora de las habilidades de resolución de problemas de los programadores novatos.

Keywords

Sistemas de recomendación, E-learning, Machine Learning, Adaptive eLearning.

1. INTRODUCCIÓN

En el dominio de la educación existe gran cantidad y diversidad de material educativo que puede contribuir al proceso enseñanza-aprendizaje, además de la variedad y profundidad de conocimientos previos relacionados al Pensamiento Computacional [6]. En el año 2006 Jeannette Wing publicó el artículo Computational Thinking en el que defendía que esta nueva competencia debería ser incluida en la formación de todos los niños y niñas, ya que representa un ingrediente vital del aprendizaje de la ciencia, la tecnología, la ingeniería y las matemáticas. El Pensamiento Computacional permite que las personas puedan cultivar la capacidad de resolución de problemas, haciendo abstracciones y división de los problemas en otros de menor complejidad para plantear la mejor solución; es aplicada en diferentes áreas del conocimiento; siendo más aplicado en unas áreas que en otras, El objetivo del Pensamiento Computacional es desarrollar sistemáticamente las habilidades de pensamiento crítico y resolución de problemas con base en los conceptos de la computación, tales como abstracción, algoritmos, programación, simulación, entre otros. En la web existen entornos para los cursos donde los estudiantes tienen el control del lugar y tiempo para aprender un tema como son las clases de programación con o sin la presencia del docente[1]. En la actualidad estamos viviendo un tiempo entre quienes usan las computadoras y quienes programan las computadoras[24], y por lo tanto varias disciplinas se están dando cuenta que es muy importante las habilidades de pensamiento computacional y depuración, ya que esta, nos da la facilidad de pensar de manera constructiva sobre cualquier número de tareas[24].

Un objeto de aprendizaje (OA) es "cualquier recurso digital que puede ser utilizado repetidamente para facilitar el aprendizaje" [23]. Los OAs pueden ser utilizados por un estudiante que desee aprender un tema, o pueden ser utilizados por los docentes para elaborar el material educativo que será enseñado a sus estudiantes, para adquirir estos OA, los pueden adquirir a través de consultas en los distintos repositorios que están en la nube, existe tanto material educativo que contienen temas que son muy útiles para unos usuarios y otros no. Todo usuario que realiza una misma búsqueda temática, obtiene como resultado la misma lista de OAs [23].

La forma de aprender conocimientos sobre temas específicos a través de los años ha ido evolucionando partiendo desde el aprendizaje asistido por computadoras, luego los contenidos fueron administrados a través de los sistemas de gestión de aprendizaje, en los últimos años aparecen los MOOCS (Massive Online Open Source) y la tendencia son los sistemas Adaptive elearning que es un nuevo medio de aprendizaje y enseñanza que utiliza un sistema de tutoría inteligente para adaptar el aprendizaje en línea con el nivel de conocimientos del alumno.

El aprendizaje en línea es un camino revolucionario para dar educación en la vida moderna beneficiando a más personas. Un sistema de recomendación es una pieza de software que ayuda a los usuarios a identificar la información de aprendizaje más interesante y relevante de entre un grupo grande de información

educativa [18]. Estos sistemas pueden estar basados en filtrado colaborativo, contenido o híbrido [19].

El objetivo de este trabajo de investigación es crear un modelo de sistema de recomendación de objetos de aprendizaje basado para el desarrollo del pensamiento algorítmico.

2. MARCO TEÓRICO

2.1 Sistemas de Recomendación

Los Sistemas de Recomendación tienen como principal objetivo brindar a los usuarios resultados de búsqueda cercanos o adaptados a sus necesidades, realizando predicciones de sus preferencias y entregando aquellos ítems que podrían acercarse más a lo esperado [9]. Los sistemas de recomendación son los aliados de la personalización de sistemas computacionales, principalmente en la web, por su capacidad de identificar preferencias y sugerir ítems relevantes para cada usuario; para ello se necesita de perfiles que almacenen la información y las preferencias de cada usuario [8].

Aunque existen diversos tipos de sistemas de recomendación, todos necesitan una gran cantidad de información sobre los usuarios y OAs para poder realizar recomendaciones de calidad.

2.1.1 Sistemas de recomendación basada en contenido

En estos sistemas las recomendaciones son realizadas basándose solamente en un perfil creado, utilizan algoritmos "ítem a ítem" generados mediante la asociación de reglas de correlación entre ellos [4].

Este tipo de recomendación aprende de los intereses de los usuarios y hace el proceso de recomendación sobre la base de las características presentes en los ítems. Para este tipo de sistema de recomendación se hace coincidir los atributos del perfil del usuario con los atributos de los ítems a recomendar [11]

Para el caso de los OAs, la recomendación se realiza utilizando los metadatos y una o varias características del perfil del usuario. Este tipo de sistemas tiene como principal limitación, los problemas en las búsquedas cuando se tienen datos poco estructurados y no se puede analizar su contenido, ejemplo en videos y sonidos [8]. Este problema se minimiza al utilizar solamente los metadatos de los OAs.

2.1.2 Sistemas de recomendación basada en filtros colaborativos

Las recomendaciones se hacen basándose en el grado de similitud entre usuarios. Se fundamentan en el hecho de que los OA que le gustan a un usuario, les pueden interesar a otros usuarios con gustos similares [21]. Para la realización de un buen sistema de recomendación colaborativo que ofrezca recomendaciones de calidad, es necesario utilizar un buen algoritmo de filtrado colaborativo, que tienen como objetivo sugerir nuevos ítems o predecir la utilidad de cierto ítem para un usuario particular basándose en las elecciones de otros usuarios similares. Estos algoritmos se clasifican en: los algoritmos basados en memoria y basados en modelos [21].

Los basados en memoria utilizan valoraciones que otros usuarios han dado a un OA, para calcular la posible valoración para el usuario actual y los basados en modelos hacen uso del modelo del estudiante para construir un perfil o modelo, a partir del cual se realizan las recomendaciones [4]. Un usuario de un sistema basó en filtraje colaborativo debe calificar cada uno de los ítems utilizados, indicando cuanto este ítem sirve para su necesidad de información. Estas puntuaciones son recolectadas para grupos de personas, permitiendo que cada usuario se beneficie de las experiencias (calificaciones) de los otros. La ventaja de estos sistemas de recomendación es que un usuario puede recibir recomendaciones de ítems que no estaban siendo buscados de forma activa- La desventaja es el problema del primer evaluador, cuando un nuevo OA es agregado a la federación no existe manera de recomendarlo por este sistema [8].

2.1.3 Sistemas de recomendación basada en conocimiento

Estos sistemas tratan de sugerir objetos de aprendizaje basados en inferencias acerca de las necesidades del usuario y sus preferencias. Se basa en el historial de navegación de un usuario, elecciones anteriores [21].

Un sistema de recomendación basado en conocimiento, hace recomendaciones según el historial de navegación de un usuario, este historial está almacenado con el fin de obtener las preferencias e intereses del usuario y con ello obtener la información necesaria para generar recomendaciones [17]. Estos sistemas también son llamados sistemas de preferencias implícitas ya que deducen las preferencias a partir del comportamiento del usuario y de su historial. Esto permite que en la mayoría de los casos no sea necesario pedir al usuario demasiada información sobre sus preferencias para que pueda ser recomendado. Sugiere ítems basado en las inferencias acerca de las necesidades y preferencias del usuario según su historial de navegación [17].

2.1.4 Sistemas de recomendación híbridos

El enfoque híbrido, busca la unión entre varios enfoques o técnicas de recomendación con el objetivo de completar sus mejores características y hacer mejores recomendaciones [8]. Existen varios métodos de combinación o integración como [8]

- Método ponderado: Donde se combinan las puntuaciones o votos para producir una única recomendación.
- Método de Conmutación: El sistema conmuta entre las técnicas de recomendación en función de la situación actual.
- Método Mixto: Se presentan las recomendaciones de diferentes sistemas de recomendación al mismo tiempo.
- Método de combinación de características de diferentes fuentes de datos se entregan como entradas a un único algoritmo de recomendación.
- Método de cascada: Cada una de las recomendaciones refina las recomendaciones dadas por los otros.
- Función de aumento: Una característica de salida de una técnica, se usa como una característica de entrada a otra.
- Meta-nivel: El modelo aprendido por un sistema de recomendación se utiliza como entrada a otro.

Aunque este tipo de sistemas híbridos tienen grandes ventajas como entregar recomendaciones de alta calidad, su presencia en la web es baja, pero en el modelo propuesto trabajaremos sobre una plataforma móvil.

2.2 Machine Learning

Machine Learning o aprendizaje automático de su traducción al español es un subcampo de la informática que evolucionó a partir del estudio de reconocimiento de patrones y teoría del aprendizaje computacional en la inteligencia artificial [12]. El aprendizaje automático explora el estudio y construcción de algoritmos que pueden aprender y hacer predicciones sobre datos [15]. Estos algoritmos operan mediante la construcción de un modelo de ejemplo, insumos con el fin de hacer predicciones basadas en datos o decisiones [5], en lugar de seguir el programa estrictamente instrucciones estáticas.

El aprendizaje automático está estrechamente relacionado con estadísticas computacionales; una disciplina que también se especializa en la predicción de decisiones.

2.2.1 Programmer's learning machine

Programmer's learning machine (PLM) es una multiplataforma para desarrollar ejercicios que ayudaran a desarrollar los pensamientos algorítmicos. Se le permite explorar diversos conceptos de programación a través de retos interactivos, que pueda solucionar, ya sea en Java, Python o Scala (soporte para el lenguaje C es experimental).

El PLM se sabe que funciona en todos los sistemas operativos (Linux, Mac, Windows). Su interfaz y el contenido se tiene para ser soportado para distintos idiomas inglés, francés, y parcialmente el brasileño e italiano; se espera que se globalice para que pueda ser distribuido a todo el mundo.

En esta plataforma se encuentra más de 200 desafíos, que van desde una introducción completa programación para principiantes reales a temas más avanzados como la recursividad o algoritmos de ordenación y tareas con un alto grado de complejidad de programación.

Se desarrolla esta plataforma machine learning desde el año 2008 para la enseñanza en la escuela Telecom Nancy para sus estudiantes, para su desarrollo como en tecnologías de información y formación como ingenieros de ciencias de la computación. Esta herramienta puede ser utilizada en varias unidades de nuestros planes de estudios, para un total de más de 30h de laboratorios de prácticas. Esto aumenta considerablemente la motivación de los estudiantes, lo que nos permite cubrir un espectro mucho más amplio de temas.

2.2.2 Características de PLM

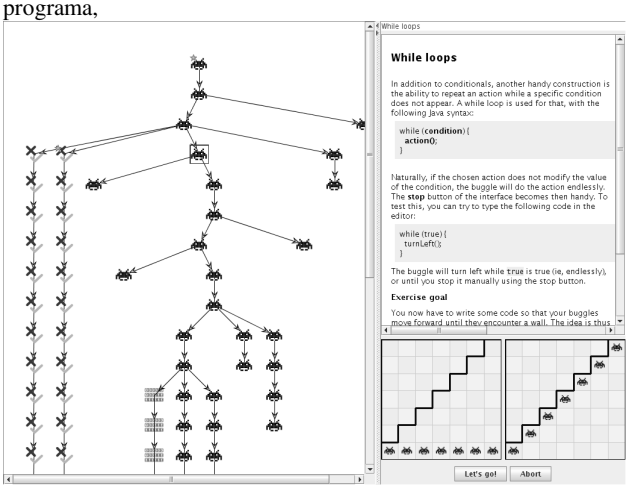
Entre las principales características que se encuentra en PLM son las siguientes:

- La interfaz que cuenta esta multiplataforma está disponible en varios idiomas como son: Inglés, Francés, brasileño, italiano.
- Los ejercicios que se plantea están disponibles en varios idiomas como son: Java, Scala, Python, C, C++, Js y Ruby.
- Modo Demo, paso a paso, sin velocidad de la animación por sesiones, de varios conjuntos de problemas paralelos, para probar mejor el código de estudiante, basándose en la documentación de la herramienta.

2.3 Análisis de código

2.3.1 Análisis de código estático

Se conoce como análisis estático al procedimiento realizado mediante varias técnicas para la obtención de información acerca del posible comportamiento en ejecución de un programa,



realizado sobre el código fuente, código intermedio, código de

Figura 1. Ejemplo 1 de ejercicio para el desarrollo de habilidades de resolución de problemas

máquina o cualquier abstracción de alguna de las opciones anteriores [7].

Su uso, a diferencia del análisis dinámico, se orienta a encontrar violaciones a buenas prácticas en el desarrollo y no a comprobar si el código cumple con su especificación. Este tipo de análisis ha sido aplicado para encontrar errores, malas prácticas y estilos de programación inadecuados; usando diferentes niveles en las posibilidades de configuración, tipos de errores detectados y profundidad, con excelentes resultados [20] [22].

2.3.2 Análisis de código dinámico

El análisis dinámico permite observar el comportamiento del código mientras se ejecuta. De forma general el programa es procesado de acuerdo con un escenario de ejecución y se obtiene información sobre la ejecución de la aplicación, conocida como traza de ejecución, que puede ser o no procesada en tiempo real [7].

Varias estrategias se han aplicado para la obtención de esta información. Entre las más trabajadas en la bibliografía es la instrumentación del programa [3], el uso de depuradores para escuchar y filtrar eventos durante la ejecución [14].

La instrumentación ha demostrado ser un soporte efectivo para la obtención de información sobre la ejecución programas, ya que permite niveles de granularidad importantes con menos esfuerzo en el desarrollo que otras técnicas usadas con el mismo propósito [14]. Su aplicación se basa en tomar un programa o una parte de este e insertar sentencias que generan datos sobre la ejecución de las sentencias originales al momento que se ejecute el código instrumentado. Puede ser realizada sobre el código fuente o archivos compilados, aunque la primera opción es preferible, ya que como indica el trabajo de [17] durante el

proceso de compilación se pierde parte de la información semántica y de forma general en cuanto a exactitud.

2.4 Medidas de similitud

Uno de los métodos preferidos de filtrado colaborativo (CF) recomendadores es utilizar el clasificador k-Nearest Neightbors. Este método de clasificación como la mayoría de los clasificadores y técnicas de agrupamiento depende en gran medida la definición de una similitud o distancia medida adecuada. El ejemplo más simple y común de una medida de distancia es la distancia euclidiana observe la siguiente ecuación:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_k - y_k)^2}$$

Donde n es el número de dimensiones (atributos) y x_k y y_k son los k^{th} atributos (componentes) de los datos del objeto x y y respectivamente.

La distancia de Minkowski es una generalización de la distancia euclidiana obsérvese la siguiente ecuación:

$$d(x,y) = (\sum_{i=1}^{n} |x_k - y_k|^r)^{\frac{1}{r}}$$

Donde r es el grado de la distancia, dependiendo del valor de r, la distancia genérica de Minkowski es conocido como los nombres específicos: Para $r=1$, *la City Block, (Manhattan, taxicab o L1 norm)* distancia; para $r=2$, la distancia euclidiana, para $r \to \infty$ la supremo *(L_{max} o $L\infty$ norm)* distancia, con lo que corresponde a los cálculos de las distancias máximas entre algunas dimensiones de los datos de los objetos.

2.5 Neares Neigbord

Los clasificadores basados en instancia funcionan almacenando registros de capacitación y utilizarlos para predecir la etiqueta de clase de casos que no se ven. Un ejemplo trivial es la llamada *rote-learner*. Este clasificador memoriza todo el conjunto de entrenamiento y clasifica solamente si los atributos del nuevo registro coinciden con uno de los ejemplos de entrenamiento exactamente. A más elaborado, y mucho más popular, el clasificador basado en instancia es el clasificador vecino más cercano (kNN)[2]. Teniendo en cuenta un punto para clasificarse, el clasificador kNN encuentra los k puntos más cercanos (vecinos más cercanos) de los registros de capacitación o entrenamiento. A continuación, se asigna la etiqueta de clase de acuerdo a las etiquetas de clase de sus vecinos más cercanos. La idea subyacente es que si un registro cae en un barrio en particular, donde una etiqueta de clase es predominante es porque el registro es probable que pertenecen a la misma clase.

Dado una consulta de un punto q para nosotros queremos conocer a que clase corresponde class l, y un conjunto de entrenamiento $X=x_1,l_1 \ldots x_n$, donde x_j es el *j-th* elemento y l_j es la etiqueta de la clase, el K vecino más cercano debería buscarse en el subconjunto $Y=y_1,l_1 \ldots y_n$ para cada Y pertenece a X y la distancia de q, y_k es mínima, Y contiene los k puntos en X que corresponden a la consulta del punto q, entonces la clase su etiqueta de q es $l=f(l_1 \ldots l_k)$.

La figura 2 la imagen izquierda muestra la formación señala con dos etiquetas de clase (círculos y cuadrados) y el punto de consulta (como un triángulo).La figura 2 de la derecha ilustra el barrio más cercano para $k = 1$ y $k = 7$. La consulta del punto sería clasificado como plaza para $k = 1$, y como un círculo para $k = 5$ según la regla simple mayoría de votos. Tenga en cuenta que los puntos de consulta eran justo en el límite entre los dos grupos.

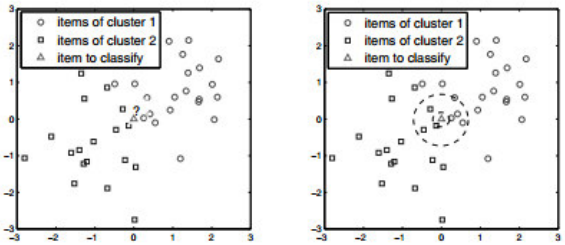


Figura 2: Ejemplo de k-vecinos más cercanos.

2.6 Arquitectura del modelo propuesto

Se propone un sistema de recomendación de objetos de aprendizaje para la mejora de las habilidades de resolución de problemas de los programadores novatos.

A continuación se enumera los detalles de la propuesta.

- A través del Programmer Machine Learning actuará como el tutor inteligente, y las soluciones a los ejercicios serán los códigos que serán analizados.
- El sistema de análisis de código nos devolverá datos procesados que entraran a un módulo que asociara los datos procesados con las rutas de aprendizaje de programación, este módulo se muestra en la figura 3 que sería el módulo de post-procesamiento.
- Los datos que son post procesados, serán utilizados por el modulo que se encargara de la recomendación
- El módulo de recomendación usara técnicas de predicción para recomendar el OA, los objetos de aprendizaje vienen a ser los ejercicios del Programmer Learning Machine.

A continuación en la figura 3 se muestra la arquitectura del modelo propuesto para el sistema de recomendación.

Para comprobar y medir el pensamiento computacional de los programadores, se les tomara el *Test de Pensamiento Computacional* [10] como parte del método de validación.

Los OA devueltos por el módulo de recomendación, están compuesto por problemas que son necesarios para que el estudiante pueda subsanar esas malas prácticas de programación, mejorando su pensamiento computacional.

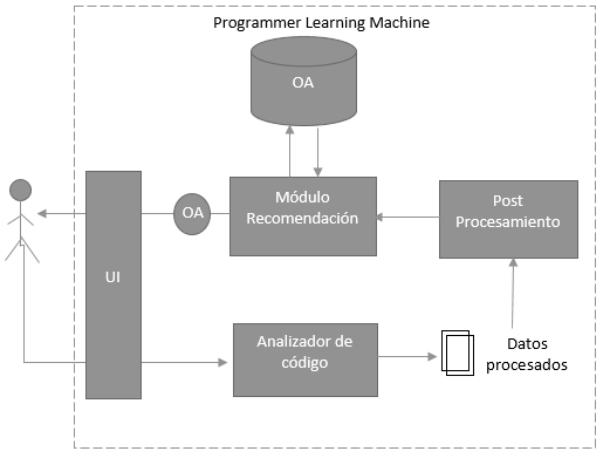


Figura 3: Modelo propuesto para el sistema de recomendación de objetos de aprendizaje

3. CONCLUSIONES

En este trabajo se ha presentado la arquitectura y la implementación de un sistema de recomendación de objetos de aprendizaje, para la mejora de las habilidades para la resolución de problemas de los programadores novatos, para ello se usa el Programmer Learning Machine como el tutor inteligente por el cual los programadores mejoran sus habilidades de programación, al existir múltiples soluciones ante un problema, estos códigos son analizados a través de un analizador de código estático, que nos devuelve las malas prácticas de programación que después de un procesado usando técnicas de predicción nos devuelve un objeto de aprendizaje recomendado.

4. REFERENCIAS

[1] Ahmad R. and Yousefi M., (2015) A Flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers: Flowchart-based intelligent tutoring system. Journal of computer assisted learning, pages 469-511.

[2] Amatriain X., Jaimes A., Oliver N., and Pujol J. M. (2011) Recommender systems handbook. Chapter Data Mining Methods for Recommender Systems, pages 39-71. Springer US.

[3] Baoxi X. (2009) Instrumentation technique and its applications in software analysis and debug.

[4] Betancur D., Moreno J., Demetrio A., and Ovalle C. (2010) Modelo para la recomendación y recuperación de objetos de aprendizaje en entornos virtuales de enseñanza/aprendizaje. Revista Avances en Sistemas e Informática, page 45-56.

[5] C. M. Bishop. (2010) Pattern recognition and machine learning. Springer. ISBN, pages 0-387-31073-8.

[6] Carrillo G. and Ochoa X. (2013) Recomendación de objetos de aprendizaje basado en el perfíl del usuario y la información de atención contextualizada.

[7] Castillo E. and Estrada L. (2013) Generación de pistas durante el aprendizaje de la programación para concursos usando el análisis estático y dinámico de las soluciones. Ingeniare. Rev. chil. ing. vol.21, page 161-216.

[8] Cazella, S., Nunes,M.. and Reategui E. (2010) Estado da arte em sistemas de recomendacao. Jornada de Atualización em Informática da SBC. Rio de Janeiro, page 161-216.

[9] Cañar K. and Heredia J. (2009) Diseño e implementación de un sistema adaptivo de recomendación de información basado en mashups. Tecnol_ogica ESPOL.

[10] Gonzalez M. (2015) Test de pensamiento computacional: principios de diseño, validación de contenido y análisis de ítems. http://www.emadridnet.org/es/test-de-pensamiento-computacional-principios-de-diseno-validacion-de-contenido-y-analisis-de-items. [Web; accedido el 20-09-2015].

[11] Hdioud F., Frikh B. and Ouhbi B. (2012) A comparison study of some algorithms in recommender systems. Information Science and Technology (CIST), Colloquium, page 130-135.

[12] Hosch W. L. (2014) Machine-learning, artifícial intelligence. http://global.britannica.com/technology/machine-learning. [Web; accedido el15-09-2015].

[13] Huajie C., Tian Z., and L. Xuandong. (2011). an instrumentation tool for program dynamic analysis in java. Fifth International Conference on Secure Software Integration and Reliability Improvement - Companion. IEEE Computer Siciety, pages 60 – 67.

[14] Kaufmann M. (2000) Data preparation for data mining.

[15] Kohavi R. and Provost F. (2000) Machine learning. page 271-274.

[16] Li-Qian Y., Lin-Zhang W., Bin L., Jian-Hua Z., and Xuan-Dong Z. (2010) Combined static and dynamic immutability analysis of java program. Chinese Journal of Computers. Vol. 33, pages 736-746.

[17] Marín P. (2014) Modelo de recomendación adaptativa de objetos de aprendizaje en el marco de una federación de repositorios, apoyado en agentes inteligentes y perfiles de usuario.

[18] Marín P., Méndezand N., Carranza D. and Morales V. (2013) Recomendación de objetos de aprendizaje basado en el perfíl del usuario y la informacion de atención contextualizada.

[19] Niemann K., Scheel M., Friedrich M., Kirschenmann U., Schmitz H., and Wolpers M. (2010) Usage-based object similarity. Journal of Universal Computer Science, 16:2272-2290.

[20] Shankar U., Talwar K., Foster J., and Wagner D. (2010) Detecting format string vulnerabilities with type qualifiers. USENIX Security Symposium.

[21]Vekariya V. and Kulkarni. (2012) Hybrid recommender systems: Survey and experiments. Second International Conference on Digital Information and Communication Technology and its Applications (DICTAP), page 469-473.

[22] Wasserman G. and Su Z. (2007) Sound and precise analysis of web applications for injection vulnerabilities. ACM SIGPLAN Conference on Programming Language Design and implementation.

[23] Wiley E. (2002) Connecting learning objects to instructional design theory: a definition, a metphor, an a taxonomy.

[24] Worsley M. and Blikstein P. (2013) Programming pathways: A technique for analyzing novice programmers' learning trajectories. Artificial Intelligence Education, pages 843-850