

# Selección de configuraciones de despliegue usando Sistemas de Recomendación en Android

Jorge L. Rodas<sup>1</sup>, José A. Galindo<sup>2</sup>, David Benavides<sup>2</sup>, Robert Soriano<sup>3</sup>

jorge.rodass@gmail.com, {jagalindo, benavides}@us.es, rsoriano@nexcol.com.ec  
email\_autor\_uno@xyz.pt, email\_autor\_dos@xpto.pt, ...

<sup>1</sup> Facultad de Ciencias de la Ingeniería, Universidad Estatal de Milagro, Cdl. Universitaria Km 1 ½ vía Km 26, Milagro, Ecuador.

<sup>2</sup> Escuela Superior de Ingeniería Informática, Universidad de Sevilla, Av. De Reina Mercedes s/n, Sevilla, España.

<sup>3</sup> Departamento de Tecnologías de Información, Nexcol S.A., Av. Francisco de Orellana – World Trade Center, Guayaquil, Ecuador.

**Pages:** 221–233

**Resumen:** Los sistemas de alta variabilidad son sistemas de software que describen una gran cantidad de configuraciones. Gestionar la variabilidad en estos sistemas es costoso y a menudo difícil por la cantidad de configuraciones que pueden derivarse. Para un desarrollador de software seleccionar la plataforma de despliegue para un producto es una actividad compleja de resolver. Para mitigar este problema investigaciones proponen el uso de técnicas automatizadas; sin embargo, hemos detectado que la información empleada en la mayoría de estas propuestas no se actualiza constantemente. Por esta razón, proponemos el uso de información extraída de informes de usuarios que es explotada mediante un sistema de recomendación que emplea algoritmos de filtrado colaborativo. En este trabajo presentamos RESDEC, una propuesta diseñada para guiar al ingeniero de software la selección y priorización de configuraciones de despliegue en sistemas de alta variabilidad. La propuesta se valida con un dataset que contiene información de dispositivos y aplicaciones móviles que se ejecutan en Android.

**Palabras-clave:** sistemas de recomendación, variabilidad, Android

## *Selection of Deployment Configurations using Recommender Systems on Android*

**Abstract:** Variability intensive systems are software systems that describe a large number of configurations. Managing variability in these systems is expensive and often difficult because of the number of configurations that can be derived. For a software developer selecting the deployment platform for a product is a complex activity to solve. To mitigate this problem, research proposes the use of automated techniques; however, we have detected that the information used in most of these proposals is not constantly updated. For this reason, we propose the use of information extracted from user reports that is exploited through a

recommendation system that uses collaborative filtering algorithms. In this paper we present RESDEC, a proposal designed to guide the software engineer in the selection and prioritization of deployment configurations in variability intensive systems. The proposal is validated with a dataset that contains information about mobile devices and applications that run on Android.

**Keywords:** Recommender systems, variability, Android.

## 1. Introducción

Los sistemas de alta variabilidad son sistemas de software cuyo comportamiento puede ser personalizado de acuerdo a las necesidades específicas de un contexto particular (Babar, Lianping Chen, & Shull, 2010). Esta personalización se materializa a través de un conjunto bien definido de puntos de variación, cada uno de los cuales captura una posibilidad de personalización que el sistema permite.

Cuando un ingeniero de software toma una decisión para todos y cada uno de los puntos de variación, se dice que define una configuración. Para representar la variabilidad existente en un sistema de alta variabilidad se emplean modelos de características. En la industria, podemos encontrar varios ejemplos de modelos que representan la variabilidad en entornos reales, como el caso de los dispositivos móviles en Android (Galindo, Turner, Benavides, & White, 2016), Linux Kernel (She, Lotufo, Berger, Wasowski, & Czarnecki, 2010) o los sistemas de gestión de precios en la nube (García-Galán, Rana, Trinidad, & Ruiz-Cortés, 2013).

En los casos citados anteriormente, hay cientos de configuraciones diferentes que complican el proceso de aseguramiento de la calidad y su gestión. Por ejemplo, para garantizar la calidad de un sistema en producción es deseable que el despliegue o ejecución (Dearle, 2007) se realice sobre configuraciones que han sido probadas previamente. Esto, en la práctica, es muy difícil realizarlo de manera manual debido al gran número de configuraciones que se derivan de un modelo. Para mitigar este problema, aparece como alternativa el análisis automatizado de modelos de características (AAFM) (Benavides, Segura, & Ruiz-Cortés, 2010). El AAFM ayuda al ingeniero de software en el manejo de configuraciones mediante algoritmos de búsqueda o programación con restricciones. Sin embargo, a pesar del uso de recursos de computación, el manejar cientos de configuraciones hace que el proceso sea lento y genere altos costos.

En el ecosistema Android podemos encontrar 224 combinaciones de configuraciones válidas (Galindo et al., 2016). La alta demanda de dispositivos Android (Forni & Vander, 2016) ha hecho que seleccionar las plataformas sobre las cuales desplegar una aplicación sea una tarea difícil para el ingeniero de software.

Para contrarrestar este problema, investigaciones sugieren que antes de desplegar un producto, primero debe ser probado sobre un conjunto limitado de configuraciones. Dichas configuraciones deben seleccionarse (Rothermel & Harrold, 1997) y priorizarse (Srikanth, Williams, & Osborne, 2005) de acuerdo a algún criterio que maximice la tasa de beneficios para la empresa y defina aquellas configuraciones que resulten “más importantes” en el cual desplegar un producto.

En este contexto, hemos encontrado en los informes de usuarios información que puede ser explotada mediante técnicas de análisis automatizado. Por un lado, podemos confiar en soluciones basadas en CSP (Galindo et al., 2016), filtrando configuraciones no óptimas. Sin embargo, los sistemas de recomendación aparecen como una alternativa ideal para recolectar información presente en los informes de usuarios para estructurarlos en una base de conocimiento dinámica que pueda ser procesada en cualquier instante del tiempo y guíe al ingeniero de software de manera personalizada en la búsqueda de configuraciones óptimas para el despliegue de sus productos.

En este artículo proponemos RESDEC, una solución basada en sistemas de recomendación para la selección de configuraciones de despliegue para un producto de software, el mismo que se encuentra estructurado de la siguiente manera: La sección 2 muestra algunos antecedentes para comprender el alcance de la propuesta. La sección 3, describe los escenarios de implementación y los algoritmos de recomendación utilizados en la propuesta. En la sección 4 presentamos el diseño, arquitectura, implementación y los datos empleados en RESDEC para su validación. Finalmente en la sección 6 se concluye el trabajo.

## **2. Preliminares**

### **2.1. Análisis automatizado de Modelos de Características**

Existen más de 30 operaciones de análisis usando modelos de características (Benavides et al., 2010). No obstante nuevas aplicaciones del mismo están siendo usadas en entorno de pruebas, configuración, entre otros.

La selección de configuraciones en sistemas de alta variabilidad representa un gran reto para el ingeniero encargado del aseguramiento de la calidad del software. Para sobrellevar el proceso de selección de configuraciones, investigaciones proponen el uso de pruebas de interacción combinatoria (Nie & Leung, 2011) que explora las características comunes de un producto para reducir el costo incurrido al realizar las pruebas. Una de estas propuestas proponen el uso de técnicas t-wise para obtener un conjunto de configuraciones que necesitan ser probadas dentro de un subconjunto de todas las configuraciones (Perrouin et al., 2012).

Se ha comprobado que las pruebas combinatorias producen buenos resultados en estudios empíricos, principalmente porque los puntos de interacción entre los productos de software han sido probados como fuentes clave de errores (Cohen, Dwyer, & Jiangfan Shi, 2008). Sin embargo, aun hay espacio para la mejora para reducir el costo de las pruebas mientras se maximiza su valor.

### **2.2. Sistemas de Recomendación**

Un sistema de recomendación se define como un sistema inteligente que proporciona a los usuarios una serie de sugerencias de forma personalizada de acuerdo a los gustos o preferencias sobre algún producto o servicio. Los sistemas de recomendación nacieron en la década de 1990 (Goldberg, Nichols, Oki, & Terry, 1992) y desde de esa fecha han crecido enormemente en términos de desarrollo algorítmico, así como en aplicaciones implementadas.

En el mercado existen experiencias practicas exitosas de la implementación de los métodos de recomendación en contextos de comercio electrónico, como los casos de Amazon (Linden, Smith, & York, 2003) y Netflix (Tuzhilin, Koren, Bennett, Elkan, & Lemire, 2008). En un contexto diferente, los sistemas de recomendación son utilizados además para la clasificación de imágenes médicas en el diagnóstico de enfermedades. Todo este panorama, pone de relieve la importancia de los sistemas de recomendación en el estado actual de las cosas abriendo espacio al desarrollo de herramientas en nuevos campos de aplicación. En (Jesus Bobadilla, Hernando, Ortega, & Bernal, 2011) an increasing need arises to develop some kind of evaluation framework for collaborative filtering measures and methods which is capable of not only testing the prediction and recommendation results, but also of other purposes which until now were considered secondary, such as novelty in the recommendations and the users\’ trust in these. This paper provides: (a y (J. Bobadilla, Ortega, Hernando, & Gutiérrez, 2013) se describen en detalle varias aplicaciones que emplean sistemas de recomendación.

La literatura (Adomavicius & Tuzhilin, 2005) divide a los sistemas de recomendación en dos grupos: sistemas basados en filtrado colaborativo y sistemas basados en filtrado por contenido. Los sistemas basados en filtrado colaborativo (Konstan et al., 1997), (Linden et al., 2003) se basan en el concepto de análisis de perfiles de usuarios, en donde las recomendaciones se generan de acuerdo a los gustos de usuarios con preferencias similares. Por otro lado, los sistemas basados en filtrado por contenido (Pazzani & Billsus, 1997) son aquellos en el cual la recomendación de un ítem o producto se genera a partir de la descripción de sus características y el perfil de intereses de los usuarios. El uso de este tipo de sistemas está enmarcado en una variedad de dominios que van desde la recomendación de sitios web, viajes, restaurantes, series de TV, artículos de venta, entre otros.

A pesar de los grandes avances tecnológicos de los sistemas de recomendación, pocas son las propuestas encaminadas a la mejora continua en el campo de la ingeniería de software, abriendo un mundo de posibilidades para introducir el concepto de sistemas de recomendación en esta área.

### 3. Descripción de la Propuesta

En esta sección presentamos RESDEC (REcommender system for Selecting Deployment Configurations), una propuesta basada en sistemas de recomendación



Figura 1 – Componentes de RESDEC

que fue diseñada para ayudar al ingeniero de software en la selección y priorización del entorno de despliegue para sus productos de software antes de ser enviados a producción. RESDEC fue construido sobre la base del backend de TESALIA (Galindo et al., 2016) y es actualmente compatible con el ecosistema FaMa (Benavides, Segura, Trinidad, & Ruiz-Cortés, 2007). La solución consta de dos componentes: 1) Base de Conocimiento y 2) Generador de Recomendaciones. La figura 1 ilustra los componentes que integra RESDEC.

### 3.1. Base de Conocimiento

La base de conocimiento representa el eje principal de nuestra propuesta y se construye a partir de los valores que recogen los informes de usuarios.

En nuestra solución hemos usado los componentes de un sistema de recomendación común: 1) usuarios, 2) ítems y 3) puntuaciones, de la siguiente manera: los usuarios están representados por los productos  $p$  que serán desplegados, los ítems por las configuraciones de despliegues  $d$  que se recomendarán y las valoraciones  $v$  por los valores que aparecen en un informe de usuario. La base de conocimiento se construye a partir de los informes de usuarios y da origen a la matriz  $M1 \in M(m, n, \mathbb{R})$ , de órdenes diferentes que contienen la información necesaria para llevar a cabo el proceso de recomendación.

La matriz  $M1$  relaciona  $m$  configuraciones de despliegue ( $d_1, d_2, d_3, \dots, d_m$ ) con  $n$  productos ( $p_1, p_2, p_3, \dots, p_n$ ), donde las celdas contienen números reales determinados por una función, en este caso por el valor promedio  $\bar{x}$  de las valoraciones emitidas por un conjunto de usuarios que han usado los productos  $p$  sobre las configuraciones  $d$ . Estos valores podrían representar criterios de los usuarios en relación a gustos, funcionalidad, errores o desperfectos que un producto pudiera tener al ser desplegado en un ecosistema en particular. En nuestro caso, estos valores reflejan los niveles de funcionalidad que podría tener un producto  $p_i$  en una configuración de despliegue  $d_j$  a partir de un valor entre 1 y 5.

La figura 2 muestra la representación gráfica de la matriz  $M1$ .

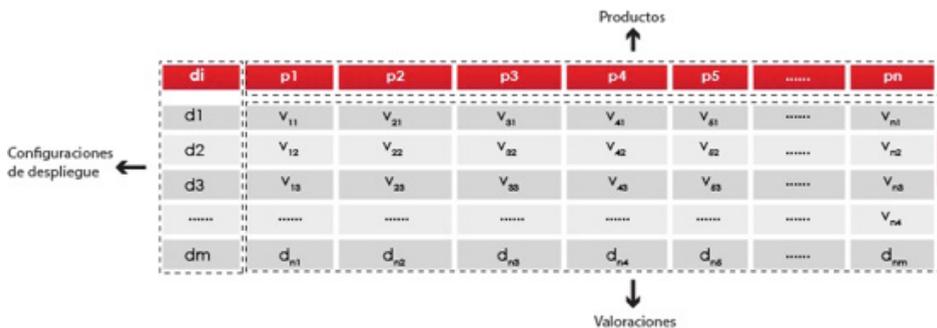


Figura 2 – Elementos de la matriz  $M1$

### 3.2. Generador de Recomendaciones

Dependiendo de las necesidades específicas del ingeniero de software, hemos identificado dos escenarios donde se puede utilizar información dinámica generada a partir de informes de usuario para definir la estrategia de despliegue de un producto.

**Arranque en cero.-** El primer escenario se presenta cuando hay un usuario nuevo y desea conocer que configuraciones de despliegue serían las más opcionadas para poner en ejecución un producto de software. Esto ocurre por ejemplo, cuando el ingeniero de software diseña una nueva aplicación y desconoce en qué dispositivos móviles sería más óptimo instalarlo, para ello, el sistema hace la recomendación considerando los dispositivos móviles más populares en donde otras aplicaciones han sido desplegadas con éxito.

Para realizar recomendaciones en este escenario hemos empleado un algoritmo basado en popularidad (1). Cuando se usa un algoritmo basado en popularidad, el sistema toma la matriz  $M1$  y prioriza configuraciones basado en la mayor cantidad de interacciones que una configuración  $d$  ha tenido con los productos  $p$ . Una vez priorizada la información, recomienda las configuraciones  $d$  por orden de popularidad, dando exactamente el mismo ranking a cada uno de los productos  $p$ .

$$f(p, d) = \{r(v, d) \neq \emptyset \mid v \in U\} \quad (1)$$

**Transición de configuraciones basadas en valoraciones.-** El segundo escenario se presenta cuando el usuario ha usado previamente una configuración de despliegue para un producto y basado en esa experiencia quiere explorar otras configuraciones para obtener nuevas alternativas de recomendación. Esto ocurre, por ejemplo, cuando el ingeniero de software ha implementado una aplicación en un dispositivo móvil y desea conocer en qué otros dispositivos podría también instalarse, para ello, el sistema considera las aplicaciones más similares a la aplicación instalada y las valoraciones que han recibido dichos dispositivos.

Como alternativa de solución hemos integrado en RESDEC el algoritmo *Item – basedCollaborativeFiltering* (Melville & Sindhvani, 2011), en el cual en lugar de encontrar productos  $p$  similares, el algoritmo se centra en la similaridad entre configuraciones  $d$ . En nuestro trabajo el algoritmo toma la matriz  $M1$  y establece la similaridad entre las configuraciones  $d$  que ha usado un producto  $p$  y los compara con configuraciones similares. Para encontrar la similaridad entre dos configuraciones  $i$  y  $j$  utilizando el algoritmo *Item – basedCollaborativeFiltering* usamos el coeficiente de correlación de Pearson tal y como se describe en (2):

$$s_{i,j} = \frac{\sum_{u \in U} (v_{u,i} - \bar{v}_i)(v_{u,j} - \bar{v}_j)}{\sqrt{\sum_{u \in U} (v_{u,i} - \bar{v}_i)^2} \sqrt{\sum_{u \in U} (v_{u,j} - \bar{v}_j)^2}} \quad (2)$$

Donde  $U$  es el conjunto de todos los productos que han valorado tanto las configuraciones  $i$  como  $j$ ,  $v_{u,i}$  es la valoración del producto  $u$  en la configuración  $i$ , y  $\bar{v}_i$  es la valoración promedio de las configuraciones  $i$  entre los productos.

Establecida la similaridad entre las configuraciones, en (3) usando un promedio de ponderación simple, el algoritmo predice la valoración para una configuración  $i$  por parte de un producto  $a$ .

$$f_{a,p} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} w_{i,j}} \quad (3)$$

Donde  $f_{a,p}$  es la predicción para la configuración  $d$  por parte del producto activo  $a$ ,  $w_{i,j}$  es la similitud entre las configuraciones  $i$  y  $j$ , y  $K$  representa el conjunto de las  $k$  configuraciones valorados por el producto  $a$  que son más similares a  $i$ .

## 4. Aplicación Web y Resultados Preliminares

En esta sección, presentamos el experimento llevado a cabo para evaluar nuestra propuesta. La sección se divide en dos partes, la primera presenta una visión general sobre el diseño y arquitectura de la herramienta web RESDEC y la segunda describe los mecanismos empleados para evaluar la solución.

### 4.1. Diseño de RESDEC

RESDEC <sup>1</sup> (<https://github.com/RESDEC>) es una aplicación web diseñada en Java que usa como motor de base de datos MySql y está organizada en tres módulos para permitir la recomendación de las plataformas de despliegue.

El primer módulo *Usuario* que permite el registro de los usuarios del sistema que pueden obtener las recomendaciones de los entornos de despliegue.

El segundo módulo *Entorno de despliegue* que alimenta el sistema con la información necesaria para establecer nuevas recomendaciones, el cual varía en función del entorno en el que se utilice el sistema; y finalmente, el módulo *Recomendaciones*, que ejecuta el algoritmo de recomendación apropiado.

### 4.2. Arquitectura

La arquitectura RESDEC se compone de dos partes, como se muestra en la Figura 3: El servidor de aplicaciones y la interfaz de usuario.

El *servidor de aplicaciones* está conformado por el servidor GlassFish 4.1 el cual contiene un conjunto de vistas, creadas en Java Server Faces 2.1<sup>2</sup>, el cual usa como framework PrimeFaces<sup>3</sup>, empleadas para el diseño de la interfaz gráfica; el controlador Faces Servlets encargado de administrar el ciclo de vida de los JSF; una capa de instrumentación en Managed Beans<sup>4</sup> el cual contiene todos los atributos, métodos y propiedades de las entidades del sistema y con los cuales interactúan los JSF; y finalmente el servidor de datos MySql el cual almacena la información que procesa el sistema en DBKnowledge. La

<sup>1</sup> RESDEC Website: <http://www.resdec.com>

<sup>2</sup> JSF website: <http://www.oracle.com/technetwork/java/javaee/overview-140548.html>

<sup>3</sup> PrimeFaces website: <https://www.primefaces.org/>

<sup>4</sup> MBeans website: <http://javabeat.net/jsf-2-managed-beans/>

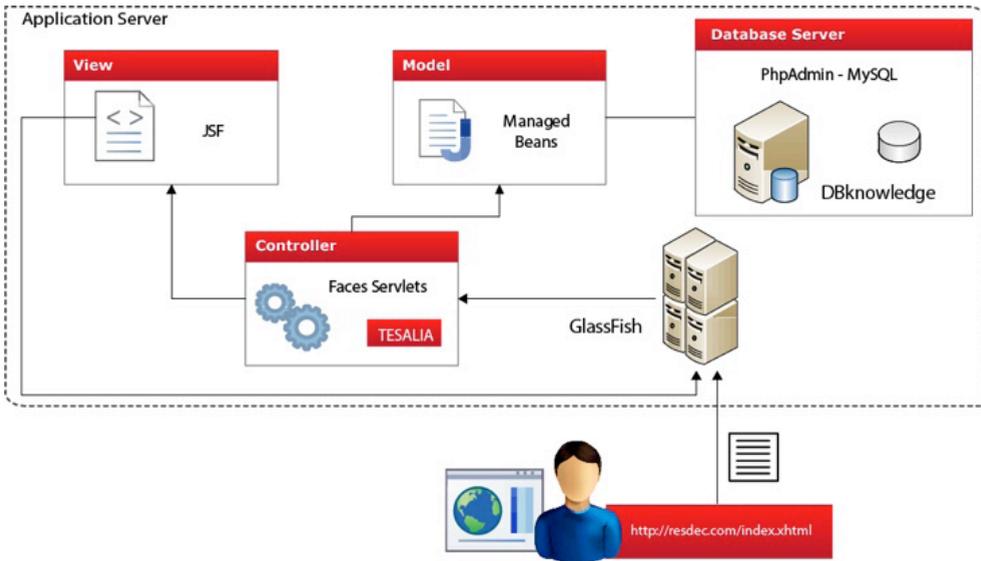


Figura 3 – Arquitectura de RESDEC

*interfaz de usuario* es la capa de presentación que permite la comunicación bidireccional entre el usuario y el sistema a través de un navegador web.

### 4.3. Dataset y Validación

**Dataset de dispositivos móviles de Android:** Para recolectar la información almacenada en el dataset de Android, se realizó una encuesta a 2,099 usuarios de la comunidad estudiantil y docente de la Universidad Estatal de Milagro, Ecuador<sup>5</sup>. La población comprende usuarios con edades entre 17 y 56 años de edad, de los cuales el 72% fueron de género femenino y el 28% del género masculino; de estos dos grupos, el 16% corresponden al área de ingeniería y tecnología, el 23% a administración, el 28% salud, el 12% educación, el 4% turismo, el 2% idiomas y el 15% a áreas de diseño gráfico, psicología y comunicación social.

Esta encuesta nos permitió conocer que aplicaciones son las más utilizadas por los usuarios, los dispositivos que poseen, el comportamiento de las aplicaciones en sus dispositivos y las valoraciones que se les asignan. El cuestionario se puede encontrar en el sitio web de RESDEC. Después de recopilar los datos para este dataset, estructuramos la información para la matriz  $M1$ , la cual establece la relación entre las aplicaciones, dispositivos y valoraciones.

**Resultados.-** Finalmente, la información fue registrada en la base de datos DBKnowledge. Las pruebas finales involucraron la generación automática de recomendaciones de entre un conjunto de datos de 832 dispositivos móviles que fueron puestas a prueba en el departamento de Tecnologías de la Información y de la Comunicación de la Universidad Estatal de Milagro para seleccionar los entornos de

<sup>5</sup> UNEMI website: <http://www.unemi.edu.ec>

despliegue en los cuales se podría implementar la versión móvil del Sistema de Gestión Académico UNEMI (SGA UNEMI)<sup>6</sup>.

## 5. Trabajo Relacionado

A continuación revisamos algunos trabajos que se relacionan directamente con nuestra propuesta.

**Sistemas de recomendación de características.**- (Mazo, Dumitrescu, Salinesi, & Diaz, 2014) presentan una propuesta en el cual mediante el uso de una colección de heurísticas y utilizando programación con restricciones, priorizan configuraciones para acelerar el proceso de configuración de un producto y evitar errores a los cuales están expuestos.

No obstante, la propuesta se basa en el uso de heurísticas que no tienen en cuenta las preferencias de los usuarios, lo cual podría llevar al usuario a configurar productos que no satisficaría sus requerimientos.

**Sistemas de recomendación de productos.**- (Martinez et al., 2015) proponen el uso de un algoritmo genético interactivo para predecir configuraciones idóneas. La propuesta pone a los usuarios a evaluar un conjunto de configuraciones de un producto utilizando una escala del 1 al 5 (1=malo, 5=bueno) a partir del cual se construye un dataset que es explotado usando técnicas de minería de datos.

Si bien es cierto que la propuesta contiene todos los elementos que necesita un sistema de recomendación (usuarios, productos y valor), los algoritmos genéticos empleados no se benefician de la información de las características de las configuraciones para hacer las recomendaciones, de hecho, el usuario no valora las características sino configuraciones.

**Sistemas de recomendación personalizados.**- (Pereira et al., 2016) nos presentan una propuesta en la cual se usa como elemento importante del sistema de recomendación al usuario, siendo la primera propuesta en la que no se excluye en su totalidad a los usuarios.

Sin embargo, los datos que procesa el sistema no provienen de un contexto donde el usuario sea quien lo retroalimente; es decir, las configuraciones que se usan para establecer la similitud no tienen una variable que determine si esa configuración ha sido evaluada de forma positiva o negativa por el usuario que hizo uso de la misma en el pasado. Siendo las opiniones o valoraciones de los usuarios una de las bases principales para establecer las recomendaciones.

## 6. Conclusiones y Trabajo Futuro

En este artículo, se ha presentado un enfoque diferente para la selección de configuraciones en sistemas de alta variabilidad. El uso de sistemas de recomendación integrado con herramientas de análisis automatizado, es uno de los aportes nuevos de esta investigación en la industria de la ingeniería de software. A futuro nuestro trabajo pretende abordar los siguientes temas:

<sup>6</sup> UNEMI SGA website: <https://sga.unemi.edu.ec>

- a. Explorar algoritmos basados en filtrado por contenido para establecer recomendaciones basadas no sólo en valoraciones sino en las características de las configuraciones de un producto.
- b. Extender las utilidades de la herramienta RESDEC incorporando diversos algoritmos de filtrado colaborativo y filtrado por contenido.
- c. Evaluar los resultados que procesa el sistema mediante el uso de técnicas y procedimientos que se usan en la evaluación de sistemas de recomendación.

## Referencias

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. <https://doi.org/10.1109/TKDE.2005.99>
- Babar, M. A., Lianping Chen, & Shull, F. (2010). Managing Variability in Software Product Lines. *IEEE Software*, 27(3), 89–91, 94. <https://doi.org/10.1109/MS.2010.77>
- Benavides, D., Segura, S., & Ruiz-Cortés, A. (2010). Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6), 615–636. <https://doi.org/10.1016/J.IS.2010.01.001>
- Benavides, D., Segura, S., Trinidad, P., & Ruiz-Cortés, A. (2007). Fama: Tooling a Framework for the Automated Analysis of Feature Models. *VaMoS*. Retrieved from <https://idus.us.es/xmlui/handle/11441/26357>
- Bobadilla, J., Hernando, A., Ortega, F., & Bernal, J. (2011). A framework for collaborative filtering recommender systems. *Expert Systems with Applications*, 38(12), 14609–14623. <https://doi.org/10.1016/J.ESWA.2011.05.021>
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132. <https://doi.org/10.1016/J.KNOSYS.2013.03.012>
- Cohen, M. B., Dwyer, M. B., & Jiangfan Shi. (2008). Constructing Interaction Test Suites for Highly-Configurable Systems in the Presence of Constraints: A Greedy Approach. *IEEE Transactions on Software Engineering*, 34(5), 633–650. <https://doi.org/10.1109/TSE.2008.50>
- Dearle, A. (2007). Software Deployment, Past, Present and Future. In *Future of Software Engineering (FOSE '07)* (pp. 269–284). IEEE. <https://doi.org/10.1109/FOSE.2007.20>
- Forni, A., & Vander, R. (2016). Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016. Retrieved March 13, 2018, from <https://www.gartner.com/newsroom/id/3415117>
- Galindo, J. A., Turner, H., Benavides, D., & White, J. (2016). Testing variability-intensive systems using automated analysis: an application to Android. *Software Quality Journal*, 24(2), 365–405. <https://doi.org/10.1007/s11219-014-9258-y>

- García-Galán, J., Rana, O., Trinidad, P., & Ruiz-Cortés, A. (2013). Migrating to the Cloud - A Software Product Line based Analysis. In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (pp. 416–426). SciTePress - Science and Technology Publications. <https://doi.org/10.5220/0004357104160426>
- Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61–70. <https://doi.org/10.1145/138859.138867>
- Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3), 77–87. <https://doi.org/10.1145/245108.245126>
- Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*. Retrieved from <http://ieeexplore.ieee.org/abstract/document/1167344/>
- Martinez, J., Rossi, G., Ziadi, T., Bissyandé, T. F. D. A., Klein, J., & Le Traon, Y. (2015). Estimating and Predicting Average Likability on Computer-Generated Artwork Variants. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference - GECCO Companion '15* (pp. 1431–1432). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2739482.2764681>
- Mazo, R., Dumitrescu, C., Salinesi, C., & Diaz, D. (2014). Recommendation Heuristics for Improving Product Line Configuration Processes, 511–537. [https://doi.org/10.1007/978-3-642-45135-5\\_19](https://doi.org/10.1007/978-3-642-45135-5_19)
- Melville, P., & Sindhvani, V. (2011). Recommender Systems. In *Encyclopedia of Machine Learning* (pp. 829–838). Boston, MA: Springer US. [https://doi.org/10.1007/978-0-387-30164-8\\_705](https://doi.org/10.1007/978-0-387-30164-8_705)
- Nie, C., & Leung, H. (2011). A survey of combinatorial testing. *ACM Computing Surveys*, 43(2), 1–29. <https://doi.org/10.1145/1883612.1883618>
- Pazzani, M., & Billsus, D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27(3), 313–331. <https://doi.org/10.1023/A:1007369909943>
- Pereira, J. A., Matuszyk, P., Krieter, S., Spiliopoulou, M., Saake, G., Pereira, J. A., ... Saake, G. (2016). A feature-based personalized recommender system for product-line configuration. *ACM SIGPLAN Notices*, 52(3), 120–131. <https://doi.org/10.1145/3093335.2993249>
- Perrouin, G., Oster, S., Sen, S., Klein, J., Baudry, B., & le Traon, Y. (2012). Pairwise testing for software product lines: comparison of two approaches. *Software Quality Journal*, 20(3–4), 605–643. <https://doi.org/10.1007/s11219-011-9160-9>
- Rothermel, G., & Harrold, M. J. (1997). A safe, efficient regression test selection technique. *ACM Transactions on Software Engineering and Methodology*, 6(2), 173–210. <https://doi.org/10.1145/248233.248262>

- She, S., Lotufo, R., Berger, T., Wasowski, A., & Czarnecki, K. (2010). The Variability Model of The Linux Kernel. *VaMoS*. Retrieved from [https://www.researchgate.net/profile/Andrzej\\_Wasowski/publication/220789837\\_Evolution\\_of\\_the\\_Linux\\_Kernel\\_Variability\\_Model/links/54ccb7f10cf298d6565ac185.pdf](https://www.researchgate.net/profile/Andrzej_Wasowski/publication/220789837_Evolution_of_the_Linux_Kernel_Variability_Model/links/54ccb7f10cf298d6565ac185.pdf)
- Srikanth, H., Williams, L., & Osborne, J. (2005). System test case prioritization of new and regression test cases. In *International Symposium on Empirical Software Engineering, 2005*. (pp. 62–71). IEEE. <https://doi.org/10.1109/ISESE.2005.1541815>
- Tuzhilin, A., Koren, Y., Bennett, J., Elkan, C., & Lemire, D. (2008). Large-scale recommender systems and the netflix prize competition. In *KDD Proceedings*. Retrieved from [http://www.exercicescorriges.com/i\\_337810.pdf](http://www.exercicescorriges.com/i_337810.pdf)

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.