

# Modelo de sistema de recomendación basado en el contexto a partir del análisis de código estático para el desarrollo del Pensamiento Computacional: Caso de Programación Web

## Model of a Recommendation System Based on the Context from the Analysis of Static Code for the Development of Computational Thinking: A Web Programming Case

Julio Vera, Klinge Villalba-Condori, Sonia Castro Cuba-Sayco

{jveras, kvillalbac}@unsa.edu.pe, (<https://orcid.org/0000-0001-5526-5223>), (<https://orcid.org/0000-0002-8621-7942>)  
scastroc@ucsm.edu.pe (<https://orcid.org/0000-0001-8715-7868>)

Universidad Nacional de San Agustín, Facultad de Ciencias de la Educación, c/ Santa Catalina, 117. Arequipa, Perú

Universidad Católica de Santa María, Facultad de Ciencias y Tecnologías Sociales y Humanidades, Urb. San José, s/n. Umacollo. Arequipa, Perú

### Resumen

Este trabajo propone un modelo de recomendación de contenidos educativos basado en el contexto de un usuario, el cual usa un modelo de contexto que incorpora el rol, las tareas, ejercicios de programación y su aplicación al problema de recomendación. Las recomendaciones se hacen sobre la base de la estimación de la diferencia que existe entre el nivel de conocimiento de un usuario frente a las habilidades que este requiere en el contexto en el que se encuentra. Los experimentos desarrollados en el contexto del estudiante muestran que, cuando se utiliza un modelo de razonamiento probabilístico, este modelo ayuda a obtener mejores recomendaciones de contenidos educativos, acorde con las competencias que le hacen falta a un estudiante respecto a un tema que necesita aprender. En consecuencia, se busca una estandarización para sistemas de recomendación.

### Palabra clave

Sistema de recomendación; Pensamiento computacional; Programación web

### Abstract

This project proposes a recommendation model for educational content based on the context of a user, which uses a context model that incorporates the role, tasks, programming exercises and their application to the problem of recommendation. The recommendations are made on the basis of the estimate of the difference between the current level of knowledge of a user in front of the skills required in their work context. The experiments developed in the context of the student, show that, using a model of probabilistic reasoning helps to get better recommendations of educational content, according to the missing competences of a student on an issue that needs to learn, which seeks to standardization for recommendation systems.

### Keywords

Recommendation System; Computational Thinking; Web Programming

---

# 1. Introducción

En el dominio de la educación existe gran cantidad y diversidad de material educativo que puede contribuir al proceso enseñanza y aprendizaje, con independencia de la variedad y profundidad de conocimientos previos (Carrillo & Ochoa, 2013). Enriquecer la práctica pedagógica —organización de aprendizajes, conocimientos, clima del aula, evaluación y estrategias y medios (Villalba-Condori, 2018)— es una tarea permanente que necesariamente involucra actividades colaborativas multidisciplinares orientadas a mejorar los aprendizajes (Lavonen, Kohonen & Villalba-Condori, 2018). Lo que evidencia que la gestión del conocimiento se hace imprescindible en estos tiempos (Fidalgo-Blanco, Sein-Echaluze & García-Peñalvo, 2014).

En la actualidad estamos viviendo un tiempo entre quienes usan las computadoras y quienes programan las computadoras (Worsley & Blikstein, 2013) y, en consecuencia, varias disciplinas se están dando cuenta de que son muy importantes las habilidades de pensamiento computacional, porque nos da la facilidad de pensar de manera constructiva sobre cualquier número de tareas (Worsley & Blikstein, 2013). El Pensamiento Computacional está concebido como una competencia del siglo XXI que permite solucionar problemas desde la perspectiva computacional (García-Peñalvo, Reimann, Tuul, Rees & Jormanainen, 2016).

Un objeto de aprendizaje (OA) es “cualquier recurso digital que puede ser utilizado repetidamente para facilitar el aprendizaje” (Wiley, 2002). Los OA 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. A estos OA se puede acceder a través de distintos repositorios. Existe gran cantidad de material educativo útil para unos usuarios y no para otros. Es por ello necesario que los OA consideren aspectos pedagógicos relevantes que permitan mejorar la calidad del servicio educativo y sobre todo atender las necesidades cognitivas de los estudiantes (Villalba-Condori, Castro Cuba, Deco, Bender & García-Peñalvo, 2017).

El aprendizaje en línea es un camino revolucionario para mejorar y adecuar los procesos de formación en la vida moderna beneficiando a más personas. Esto exige a las organizaciones educativas gestionar el conocimiento de manera eficiente y eficaz (Fidalgo-Blanco, Sein-Echaluze & García-Peñalvo, 2015).

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 un grupo grande de información educativa. Estos sistemas de recomendación pueden estar basados en filtrado colaborativo, contenido o híbrido (Niemann, Scheffel, Friedrich, Kirschenmann, Schmitz & Wolpers, 2010), atendiendo las necesidades de los usuarios. Es el caso del uso de dispositivos móviles (Vera, Mamani & Villalba-Condori, 2015.)

El objetivo de este trabajo de investigación es crear un modelo de sistema de recomendación para

---

la programación Web. Las recomendaciones tienen en cuenta la brecha que existe entre el nivel de conocimiento actual de un usuario frente a las habilidades que requiere en su contexto. También se considera a los que recién se involucran en la programación (Batallanos, Poccohuanca & Sayco, 2017).

## 2. Trabajos relacionados

Sobre Pensamiento Computacional existen diferentes proyectos, como el Proyecto TACCLE 3 Consortium (2017), donde se desarrolla un conjunto de actividades orientadas a desarrollar el Pensamiento Computacional (García-Peñalvo, 2016). O investigaciones en las que se ha encontrado efectos que vinculan una mayor predisposición hacia las habilidades del Pensamiento Computacional con lo recibido desde el jardín de infancia hasta la educación secundaria (García-Peñalvo & Mendes, 2018). Dentro del Pensamiento Computacional puede considerarse inclusive los movimientos que consideran aplicaciones robóticas (D'Abreu & Villalba-Condori, 2017).

Para un motor de recomendación para contenidos educativos en el contexto organizacional de un usuario, la novedad en este documento radica en la creación de un modelo de contexto para que el usuario incorpore el rol y las tareas asignadas, y su aplicación al problema (Saraswat, Ghosh, Agrawal & Narayanan, 2015). Las recomendaciones se hicieron sobre la base de la brecha estimada que existe entre el nivel de conocimiento actual de un empleado y el conjunto de habilidades requeridas en su contexto laboral. Se utiliza un marco de razonamiento probabilístico para las recomendaciones, para tener en cuenta las especificaciones inexactas de las competencias y los requisitos del usuario en el contexto laboral.

En Chen, Tong, Liu, Han, & Xia (2017) se propone la necesidad de métodos inteligentes para organizar la gran cantidad de recursos de enseñanza y aprendizaje. Un aspecto importante es planificar la ruta de aprendizaje. De acuerdo con esta investigación, el algoritmo de colonia de hormigas mostró grandes ventajas en la planificación del camino de aprendizaje. A diferencia del algoritmo tradicional de colonia de hormigas, se adoptó la distancia Mahalanobis para calcular la distancia entre los datos en el ACO mejorado. Además, con el fin de transformar el mapa conceptual e información incommensurable en el programa en datos mensurables, se construyó un nuevo sistema de coordenadas para dibujar puntos que representan las unidades de enseñanza o aprendizaje en él. Los resultados experimentales mostraron que este método puede recomendar una ruta de aprendizaje eficiente.

También proponía un método híbrido de recomendación adaptativa de objetos de aprendizaje, el cual debe inicialmente realizar la selección de características del perfil del estudiante que mejor lo definan. Los sistemas de recomendación híbrida de materiales educativos digitales apoyan procesos de aprendizaje virtual y ayudan a los estudiantes a encontrar recursos relevantes que se adapten a

---

sus necesidades y preferencias (Rodríguez, Duque & Ovalle, 2016). Se seleccionan los algoritmos de recomendación que se van a aplicar, seguido de la elección en la combinación de los resultados de esas técnicas. Finalmente, se evalúa la integración de técnicas teniendo en cuenta la relevancia que se obtiene de las recomendaciones. Se puede concluir que al utilizar sistemas adaptativos de recomendación híbrida se mejoran los resultados, especialmente cuando la combinación incluye la técnica de filtrado por conocimiento.

## 3. Marco teórico

### 3.1. Sistema de recomendación

Los sistemas de recomendación son técnicas y herramientas de software que proporcionan como sugerencias un subconjunto de elementos pertenecientes al universo de alternativas que se consideran los más apropiados para el usuario (Mahmood & Ricci, 2009). 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 (Cazella, Nunes & Reategui 2010).

En los sistemas de recomendación basados en contexto, el contexto es cualquier información que puede ser utilizada para caracterizar la situación de las entidades (persona, lugar u objeto) que se consideran relevantes para la interacción entre un usuario y una aplicación, incluyendo el usuario y la aplicación en sí mismos. El contexto es normalmente la ubicación, identidad y el estado de personas, grupos y objetos físicos u objetos digitales.

Así, los sistemas de recomendación basados en el contexto se ocupan de modelar y predecir los gustos del usuario y preferencias mediante la incorporación de información contextual. Esta información contextual puede ser de diferentes tipos: como el tiempo, la ubicación, el compañero, el propósito de una compra, etc. Además, cada tipo contextual puede tener una estructura complicada que refleja la complejidad de la información contextual.

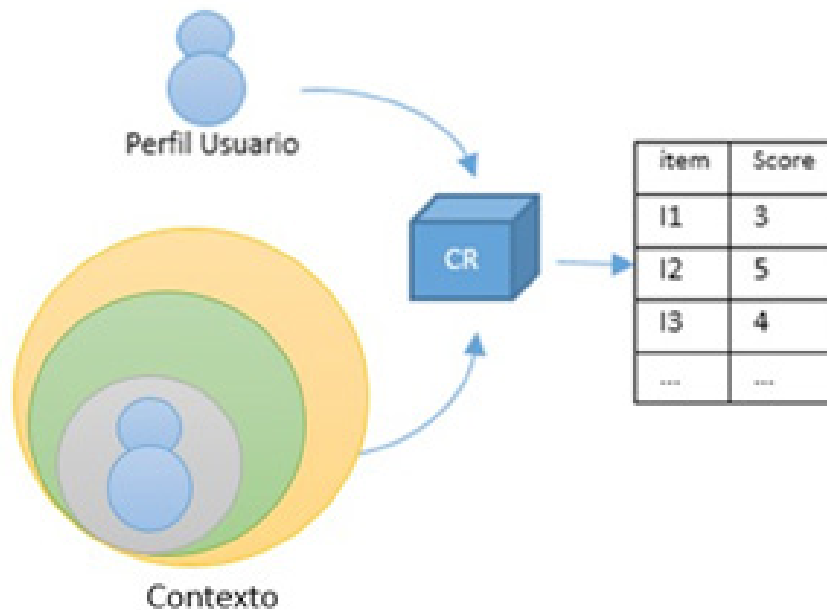


Figura 1. Recomendaciones basadas en contexto

### 3.2. 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 máquina o cualquier abstracción de alguna de las opciones anteriores (Altuna & Estrada, 2013).

Su uso 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 (Wasserman & Su, 2007); usando diferentes niveles en las posibilidades de configuración, tipos de errores detectados y profundidad, con excelentes resultados (Shankar, Talwar, Foster & Wagner, 2001).

### 3.3. Abstract Syntax Tree

Con el fin de abordar las diferencias sintácticas, los AST se adoptan como “una representación formal de una estructura sintáctica del software” (Newcomb, 2005). En un nivel superficial, las construcciones de muchos lenguajes de procedimiento parecen similares, y la eliminación de la sintaxis de estos haría que todos los AST sean análogos. Los AST resultantes tras el análisis de código fuente se basan en la gramática libre de contexto, utilizada para definir la sintaxis del lenguaje (Tripp, 2006). Este análisis puede llegar a ser aún más significativo al dirigirse a las características adicionales del lenguaje, tales como tipos de datos.

El uso principal de AST es facilitar la implementación de herramientas de compilación. Para este propósito, los AST se construyen a partir de una serie de tokens después de análisis de léxico del código Fuente (Owens & Anderson, 2013). Uno de esos usos que un AST puede soportar es de detección de código duplicado, por el que un AST diseñado para apoyar la eficiencia de comparación de datos (Owens & Anderson, 2013) requiere solo un patrón que se encuentran. Esta técnica es similar a la de detección de plagio, que hace uso de la comparación de código, descrita por Baojiang, Jiansong, Tao, Jianxin & Ding (2010).

Para nuestro trabajo, calculamos la similitud de los AST generados por el trabajo del docente, generaremos su AST, y luego lo compararemos con los AST de los estudiantes.

### 3.4. ANTLR

Para el Análisis de Código Estático se utiliza la herramienta ANTLR, un generador de análisis de gran alcance para la lectura, gestión, ejecución, o la traducción de texto estructurado o archivos binarios. A partir de una gramática, ANTLR genera un analizador que puede construir y caminar árboles de análisis sintáctico (Parr, 2014).

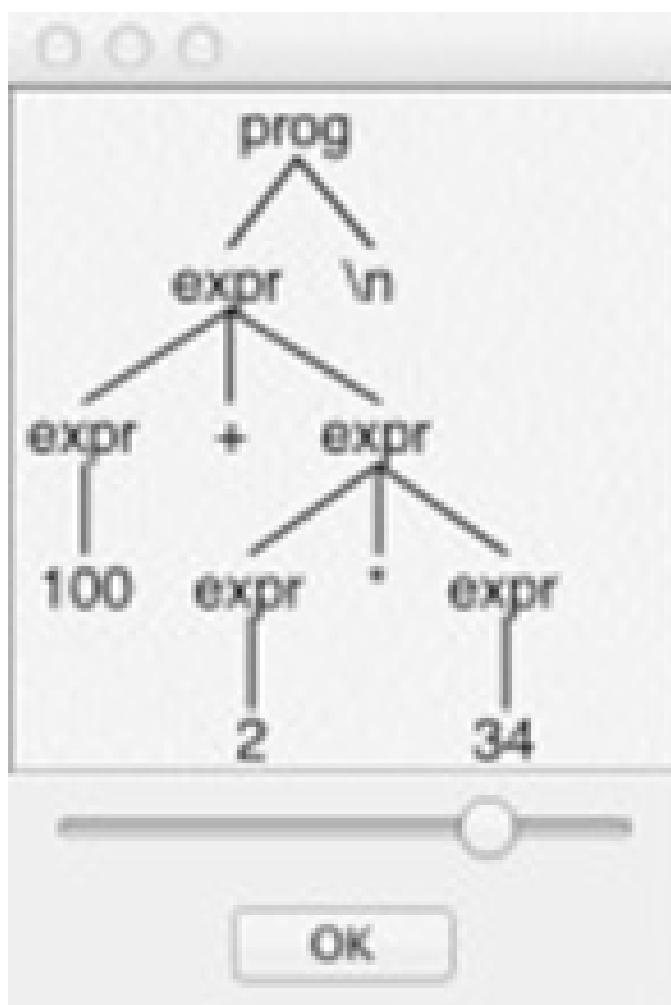


Figura 2. Ejemplo de la gramática de cálculo de operaciones convertido en un árbol

---

### 3.5. Medidas de similitud

Para evaluar y corregir una solución enviada por un estudiante se plantea utilizar las siguientes medidas de similitud frente a la solución planteada por el profesor:

- *Similitud Levenshtein*. Este algoritmo permite calcular la similaridad entre dos cadenas de texto A y B; se basa en el conjunto mínimo de operaciones de edición necesarias para transformar A en B, o viceversa. Este algoritmo tiene las operaciones: destrucción, inserción y sustitución. Cuanto más cerca de cero sea la distancia de Levenshtein, más parecidas son las hileras.

Esta medida de similitud se usó al convertir el AST en cadena de texto. Cuando se hace un recorrido del AST y se concatenan los tokens en una cadena de texto, al contener la parte abstracta del código, se aplica esta medida de similitud respecto al AST de la solución del docente. Para ello, se toma como base la rúbrica propuesta del docente, de esta manera obtendremos valores positivos que indican cuan distante es la solución del profesor respecto de la del alumno, y si comparamos con miles de soluciones de muchos alumnos obtendremos las soluciones más similares.

- *Similitud por AST*. Permite calcular la similitud por medio de AST. Lo que se hace es recorrer el árbol abstracto. En el caso de HTML, cuando llega a una estructura definida en la gramática como son los HTMLElement —que son los elementos de HTML—, hacemos una evaluación y tratamiento de los contenidos y atributos que pueda contener el elemento. Esto se realiza utilizando el patrón Visitor. Este patrón visitor tiene un conjunto de clases elemento que conforman la estructura de un objeto. Cada una de estas clases elemento tiene un método aceptar (accept) que recibe al objeto visitante (visitor) como argumento. El visitante es una interfaz que tiene un método visit diferente para cada clase elemento; por lo que habrá implementaciones de la interfaz visitor de la forma: visitorClase1, visitorClase2 ... visitorClaseN. El método accept de una clase elemento llama al método visit de su clase. Clases concretas de un visitante pueden entonces ser escritas para hacer una operación en particular.

Al usar este patrón visitor, podremos saber cuándo está siendo llamada una parte de la gramática, es decir, cuando se recorre el árbol del AST sabremos a qué parte de la gramática se está invocando, y realizar las consultas y tratamientos de los datos, para hacer la comparación. Usamos esta misma técnica (el patrón visitor) para construir la estructura del AST del docente a partir de su rúbrica, el cual se almacena en una estructura de una pila, lo cual nos permite hacer la evaluación de similitud.

---

## 4. Propuesta

A continuación se enumeran los detalles de la propuesta:

- Los usuarios podrán interactuar y obtener información a través de preguntas de selección múltiple, tareas o ejercicios de un editor de programación, por lo que cualquier actividad estará controlada por tiempo.
- Estos registros formarán parte de la entrada de nuestro sistema de recomendación, junto al plan curricular, y el perfil del usuario.
- El sistema de recomendación evaluará y analizará los datos de entrada a través de un algoritmo probabilístico, para lo cual usaremos una red bayesiana.
- Al finalizar el componente de recomendación como salida nos dará una lista de objetos de aprendizaje a un determinado usuario sobre las competencias que le faltan.

### 4.1. Modelo

La Tabla 1 muestra el modelo de datos usado como propuesta en el sistema de recomendación de contenidos educativos. Daremos una breve explicación a continuación.

Un bloque básico para los modelos de datos es la tupla < tema, habilidad >. Indica el valor de las habilidades en un determinado tema. En nuestro prototipo de investigación, los temas han sido seleccionados a partir de una clasificación establecida a partir de una malla curricular de programación web de Laboratoria (2015) en el dominio de FrontEnd con HTML, CSS y JavaScript. Modelamos el nivel de habilidades para varios temas en una escala de cinco puntos. La tupla tema-habilidad tiene semánticas distintas en diferentes contextos. Por ejemplo, se describe el nivel de habilidades de un usuario en varios temas en el perfil de usuario. Por otra parte, indica la habilidad requerida en un tema en un contexto de tareas, y desarrollo a nivel de un editor de código de programación.

El perfil de usuario en su contexto programación web, generalmente, comprende competencias de los usuarios y sus metas de aprendizaje. La obtención de información confiable en cualquiera de los dos aspectos es un trabajo difícil. Hay dos fuentes de dicha información: (a) autodeclarado: según lo declarado por el alumno, por ejemplo, en una biodata o de su sitio web personal, y (b) evaluado: se obtiene a partir de una evaluación formal y registrado en algún sistema de gestión de aprendizaje o su término en inglés, *Learning Management System* (LMS). Si bien los resultados de la evaluación formal pueden ser más fiables, son generalmente menos granulares. Se utiliza sobre todo la información



---

según lo declarado por el alumno hacia sus competencias. Esta información ha sido subjetivamente validada con los registros disponibles en el LMS. Para el aprendizaje como meta, necesitamos combinar sus tareas, ejercicios en editor de programación y sus aspiraciones autodeclarados.

El contexto es uno de los aspectos más difíciles de entender y modelar. En general, el contexto puede incluir muchas facetas, entre otras, la personalidad del usuario, el medio ambiente donde está situado etc. (Dey, 2001). Nosotros hemos considerado dos aspectos principales: (a) el rol asignado a un usuario y (b) las tareas asignadas al usuario. En cualquier punto del tiempo, un usuario se le asigna un rol que define en términos generales las actividades que se espera realizar. Cada función requiere ciertas competencias.

Cada tarea requiere un conjunto de habilidades y debe ser completada en un plazo concreto. Nosotros clasificamos el marco temporal en tres categorías, de acuerdo a un análisis de estudio. Suponemos que el conjunto de tareas es repetitivo y existe una base de datos de tareas en la organización. Además, tratándose de un sistema enfocado a la programación, hay un módulo editor de programación donde los estudiantes suben y prueban sus códigos de acuerdo a un ejercicio planteado. Con el fin de integrar la variedad de recursos, un sistema de recomendación necesita utilizar algunos metadatos comunes. La biblioteca virtual de contenidos puede ser construida a través de un esfuerzo de colaboración de expertos en el tema y de referencias bibliográficas. Los metadatos de los contenidos de aprendizaje se pueden crear de forma manual o en una forma automatizada (Motelet & Baloian, 2006).

## 4.2. Algoritmo de recomendación

El sistema de recomendación está motivado por el marco genérico y conceptual para la organización propuesto en Shroff, Dey & Ghosh (2014). Diferentes atributos pertenecientes a los usuarios como el contenido y contexto en un ambiente de aprendizaje no puede ser estrictamente cuantificado. Además, las utilidades contextuales de diferentes atributos de los contenidos son inexactamente conocidos y, a veces pueden ser contradictorios. Un modelo de razonamiento probabilístico es necesario para lograr resultados sólidos; combinando estos, posiblemente encontremos conflictos con parámetros especificados. Usamos un modo de razonamiento mixto, en donde se determina la utilidad de un contenido de aprendizaje desde diferentes perspectivas: utilizando un modo de razonamiento casual de las redes bayesianas y combinando los resultados con distribución de probabilidad con una función sigmoide.

Una tarea se caracteriza por un arreglo de <Tema, Habilidad>. Adicionalmente, una tarea debe completarse en un tiempo finito. Por lo tanto, la utilidad de un objeto de aprendizaje con respecto a una tarea tiene dos componentes:

(a) Utilidad de contenido. Está determinada por la superposición de los temas presentes en el contenido y los que son requerido por la tarea. El logro de una habilidad está en función de la habilidad actual del alumno y la naturaleza del objeto de aprendizaje.

(b) Utilidad del tiempo. Cada tarea debe ser completada dentro de un tiempo finito. Por lo tanto la utilidad de tiempo de un objeto de aprendizaje se determina por una estimación de si se puede utilizar en el plazo deseado.

La determinación de la utilidad de tarea de un objeto de aprendizaje, que combina la utilidad de contenido y utilidad de tiempo, es un proceso muy importante. Para ello se tiene en cuenta que un usuario se le asigne más de una tarea. La utilidad de un objeto de aprendizaje con respecto a cada tarea se determina de forma independiente utilizando la red bayesiana.

A. Tupla Tema-Habilidad	
Tema	Uno o más conjuntos de temas de Plan estudios (Laboratoria, 2015)
Habilidad	Uno de {0:Ingenuo, 1:Conocimiento Basico, 2:Intermedio, 3:Avanzado, 4:Experto}
B. Modelo perfil de usuario	
Usuario-ID	Único identificador para un usuario
Nombre Usuario	El nombre del usuario
Competencias	Un arreglo de tupla < tema, habilidad >
C. Modelo Rol	
Rol-ID	Único identificador para un rol
Habilidad Requerida	Un arreglo de tuplas < tema, habilidad >
D. Modelo Tarea	
Tarea-ID	Único identificador para una tarea
Descripción Tarea	Descripción de la tarea asignada
Tema-Habilidad	Un arreglo de tuplas < tema, habilidad >
E. Modelo Editor	
Editor-ID	Único identificador para un ejercicio en Editor
Descripción	Descripción del ejercicio planteado
Similitud	{0:diferente, 1:similar 2:idéntico }
Tema-Habilidad	Un arreglo de tuplas < tema, habilidad >
F. Tupla Tarea-Tiempo	

Tarea-ID	Único identificador para una tarea
Tiempo	Uno de {0:Inmediato, 1:Corto, 2:Largo}
G. Asignación	
Usuario-ID	Único identificador para un usuario
Rol usuario	Uno de los rol-id
Tarea usuario	Un arreglo de tuplas < Tarea – ID, tiempo >
H. Modelando Contenido	
General/Identificador	Único identificador para el objeto de aprendizaje
General/Titulo	Nombre del objeto de aprendizaje
General/Autor	Autor del objeto de aprendizaje
General/Descripción	Descripción sobre el objeto de aprendizaje
General/Keyword	Temas seleccionados de un conjunto de temas
Educacional/Nivel Dificultad	Uno de {0:Muy Fácil, 1:Fácil, 2:Normal, 3:Difícil, 4:Muy Difícil}
Educacional/ Aprendizaje tipo recurso	Uno de {0:Categoría-1(E-book,Curso Web, etc.) 1:Categoría-2(Libro, Tutoriales, Presentaciones, etc.), 2:Categoría-3(Documentos referencias)}
Educacional/Típico tiempo aprendizaje	Uno de {0:Muy bajo(<=1/2 día), 1:Bajo((1/2)-2 días), 2:Medio(2-15 días), 3:Alto(15-45 días), 4:Muy alto(45 días)}
Localización/Técnica	URL del objeto de aprendizaje

Tabla 1. Modelo de datos usado para el sistema de recomendación

La utilidad de un ejercicio de un editor viene determinada por una red bayesiana, que combina la utilidad de la complejidad del ejercicio, con la utilidad del tiempo. Este proceso es muy importante dado que con solo preguntas de selección múltiples o preguntas no podremos saber qué tanto aprendió y/o puso en práctica. La utilidad de un objeto de aprendizaje asociado a un editor se determina de forma independiente, los resultados se combinan en una etapa posterior, al igual que la utilidad de una tarea respecto a un objeto de aprendizaje.

Un rol igual que la tarea se caracteriza por un arreglo de <Tema, habilidad>. Con todo, no hay marco de tiempo asociado con él. Nosotros interpretamos que las habilidades relacionadas con un rol se pueden adquirir en un plazo más largo. Por lo tanto, el cálculo para la utilidad de un objeto de aprendizaje con respecto al rol del usuario es similar a la estimación de la utilidad de contenido, tal y como se ha descrito más arriba.

Un usuario debe recibir recomendaciones de los contenidos de aprendizaje que se adapten a su actual situación. Estos no deben ser demasiado difíciles ni demasiado elementales para su nivel. Por tanto, la utilidad de un contenido de aprendizaje respecto al actual conocimiento del usuario se determina

por el conocimiento actual de este en el tema y la naturaleza del objeto de aprendizaje y su nivel de dificultad.

En los cuatro casos, los nodos raíz de las redes bayesianas se inician en ciertos estados, en base a los valores observados de algunos usuarios, tareas, ejercicios de editor y atributos del objeto de aprendizaje. Esto conlleva una propagación en la red. La probabilidad posterior de la variable de destino representa la creencia en el valor de uso del objeto de aprendizaje desde la perspectiva respectiva.

Debemos combinar los resultados de las etapas anteriores utilizando funciones de distribución de probabilidad sigmoide para producir un contexto global de la utilidad de un objeto de aprendizaje. Los valores de utilidad de las tareas para aquellas asignadas a un usuario se combinan para determinar un valor global de utilidad de la tarea, que se combina entonces con la utilidad del rol y la utilidad de usuario para determinar la utilidad del contexto global. La utilidad contextual para cada uno de los objetos de aprendizaje se determina con este modelo de razonamiento, que se utiliza posteriormente para el ranking de los objetos de aprendizaje, considerando la perspectiva abierta o Recursos Educativos Abiertos (Frango Silveira & Villalba-Condori, 2018).

### 4.3. Arquitectura del sistema de recomendación

Como podemos observar, la arquitectura del sistema de recomendación recibe múltiples entradas, como son: Entradas del usuario, plan curricular, perfil de usuario y el contexto de trabajo. A continuación detallaremos cada uno de estas entradas.

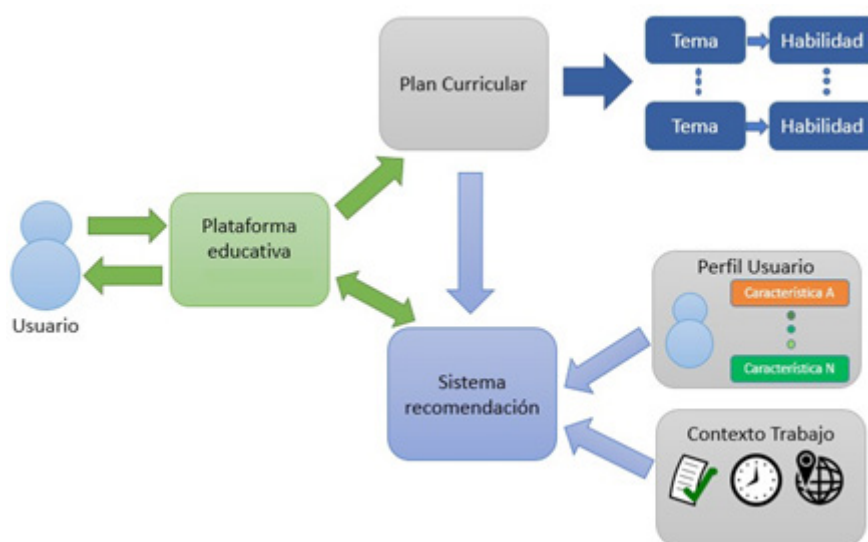


Figura 3. Arquitectura del modelo de aprendizaje para sistemas de recomendación basado en contexto

- Plataforma educativa. En este módulo encontramos la parte de interfaz de usuario y cómo se relaciona el usuario con el sistema. En el caso de estudio, al ser programación web, el usuario, sus entradas, son tareas, preguntas de selección múltiple, ejercicios de programación a través de un editor de códigos.

- Plan curricular. El plan curricular está formado por un conjunto de tuplas que están basados en tema, habilidad; cada unidad del plan curricular está conformado por esa tupla. Además, este plan curricular está enfocado a lo que es programación web.
- Perfil usuario. En el perfil de usuario encontramos los datos del estudiante, tales como nombre, edad, sexo, correo electrónico, etc. Así mismo el perfil del usuario tiene un campo que son las habilidades y/o competencias que tiene que desarrollar en el transcurso del curso de programación web.
- Contexto trabajo. El contexto de trabajo viene a estar formado por todas las actividades que realiza el usuario y todo lo que le rodea, de donde podemos obtener información que es relevante para el sistema de recomendación. Dentro del contexto de trabajo se está tomando lo que son las tareas, ejercicios de programación, tiempos en que se demora y/o estimaciones según la complejidad de las actividades que se les deja a los estudiantes.
- Sistema recomendación. Este módulo se encarga de recibir todas las entradas, y hacer un preprocesamiento, análisis e interpretación de los resultados a través de un modelo de razonamiento probabilístico, basado en redes bayesianas.

#### 4.4. Arquitectura del modelo de razonamiento probabilístico

En la Figura 4 se muestra un modelo de Red bayesiana para el cálculo del aprendizaje según las tareas resueltas. Las entradas a la red bayesiana están conformadas por las competencias del usuario respecto a un tema (u-c), las preguntas de la tarea (p), la dificultad del contenido de la tarea (c-d), el tiempo estimado para la tarea que se le deja al estudiante (t) y la complejidad de acuerdo con un tiempo planteado a un estudiante (c-t).

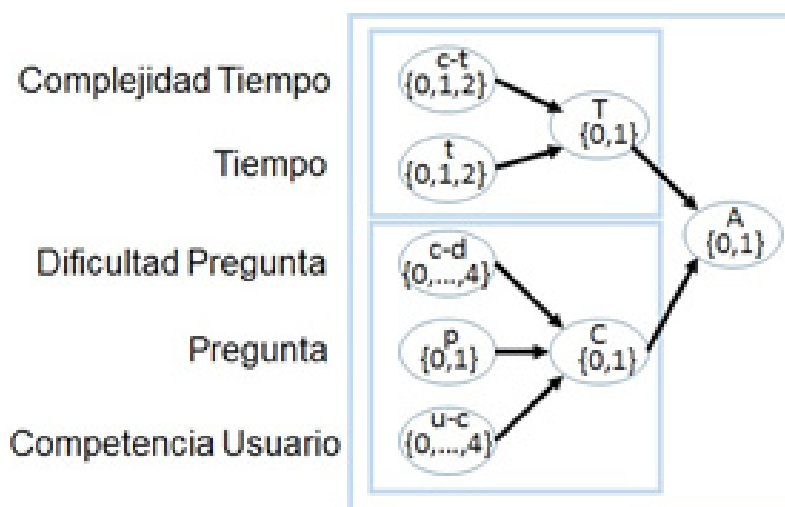


Figura 4. Red bayesiana para el cálculo del valor de aprendizaje en tareas

Para esta red tomamos como entrada las competencias iniciales del usuario respecto a un contenido; las preguntas, si contesto bien o no; la dificultad del contenido; el tiempo, y la complejidad del contenido respecto al tiempo. Esta arquitectura se debe aplicar para cada una de las tareas que se tiene para un usuario. Finalmente, en la siguiente etapa, se hace un análisis de todos los valores de aprendizaje por cada tarea, conformando un conjunto de redes bayesianas para tareas.

En la Figura 5 se muestra el modelo de red bayesiana para el cálculo de aprendizaje de acuerdo a los ejercicios que se plantea para el desarrollo de un curso de programación. Para ellos las entradas son las notas del editor; ast es el análisis del código a nivel de su representación abstracta que tomaría los valores de evidencia de: idéntico, similar, diferente. El otro parámetro de entrada es la dificultad del ejercicio, el tiempo y la complejidad respecto al tiempo.

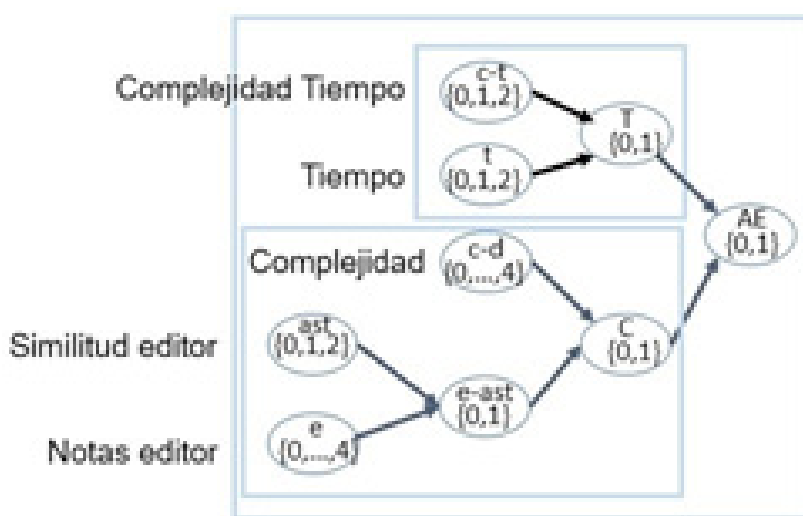


Figura 5. Red bayesiana para el cálculo del aprendizaje en Editor de programación

Las entradas a la red están formadas por las notas del editor (e). Estas notas están dadas por el profesor del curso. La similitud en que la solución del usuario se parece, dado una rúbrica del profesor ast, la cual esta similitud la da un sistema independiente a la nota que da el profesor; la dificultad del contenido para resolver ese ejercicio (c-d); el tiempo estimado (t) para resolución del problema; la complejidad según un tiempo planteado a un estudiante (c-t), al final nos dará una salida, que es el aprendizaje en el módulo del editor de programación.

En la Figura 6 se muestra el modelo de red bayesiana para el cálculo la utilidad del contenido respecto a un usuario y su dificultad (u-c y c-d).

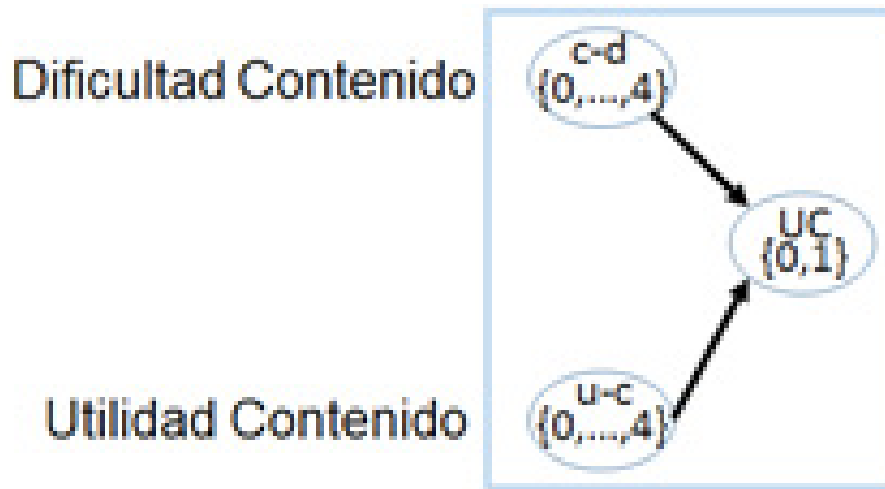


Figura 6. Red bayesiana para el cálculo de la utilidad de un contenido

En la Figura 7 podemos ver que tenemos otro conjunto de redes bayesianas para cada uno de los ejercicios que se resolvió con el editor de programación, que luego se hace un post procesamiento para obtener un único aprendizaje por editor, representado por AE. Podemos observar que tenemos también el cálculo de la utilidad del contenido, representado por UC. Tanto A, AE y UC serán post procesados para obtener el aprendizaje respecto a un contenido específico que está identificado por A-C, para lo que usamos una función de distribución probabilístico sigmoide

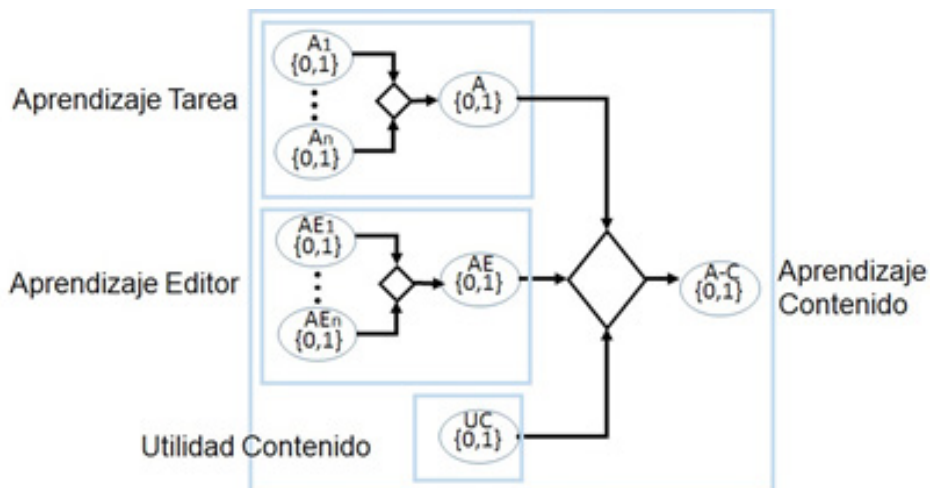


Figura 7. Conjunto de redes bayesianas para el cálculo del aprendizaje del contenido

---

La función sigmoide, como muchos procesos naturales y curvas de aprendizaje de sistemas complejos, muestra una progresión temporal desde unos niveles bajos al inicio, hasta acercarse a un clímax transcurrido un cierto tiempo; la transición se produce en una región caracterizada por una fuerte aceleración intermedia. La función sigmoide permite describir esta evolución a través de la siguiente ecuación.

$$y = 1/(1 + e^x)$$

La representación gráfica está dada por una función sigmoide.

## 5. Resultados

La muestra de población son estudiantes de género femenino, con edades comprendidas entre los 21 a los 29 años, nacidas en Perú, México y Chile. En total, 323 estudiantes.

La evaluación se realizó de tres maneras:

En primer lugar se hicieron las pruebas sobre datos reales. Estos están almacenados en la base de datos del "Capitán". Se realizó una lectura de todos los ejercicios propuestos sobre HTML por el profesor: en total se encontraron nueve ejercicios y se obtuvieron 2.061 soluciones enviadas por las estudiantes (Tabla 2).

En segundo lugar se ejecutaron pruebas con datos sintéticos. Es decir, se generaron datos aleatorios a partir de la solución del profesor, utilizando el AST de la rúbrica del profesor respecto a su solución óptima, y recorriendo el AST se generaron errores de sintaxis, errores de contenido y estructuración del código. Se generaron 1.000 soluciones por cada ejercicio propuesto, obteniéndose en total 9.000 soluciones para analizar (Tabla 2).

En tercer lugar se realizaron pruebas con las tareas de las chicas que, en conjunto, con los ejercicios del editor, sumando la medida de similitud por parte de un análisis del AST, obtendremos las recomendaciones y la precisión de aquellas, tal como se muestra en la Figura 8.

	Datos reales	Datos sintéticos
Total problemas analizados	2.061	9.000
Error sintaxis	75	41.708
Error diseño	39	12.510
Recomendaciones	35	5.490

Tabla 2. Evaluación de Resultados



En la Tabla 3 se muestran los resultados de la evaluación de los estudiantes, obtenidos a partir de las encuestas que se les realizaron, tras haber interactuado con el sistema. La muestra poblacional de la encuesta recayó en 30 estudiantes de Laboratorio, con edades que oscilan entre los 18 y los 29 años, de la sede de Arequipa. La encuesta tuvo como finalidad conocer la precisión del sistema, la parte intuitiva del mismo, si consideran las recomendaciones eficientes, y si los elementos que necesitaban eran conformes a las recomendaciones proporcionada por el sistema.

Descripción de la pregunta	Porcentaje
¿Le parece muy intuitivo el sistema?	80% intuitivo
¿El sistema es eficiente en cuanto a recomendaciones?	80% eficiente
¿Lo que necesito se pueden encontrar en la lista de recomendaciones?	66.7% sí
¿Los elementos de la lista de recomendaciones satisfacen mis necesidades?	70% sí
¿Es necesario contar con un sistema recomendación para problemas de editor?	93,8% sí

Tabla 3. Resultados de la evaluación del estudiante

Para saber la precisión de las recomendaciones, en la Figura 8 se construye el gráfico promedio interpolado de recall-precision (Nist, 2013) para cada una de las recomendaciones, se establece para un usuario de la siguiente manera: se representa  $U_i$  como el conjunto de Objetos de Aprendizaje que el usuario  $i$  considera contextualmente útil y deja que  $R_i$  represente la recomendación establecida. El recall-precision de la recomendación del usuario  $i$  se calculan como  $R_i = |U_i \cap R_i| / |U_i|$  y  $P_i = |U_i \cap R_i| / |R_i|$  respectivamente. Repitiendo los experimentos sobre  $K$  usuarios nosotros obtenemos  $K$  tuplas  $\{ \langle R_1; P_1 \rangle, \dots, \langle R_k; P_k \rangle \}$ . Interpolamos estas llamadas y valores de precisión para obtener el gráfico de promedio interpolado de recall-precision.

Los resultados se obtuvieron a partir del pre-procesamiento y análisis de las entradas de los usuarios. Para ello, se evaluó a cada uno de los estudiantes según sus tareas y ejercicios de programación y formando parte de la entrada del conjunto de redes bayesianas. Y por cada usuario respecto a una unidad se recomienda y hace una retroalimentación según las unidades que debe volver a repasar para completar las competencias que le son necesarias.

En la Figura 8 se muestra el gráfico recall-precisión. Este gráfico se obtuvo evaluando todos los usuarios, y los usuarios que tenían recomendaciones, se comparó con los resultados esperados contra los resultados del sistema de recomendación.

Como vemos en la gráfica, las recomendaciones tienen buena precisión, lo que demuestra que el

modelo probabilístico hace recomendaciones que se ajustan a las recomendaciones sugeridas. Los valores muestran una precisión del sistema de recomendación entre el 66% y el 100%.

En la Tabla 4 podemos ver la cantidad de objetos de aprendizaje que fueron retornados por unidad desarrollada durante el transcurso de programación web.

En la Figura 9 se muestra el porcentaje de ítems recomendados por unidad, después de evaluar todos los usuarios sus tareas y editores de código.

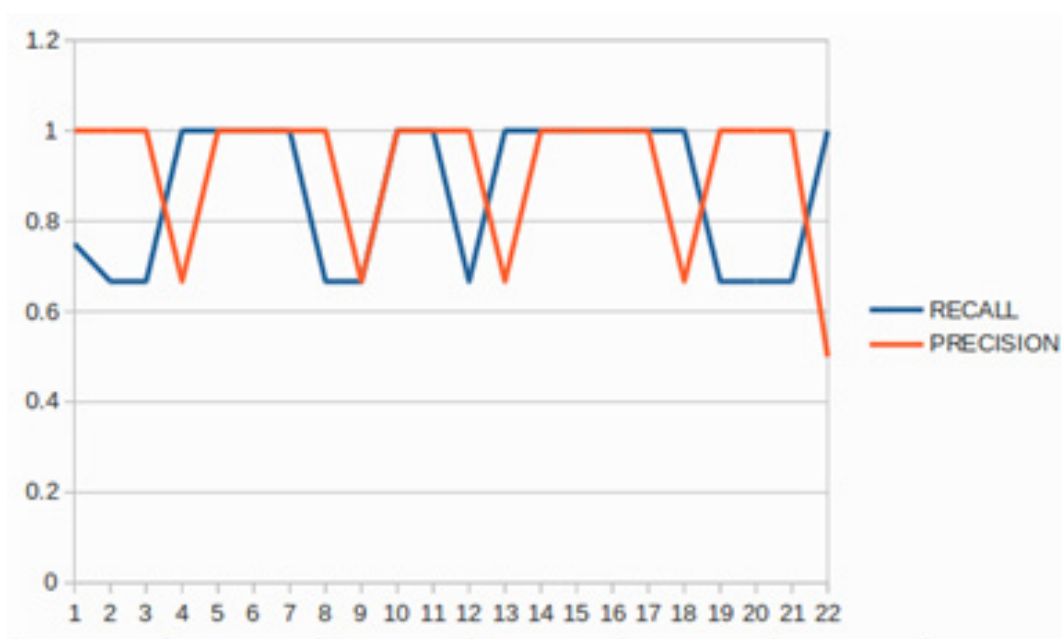


Figura 8. Gráfico recall-precision para el sistema de recomendación

<b>Unidad</b>	Porcentaje
Curso de introducción al desarrollo web	29 %
Diagnóstico de entrada (HTML / CSS / JS)	41 %
Unidad 1: HTML /CSS	30 %

Tabla 4. Tabla de cantidad de tems recomendados por unidad

Al ser este un sistema de recomendación para el caso de programación web, con un módulo de analizador de código estático, hay muchos sistemas similares. A continuación en la Tabla 5 se comparará las características que cuenta esta propuesta respecto de otras propuestas con características similares.

	Similaridad	Clustering	SR	Errores Sintaxis	Error Estructura Contenido
Capitan	x	x	x	x	x
Overcode	x	x		x	
Solminer	x	x			
CodeOp- tion				x	x
AFD	x	x		x	

Tabla 5. Tabla de comparación de sistemas similares a la propuesta

AFD es Automated Feedback Generation y SR es Sistema de recomendación. Como se observa, la mayoría de los sistemas se centran exclusivamente en análisis de errores de sintaxis, y muchos no cuentan con un feedback o un sistema de recomendación según las habilidades que le competen a un usuario, que le permitan mejorar sus competencias respecto de un tema.

## 6. Discusión de resultados

De los resultados obtenidos en esta investigación, se menciona solo los objetos que han sido recomendados; no se ha considerado los que no fueron recomendados, debido a que estos estudiantes cumplieron con las competencias que son necesarias para su aprendizaje.

Los resultados son los siguientes: del total de 323 estudiantes, 50 obtuvieron recomendaciones; de los 50 estudiantes, 22 obtuvieron alguna incidencia en su recomendación como es la falta de una recomendación; 28 estudiantes recibieron recomendaciones de acuerdo a su competencia. Aquellos 22 estudiantes realizaron el recall-precision, como se observa en la Figura 8.

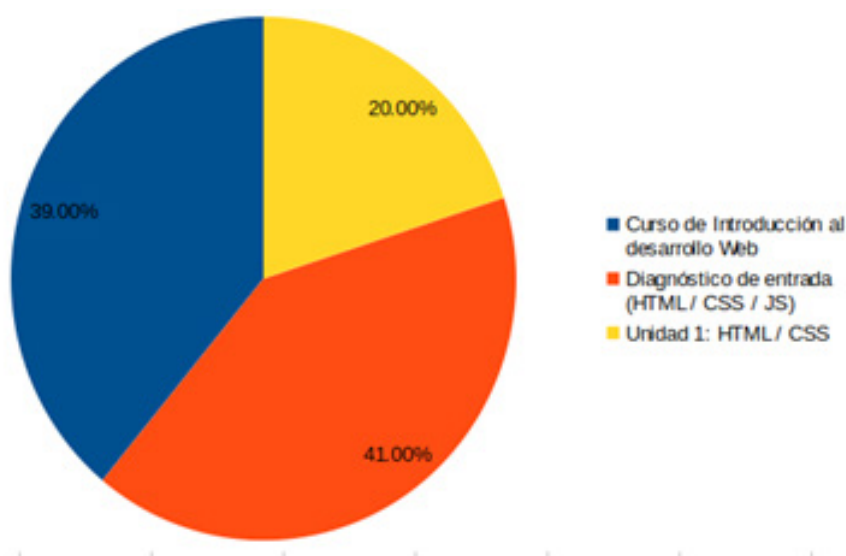


Figura 9. Gráfico de porcentaje de ítems recomendados por unidad

---

Los resultados obtenidos por las medidas de similitud nos muestran cuánto se asemejan las soluciones en el módulo del editor de códigos, y nos servirá como entrada a nuestro modelo de razonamiento probabilístico. En la Tabla 2 podemos ver que los 323 estudiantes generaron 2.061 resultados reales, que se evaluarán en la siguiente etapa.

## 7. Conclusiones

En este trabajo se ha realizado una investigación, análisis y desarrollo de los sistemas de recomendación basados en el contexto del usuario, usando un modelo de razonamiento probabilístico para el caso de programación web. Se ha realizado una revisión de los conceptos matemáticos probabilísticos y su implementación en una red Bayesiana, mostrándonos que los valores de precisión de las recomendaciones estaban entre el 66% y el 100%.

En el desarrollo del modelo de recomendación basado en contexto se ha visto la importancia que tiene la parte de desarrollo de ejercicios, y que este debe tener mayor peso. Si bien es cierto que la parte conceptual es importante, no es menor la importancia de la práctica, por lo que se realizó el proceso de modelación de una red bayesiana para este tipo de problemas.

De la investigación se deduce que los modelos tradicionales para hacer un sistema de recomendación a partir de un matching, de las entradas del usuario como son las tareas respecto a las notas de un alumno, se pierde mucha información imprescindible para analizar las competencias necesarias para el alumno en su proceso de aprendizaje.

Las principales contribuciones del trabajo son:

- Propuesta de una arquitectura de un sistema de recomendación de un usuario que toma como entradas, su plan curricular y su contexto (tareas, ejercicios, tiempos, etc.).
- Propuesta de un modelo de razonamiento probabilístico basado en redes bayesianas para las tareas de un usuario.
- Propuesta de un modelo de razonamiento probabilístico basado en redes bayesianas para los ejercicios en un editor de código.
- Propuesta de un modelo de datos para el contexto de un usuario.
- La importancia y la relación que existe entre un usuario y las competencias que necesita desarrollar.

---

## 8. Referencias

- Altuna C. E. J., & Estrada, L. G. (2013). Generación de pistas durante el aprendizaje de la programación para concursos usando el análisis estático y dinámico de las soluciones. *Ingeniare. Revista chilena de ingeniería*, 21(2), 205-217. doi:<http://dx.doi.org/10.4067/S0718-33052013000200005>
- Baojiang, C., Jiansong, L., Tao, G., Jianxin, W., & Ding, M. (2010). Code Comparison System based on Abstract Syntax Tree 2010 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT) (Beijing, China, 26-28 Oct. 2010) (pp. 668-673). USA: IEEE. doi: <http://dx.doi.org/10.1109/ICBNMT.2010.5705174>
- Batallanos, C. A., Poccohuanca, O. Q., & Sayco, S. C. C. (2017). Modelo de sistema de recomendación de objetos para incentivar el desarrollo del pensamiento computacional. *Revista Referencia Pedagógica*, 5(1), 96-108.
- Carrillo, G., & Ochoa, X. (2013). *Recomendación de objetos de aprendizaje basado en el perfil del usuario y la información de atención contextualizada*. Paper presented at the 8th Latin American Conference on Learning Objects and Technologies-LACLO, Valdivia, Chile.
- Cazella, S. C., Nunes, M. A. S. N., & Reategui, E. (2010). *A Ciência da Opinião: Estado da arte em Sistemas de Recomendação*. Paper presented at the Jornada de Atualização de Informática, JAI 2010 - CSBC2010, Rio de Janeiro, Brazil.
- Chen, M., Tong, M., Liu, C., Han, M., & Xia, Y. (2017). Recommendation of Learning Path Using an Improved ACO Based on Novel Coordinate System 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (Hamamatsu, Japan, 9-13 July 2017) (pp. 747-753). USA: IEEE. doi:<http://dx.doi.org/10.1109/IIAI-AAI.2017.90>
- D'Abreu, J. V. V., & Villalba-Condori, K. O. (2017). Educación y Robótica Educativa. *Revista de Educación a Distancia*, (54). doi:<http://dx.doi.org/10.6018/red/54/11>
- Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 4-7. doi:<http://dx.doi.org/10.1007/s007790170019>
- Fidalgo-Blanco, Á., Sein-Echaluce, M. L., & García-Peñalvo, F. J. (2014). Knowledge spirals in higher education teaching innovation. *International Journal of Knowledge Management*, 10(4), 16-37. doi:<http://dx.doi.org/10.4018/ijkm.2014100102>

---

Fidalgo-Blanco, Á., Sein-Echaluce, M. L., & García-Peñalvo, F. J. (2015). Epistemological and ontological spirals: From individual experience in educational innovation to the organisational knowledge in the university sector. *Program: Electronic library and information systems*, 49(3), 266-288. doi:<http://dx.doi.org/10.1108/PROG-06-2014-0033>

Frango Silveira, I., & Villalba-Condori, K. O. (2018). An Open Perspective for Educational Games. *Journal of Information Technology Research*, 11(1), 18-28. doi:<http://dx.doi.org/10.4018/JITR.2018010102>

García-Peñalvo, F. J. (2016). A brief introduction to TACCLE 3 – Coding European Project. In F. J. García-Peñalvo & J. A. Mendes (Eds.), *2016 International Symposium on Computers in Education (SIIE 16)*. USA: IEEE. doi:<http://dx.doi.org/10.1109/SIIE.2016.7751876>

García-Peñalvo, F. J., & Mendes, J. A. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. doi:<http://dx.doi.org/10.1016/j.chb.2017.12.005>

García-Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). *An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers*. Belgium: TACCLE3 Consortium. <http://doi.org/10.5281/zenodo.165123>

Laboratoria (2015). Code that transforms. Retrieved from <http://laboratoria.la>.

Lavonen, J., Korhonen, T., & Villalba-Condori, K. O. (2018). Descripción de las estrategias colaborativas frente a los retos de la educación finlandesa / Description of collaborative strategies to meet the challenges of Finnish education. *Revista Referencia Pedagógica*, 6(1), 86-100.

Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. *Proceedings of the 20th ACM conference on Hypertext and hypermedia, HT '09 (Torino, Italy – June 29 - July 01, 2009)* (pp. 73-82). New York, NY, USA: ACM. doi:<http://dx.doi.org/10.1145/1557914.1557930>

Motelet, O., & Baloian, N. (2006). Hybrid System for Generating Learning Object Metadata. *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06) (Kerkrade, Netherlands, 5-7 July 2006)* (pp. 563-567). USA: IEEE. doi:<http://dx.doi.org/10.1109/ICALT.2006.1652504>

Newcomb, P. (2005). Abstract Syntax Tree Metamodel Standard ASTM Tutorial 1.0.

---

Retrieved from <https://goo.gl/SdF9ky>.

Niemann, K., Scheffel, M., Friedrich, M., Kirschenmann, U., Schmitz, H.-C., & Wolpers, M. (2010). Usage-based object similarity. *Journal of Universal Computer Science*, 16(16), 2272–2290. doi:<http://dx.doi.org/10.3217/jucs-016-16-2272>

Nist (2013). Common evaluation measures. *The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings*. <https://goo.gl/Hjp9MM>.

Owens, D., & Anderson, M. (2013). A generic framework for automated Quality Assurance of software models - Application of an Abstract Syntax Tree 2013 *Science and Information Conference, SAI, (London, UK, 7-9 Oct. 2013)* (pp. 207-211). USA: IEEE.

Parr, T. (2014). ANTLR. Retrieved from <http://www.antlr.org/>.

Rodríguez, P. A., Duque, N. D., & Ovalle, D. A. (2016). Modelo híbrido de recomendación Adaptativa de Objetos de Aprendizaje basado en Perfiles de Usuario. *Formación universitaria*, 9(4), 83–94. doi:<http://dx.doi.org/10.4067/S0718-50062016000400010>

Saraswat, N., Ghosh, H., Agrawal, M., & Narayanan, U. (2015). Contextual Recommendation of Educational Contents. In C. Conati, N. Heffernan, A. Mitrovic, & M. F. Verdejo (Eds.), *Artificial Intelligence in Education. Proceedings of the 17th International Conference, AIED 2015, Madrid, Spain, June 22-26, 2015* (pp. 439-448). Cham: Springer International Publishing. doi:[http://dx.doi.org/10.1007/978-3-319-19773-9\\_44](http://dx.doi.org/10.1007/978-3-319-19773-9_44)

Shankar, U., Talwar, K., Foster, J. S., & Wagner, D. (2001). Detecting format string vulnerabilities with type qualifiers. *Proceedings of the 10th conference on USENIX Security Symposium, SSYM'01 (Washington, D.C. – August 13 - 17, 2001)* (Vol. 10, Article No. 16). Washington, D.C.: USENIX Association.

Shroff, G., Dey, L., & Ghosh, H. (2014). Enterprise Contextual Intelligence. *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), WI-IAT '14* (Vol. 2, pp. 202-209). USA: IEEE Computer Society. doi:<http://dx.doi.org/10.1109/wi-iat.2014.99>

TACCLE 3 Consortium. (2017). TACCLE 3: Coding Erasmus + Project website. Retrieved from <https://goo.gl/f4QZUA>

Tripp, A. (2006). Manual Tree Walking Is Better Than Tree Grammars. Retrieved from <https://goo.gl/6vyFKn>.

---

Vera, J., Mamani, A. O., & Villalba-Condori, K. O. (2015). Modelo de sistema de recomendación de Objetos de Aprendizaje en dispositivos móviles, caso: Desarrollo del pensamiento computacional. *XX Congreso Internacional de Informática Educativa, TISE 2015. Nuevas Ideas en Informática Educativa (Santiago, Chile, 1-3 de diciembre de 2015)* (pp. 730-734).

Villalba-Condori, K. O. (2018). Teaching Formation to develop Computational thinking. In F. J. García-Peñalvo (Ed.), *Global Implications of Emerging Technology Trends* (pp. 59-72). Hershey PA, USA: IGI Global. doi:<http://dx.doi.org/10.4018/978-1-5225-4944-4.ch004>

Villalba-Condori, K. O., Castro Cuba, S., Deco, C., Bender, C., & García-Peñalvo, F. J. (2017). A Recommender System of Open Educational Resources based on the Purpose of Learning. In A. Díaz, A. Casali, M. Chacón Rivas, & A. Silva Sprock (Eds.), *Proceedings of 2017 Twelfth Latin American Conference on Learning Technologies - LACLO 2017 (9-12 October 2017, La Plata, Argentina)* (pp. 104-107). EEUU: IEEE. doi:<http://dx.doi.org/10.1109/LACLO.2017.8120899>

Wassermann, G., & Su, Z. (2007). Sound and precise analysis of web applications for injection vulnerabilities. *ACM SIGPLAN Notices*, 42(6), 32-41. doi:<http://dx.doi.org/10.1145/1273442.1250739>

Wiley, D. A. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*: Agency for Instructional Technology.

Worsley, M., & Blikstein, P. (2013). Programming Pathways: A Technique for Analyzing Novice Programmers' Learning Trajectories. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Artificial Intelligence in Education. Proceedings of the 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013* (pp. 844-847). Berlin, Heidelberg: Springer. doi:[http://dx.doi.org/10.1007/978-3-642-39112-5\\_127](http://dx.doi.org/10.1007/978-3-642-39112-5_127)