



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

01 de marzo de 2013

Concepción – Chile

Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática

Marcelo Torres Ríos
Teresa Ulloa Placencia

Profesor guía
Clemente Rubio Manzano

TESIS
para obtener el grado de
INGENIERO CIVIL EN INFORMÁTICA

**“Todo gran cometido debe tener un principio,
pero es en la continuidad hasta el final,
hasta donde se ha acabado totalmente,
donde está la verdadera gloria.”**

Sir Francis Drake, 1587

**A mi familia, por todo su apoyo
y a todos quienes confiaron en mi
con altas expectativas.**

Marcelo Torres Ríos.

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para la carrera de *Ingeniería Civil en Informática*. El proyecto titulado *Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática* consiste en una propuesta que nace por la escasa información sobre los electivos que se impartirán, lo que dificulta la decisión al momento de inscribir un electivo. El sistema es una aplicación web desarrollada en ASP.NET; que, si bien su objetivo principal es recomendar electivos a los estudiantes en base a sus preferencias, cuenta con las características necesarias para revisar la información de cada uno de los electivos e inscribirlos, resolviendo así el problema de origen.

El presente documento consta principalmente de dos partes; una sección de investigación, donde se expone el origen de los sistemas de recomendación, sus características y enfoques más importantes. La segunda parte consiste en la documentación del desarrollo de la propuesta, bases y fundamentos, características del sistema y la solución final al problema.

Abstract

This project is presented according to the Bío-Bío University requirements, in the certification process for the *Civil Engineering in Computer Science* degree. The project entitled *Elective Recommender System for Civil Engineering in Computer Science students*, is a proposal that born because of the limited information about elective that will be taught, which hampers the decision in the moment to enroll an elective. The system is a web application developed in ASP.NET, which, although its main purpose is to recommend elective to students based on their preferences, has the necessary characteristics to review the information from each elective and register it, solving so the origin problem.

This document consists mainly of two parts, a research section, which explains the origins of recommender systems, its most important features and approaches. The second part is the documentation of the development of the proposal, bases and foundations, system characteristics and the final solution to the problem.

Índice General

1. INTRODUCCIÓN	9
2. PRELIMINARES	12
2.1. Sistemas de recomendación	12
2.1.1. Qué son los sistemas de recomendación	12
2.1.2. Origen de los sistemas de recomendación.....	12
2.1.3. Para qué sirven los sistemas de recomendación	13
2.1.4. En qué se han utilizado los sistemas de recomendación	13
2.1.5. Problemas generales	14
2.1.6. Principales enfoques	17
2.1.7. Principales técnicas.....	21
2.2. Herramientas de desarrollo	24
2.2.1. PHP.....	25
2.2.2. Java	26
2.2.3. ASP.NET	27
3. DEFINICIÓN DEL PROYECTO.....	31
3.1. Objetivos del proyecto.....	31
3.2. Ambiente de Ingeniería de Software	31
4. CARACTERÍSTICAS DEL SOFTWARE.....	35
5. BASES DE LA RECOMENDACIÓN.....	36
5.1. Diseño de la recomendación	36
5.1.1. Recomendación basada en preferencias.....	36
5.1.2. Recomendación basada en calificaciones	42
6. ANÁLISIS.....	51
6.1. Diagrama de casos de uso.....	51
6.1.1. Actores	51
6.1.2. Casos de uso y descripción	52
6.2. Modelamiento de datos.....	63
7. DISEÑO	64
7.1. Diseño físico de la base de datos	64

7.2. Diseño de arquitectura funcional.....	65
7.3. Diseño de interfaz y navegación	66
8. IMPLEMENTACIÓN.....	67
8.1. Funcionamiento del sistema.....	67
9. CONCLUSIONES Y TRABAJO FUTURO	86
10. ANEXO.....	89

Índice de Tablas

Tabla 1: Aceptación/Valor.....	37
Tabla 2:Asignatura/Especialidad.....	43

Índice de Ilustraciones

Ilustración 1: Estructura del framework .NET	29
Ilustración 2: Representación gráfica del Modelo Cascada con reducción de riesgos	32
Ilustración 3: Gráfico del grado de desempeño en base a calificaciones promedio	45
Ilustración 4: Diagrama de Casos de uso del sistema	52
Ilustración 5: Modelo Entidad-Relación – Vista general	63
Ilustración 6: Modelo Relacional - Vista general	64
Ilustración 7: Diseño de arquitectura funcional del sistema	65
Ilustración 8: Diseño de interfaz del sistema	66
Ilustración 9: Interfaz general del sistema	67
Ilustración 10: Establecer fecha máxima de preinscripción	68
Ilustración 11: Gestión de electivos - Jefe de carrera	68
Ilustración 12: Módulo de ingreso de nuevo electivo.....	69
Ilustración 13: Lista de estudiantes pre-inscritos por electivo.....	70
Ilustración 14: Gestión d electivos - Profesor.....	71
Ilustración 15: Gestión de requisitos	73
Ilustración 16: Lista de afirmaciones que influyen en un electivo	75
Ilustración 17: Encuesta para establecer preferencias	77
Ilustración 18: Vista si la encuesta ya se respondió.....	78
Ilustración 19: Lista de recomendaciones obtenidas.....	79
Ilustración 20: Lista de electivos a impartir en el semestre	81
Ilustración 21: Lista de electivos que se puede cancelar su preinscripción	82
Ilustración 22: Recuperación de datos de ingresos	83
Ilustración 23: Formulario para envío de Informes o sugerencias	84
Ilustración 24: Formulario para cambio de contraseña	85

1. INTRODUCCIÓN

La malla curricular de Ingeniería Civil en Informática de la Universidad del Bío-Bío está formada por seis áreas de formación diferentes, que son: Ciencias Básicas, Ciencias de la Ingeniería, Ingeniería de Software y Base de Datos, Sistemas Computacionales, Gestión en Informática y Formaciones generales y otros. En estas áreas de formación se clasifican las asignaturas que se imparten, de acuerdo a sus características y las competencias necesarias para aprobarlo.

Cada año se imparten diferentes asignaturas electivas, cada una de ellas se clasifica dentro de una o más áreas de formación, entonces, si un alumno tuvo buen rendimiento en un área determinada se podría esperar un rendimiento similar en un electivo relacionado con la misma. A la vez, los intereses de un estudiante también influyen en el desempeño que éste tiene en una asignatura, por lo que se presume que si una asignatura cumple con las expectativas del alumno, su rendimiento será mayor, por el contrario, si no las cumple, el rendimiento será menor.

Para los estudiantes no resulta fácil elegir asignaturas electivas, ya que muchas veces no cuentan con la información necesaria para tomar una buena decisión, y prácticamente terminan escogiendo a ciegas. Además no todos los semestres se imparten los mismos electivos de manera regular, lo que justifica la falta de información en algunos casos.

Por lo general, para escoger un electivo, los estudiantes preguntan a otros compañeros qué electivos tomaron ellos y la impresión que se llevaron al momento de cursar la asignatura, para luego optar por aquellos que más les convienen y/o son más atractivos para sus intereses, ya sea si sus habilidades están orientadas al área comercial, de investigación o desarrollo, por nombrar algunos. Por ello, esta propuesta busca agilizar el proceso de elección e inscripción, permitiendo a los estudiantes tomar electivos seleccionados en base a sus gustos y desempeño académico, logrando una mayor expectativa de rendimiento.

Considerando esta situación, se cree factible el desarrollo de un sistema recomendador que permita, mediante el análisis de los intereses de los alumnos y su rendimiento previo, recomendar asignaturas electivas en las que podrían desempeñarse mejor y con mayor satisfacción, ya que los sistemas de recomendación son considerados como una excelente herramienta, que permite agilizar la toma de decisiones.

El sistema hará dos tipos de recomendaciones, una basada en los gustos de los estudiantes, cuyas preferencias se obtendrán mediante encuestas que permitan inferir qué electivos serán de mayor agrado para cada alumno. El segundo tipo de recomendación busca predecir cuáles serán los electivos donde el estudiante tendrá mejores resultados en cuanto a calificaciones, para ello se tendrán en cuenta las calificaciones de las asignaturas relacionadas a los electivos.

Para realizar las recomendaciones se han desarrollado algoritmos propios, que consideran todas las variables necesarias para solucionar de la mejor forma la problemática identificada. Se han diseñado fórmulas para cada tipo de recomendación y han sido traspasadas a código.

Se ha tomado un enfoque basado en contenido, lo que permite relacionar las características de un electivo a los intereses de los estudiantes, de una manera más cercana a la realidad; por otra parte la relación entre las calificaciones de asignaturas requeridas en un electivo, también es más veraz.

En resumen, el objetivo de esta propuesta es agregar mecanismos de recomendación al proceso de inscripción de electivos, permitiendo a los estudiantes tomar electivos con mayor información, en base a sus gustos y preferencias y al rendimiento previo obtenido en asignaturas relacionadas, con lo que se espera un mayor rendimiento en el transcurso del ramo.

A continuación se detalla la estructura general de este documento:

- En este capítulo se ha hecho una pequeña introducción al proyecto.
- **Capítulo 2 – Preliminares:** Trabajo investigativo que busca mostrar al lector los sistemas de recomendación, características principales, enfoques y algoritmos más comunes. También una pequeña investigación sobre los posibles lenguajes de programación a usar.
- **Capítulo 3 – Definición del proyecto:** Se exponen los objetivos del proyecto, la metodología de desarrollo a seguir y las herramientas a utilizar.
- **Capítulo 4 – Bases de la recomendación:** Se exponen los tipos de recomendación que hará el sistema, sus características y fórmulas.
- **Capítulo 5 - Análisis:** Se describen los casos de uso y modelos de datos.
- **Capítulo 6 - Diseño:** Se expone el diseño físico, la arquitectura funcional y la interfaz gráfica del sistema.

- **Capítulo 7 – Implementación:** Se describe y expone el funcionamiento del sistema para cada uno de los usuarios.
- **Capítulo 8 Conclusiones y trabajo futuro:** Se exponen las conclusiones sobre el desarrollo de la tesis y las tareas que se esperan hacer al corto y largo plazo.
- Por último se añade un anexo donde se detallan los atributos del modelo relacional y manuales de usuario para una mejor utilización del sistema.

2. PRELIMINARES

2.1. Sistemas de recomendación

2.1.1. Qué son los sistemas de recomendación

Algunas definiciones formales sobre sistemas recomendadores se citan a continuación:

Se trata de aquel sistema que tiene como principal tarea seleccionar ciertos objetos de acuerdo a los requerimientos del usuario. [1]

El sistema que utiliza las opiniones de los usuarios de una comunidad para ayudar a usuarios de esa comunidad a encontrar contenidos de su gusto entre un conjunto sobrecargado de posibles elecciones. [1]

Por lo tanto, los sistemas recomendadores son una herramienta que busca predecir los intereses de los usuarios y de esta forma sugerirle temas de interés, optimizando toda una búsqueda en un mundo donde existe muchísima información; son un tipo particular de aplicaciones y de técnicas especialmente desarrolladas para filtrar información [2]. El proceso de recomendación tiene tres acciones básicas:

- Obtener información sobre las preferencias e intereses de los usuarios.
- Filtrar la información o artículos que son más relevantes para los usuarios, de acuerdo a sus preferencias.
- Integrar la información de diferentes bases de datos.

2.1.2. Origen de los sistemas de recomendación

Tienen su origen a comienzo de los años 90, y eran conocidos como filtros colaborativos, término dado en 1992 para un sistema de correo electrónico no automatizado. Ya en 1994 se notó la posible utilidad y aplicación en diferentes áreas [3]. Los sistemas recomendadores, como tal, tienen su origen en el comercio electrónico [4], ya que resulta una ventaja considerable para ambas partes; es decir, favorece al vendedor porque le permite obtener un número mayor de posibles clientes, y favorece al comprador al ofrecerle productos de interés entre una colección considerable de artículos. En otras palabras, las tiendas que venden a través de internet ofrecen una gran cantidad de

productos, por lo que a los usuarios les complica encontrar lo que buscan, entonces los sistemas de recomendación le muestran posibles ítems de interés.

2.1.3. Para qué sirven los sistemas de recomendación

La información que hoy existe en la Web es tan extensa e incierta y, en muchos casos, sin un origen claro, que hace que sea muy difícil llegar a lo que realmente se busca, es por ello que se crearon los sistemas recomendadores; éstos tienen la función de sugerir diversos elementos, que se clasificarán en:

- **Artículos:** Corresponden a elementos que representan los gustos del usuario en términos de tecnología, elementos físicos u elementos de ocio tales como música, películas, videojuegos, ropa, tecnología, etc.
- **Información:** Corresponden a elementos abstractos que representan los temas de interés para el usuario, ya sean noticias, conceptos, etc.
- **Ideas:** Corresponden a elementos que van más allá de los temas de interés para el usuario, ya que considera además pensamientos y soluciones informales, como opiniones, críticas, ordenanzas, etc.

Se utilizará la palabra ítem para estandarizar lo que se recomienda, independiente a la clasificación que pertenezca. Estos ítems se recomendarán de manera óptima, de acorde a los diferentes intereses de determinados perfiles de usuario.

2.1.4. En qué se han utilizado los sistemas de recomendación

Como ya se mencionó anteriormente, los sistemas de recomendación se usan el comercio electrónico; tomando un producto recomiendan otros de posible interés, en base a que otros usuarios compraron dicho producto. Amazon.com es pionera en este tipo de recomendaciones. Casi muy ligado al comercio en línea, están los recomendadores de música, libros, películas o videojuegos. En este tipo de servicios, desde hace algunos años, last.fm destaca; ofrece radios personalizadas para cada usuario en función de las recomendaciones que reciba. El sitio web de Nintendo recomienda videojuegos, dependiendo del perfil del usuario (intereses, comentarios, etc.). Una estrategia de negocio de estas empresas, es enviar directamente las recomendaciones con un servicio de ventas.

Otro tipo de aplicaciones son el filtrado de noticias, donde se almacenan las consultas de noticias para construir perfiles de usuarios.

Desde hace algunos años, las redes sociales han comenzado a formar parte importante en la vida de cada persona, y no están exentas de la implementación de sistemas de recomendación; Facebook sugiere amigos en base a los amigos de los amigos.

Útilmente, también se han utilizado como método de educación electrónica. Por lo general, cuando se necesita ayuda sobre un tema académico, se recurre a compañeros de estudio o profesores para que se les recomienden materiales de apoyo que puedan ser de utilidad, pero ¿qué hacer si se trata de educación a distancia o virtual?, en estos casos se utilizan los sistemas de recomendación. Mientras estos sistemas apoyen la consecución de las expectativas de los usuarios, los medios y materiales necesarios para el aprendizaje y minimizando las distracciones que provocan los objetivos de navegación, la valoración de la experiencia de la enseñanza virtual será mayor [5]. Uno de los investigadores que más importancia le ha dado a la implementación de sistemas recomendadores en los procesos de e-learning es *Zaine* [6], planteando la posibilidad de incorporar estos sistemas en actividades de aprendizaje o cursos online. En resumen, los sistemas de recomendación académicos, tienen como objetivo apoyar el aprendizaje virtual, o dentro de instituciones resulta de gran utilidad para recomendar material de ayuda, algo que resultaría muy complicado sin un sistema, debido a la cantidad enorme de información que hoy existe en la red. Respecto a esta última idea, las instituciones como universidades, pueden ofrecer estos servicios de educación no solamente a sus estudiantes, sino que a personas externas, aumentando el nivel de conocimiento académico, científico y tecnológico en la sociedad.

2.1.5. Problemas generales

Existen una serie de problemas que son de ámbito general e independiente del sistema de recomendación que se utilice [7]:

2.1.5.1. Falta de información (*Lack of data*)

Los sistemas de recomendación necesitan suficiente información para hacer recomendaciones confiables. Entonces, para que los algoritmos de recomendación funcionen, necesitan información sobre los factores que participan en las recomendaciones, ítems y usuarios.

No siempre es necesario saber sobre ambas partes, sin embargo se necesita información sobre, al menos, uno de ellos; a mayor y mejor información, más precisa será la recomendación.

Existen dos formas de obtener información, explícita e implícitamente. Muchas recomendaciones se realizan en base a evaluaciones que los usuarios hacen sobre ciertos productos, utilizando ciertas escalas de valoración [8]; por lo tanto, los usuarios introducen información explícitamente, sin embargo esta forma de obtención de información necesita un esfuerzo por parte de los usuarios, además se ha demostrado que estas evaluaciones no proporcionan estimaciones fiables respecto a las preferencias del usuario. El otro método de obtención de información es que el sistema la encuentre implícitamente; los usuarios no saben de qué manera les realizan las recomendaciones, el sistema analiza su comportamiento e infiere sus preferencias [9]; esto puede hacerse al revisar los URL a los que accede el usuario o las compras que ha hecho, Amazon emplea ambas técnicas para hacer recomendaciones.

Independiente del método utilizado para obtener las evaluaciones del usuario, explícita o implícita, la falta de información es un importante problema en esta clase de sistemas [10], y puede darse por:

- **Falta de evaluaciones:** Significa que no se alcanzó la cantidad necesaria de evaluaciones para realizar las recomendaciones de manera óptima. Recomendaciones insatisfactorias pueden deberse a la densidad de las evaluaciones del usuario.
- **Falta de información:** Puede ocurrir por tres situaciones:
 - Un nuevo usuario no ha evaluado ningún producto.
 - El sistema busca recomendar un producto nuevo que no ha sido evaluado.
 - Un nuevo sistema recomendador al que no han evaluado ninguno de sus productos iniciales.

2.1.5.2. Información cambiante (*Changing data*)

Se refiere contenidos que son más útiles durante algún tiempo, en otras palabras, los ítems de moda, éstos representan las preferencias de los usuarios en períodos muy puntuales. El problema es que estos ítems reciben muchas valoraciones muy positivas cuando están de moda. Además, cuando termina este período de preferencia, los usuarios no quieren recomendaciones de estos ítems; pero el sistema continúa surgiéndolos, ya que tienen una gran cantidad de evaluaciones

positivas. Este es un problema habitual en el área de vestimenta, ya que la ropa varía considerablemente por temporadas.

2.1.5.3. Cambio de preferencias de usuario (*Changing user preferences*)

Por lo general, los usuarios buscan recomendaciones para ellos mismos, pero eventualmente pueden buscar recomendaciones para otros usuarios de diferente perfil. Esta situación puede darse cuando un usuario busca un regalo para otra persona, con preferencias y gustos definidos.

2.1.5.4. Ítems impredecibles (*Unpredictable items*)

Existen varios ítems que son difíciles de recomendar, ya que su relación con los usuarios puede ser diversa e impredecible. Un ejemplo para esta situación es el caso de las *películas de culto*, las que generalmente, para determinados usuarios, pueden ser muy buenas o muy malas, pero no tienen un término medio.

2.1.5.5. Voto prematuro y frecuente (*Vote early and often*)

Si cualquiera puede hacer recomendaciones, los propietarios de un ítem pueden hacer recomendaciones positivas de él y recomendaciones negativas para los ítems de sus competidores [11]. En este aspecto se puede considerar como un atentado a los sistemas de recomendación, ya que se intenta alterar las recomendaciones de determinados ítems.

2.1.5.6. Complejidad computacional (*Computer complexity*)

Los sistemas de recomendación tienen mejores aciertos cuando cuentan con más información; sin embargo, a mayor cantidad de información, mayor tiempo se emplea en los cálculos de obtención de resultados. La forma más habitual de reducir estos cálculos es el proceso off-line de datos, es decir, al momento de realizar una recomendación, ya se tienen datos calculados. Otra manera de abordar el problema consiste en la creación de grupos o *clusters* de ítems similares, de tal forma que los ítems similares se agrupan y computan como si fuesen el mismo.

2.1.6. Principales enfoques

Aunque tienen un mismo objetivo, los sistemas de recomendación tienen diversos enfoques, y se clasifican de acuerdo a su funcionamiento, los más utilizados y conocidos son el enfoque *colaborativo* y *basado en contenido*. Sin embargo, estos parten de una base de datos de valoraciones que los usuarios ya hayan dado para poder hacer recomendaciones; por lo tanto, estos enfoques requieren que los usuarios ya hayan interactuado anteriormente con el sistema. Debido a esta situación, han surgido otros sistemas recomendadores que ofrecen recomendaciones de calidad sin la necesidad de grandes cantidades de información [12]. Los enfoques más considerables son cuatro:

2.1.6.1. Filtrado colaborativo

Los sistemas de recomendación basados en filtrado colaborativo se caracterizan porque no utilizan la información contenida en cada elemento a recomendar, sino sólo las valoraciones que los usuarios entregan respecto a esos ítems. Esta información es utilizada para poder relacionar posteriormente a diferentes usuarios que comparten los mismos gustos.

Las valoraciones entregadas al sistema son compartidas permitiendo a otros usuarios tomar sus propias decisiones sobre qué ítems elegir, en otras palabras, cada usuario recomienda implícitamente qué elementos son recomendables a cada usuario y al mismo tiempo el sistema le recomienda ítems relacionados a sus preferencias.

Este enfoque, como ya se ha mencionado, trabaja tanto con los datos de ítems como con los datos de usuarios para hacer las predicciones, en primer lugar se establecen qué usuarios son similares según sus características, a los cuales se les denominará *Vecinos Cercanos*, para luego combinar sus preferencias y generar una lista con los productos más recomendables para los usuarios. Existen dos clasificaciones para este enfoque:

- **Basados en memoria:** Son algoritmos que realizan la recomendación en base a toda la información disponible en el sistema.
- **Basado en modelo:** Son algoritmos que utilizan las valoraciones entregadas por el usuario para crear un “modelo”, el cual será utilizado para crear predicciones de valoraciones.

Características de sus algoritmos

Los diferentes algoritmos que se utilizan en los sistemas de filtrado colaborativo basado en memoria, tienen el objetivo de calcular el peso que representa ya sea la distancia, igualdad o relación del usuario con los diferentes usuarios del sistema.

Problemas comunes

Por la naturaleza de este enfoque, se pueden encontrar variados problemas al momento de implementarlo:

- **Escalabilidad:** A medida que aumenta el número de usuarios e ítems la complejidad del sistema será mayor.
- **El problema del nuevo artículo:** Este problema se presenta al existir un nuevo elemento que no ha sido valorado aún, lo cual provoca que no pueda ser recomendado a ningún usuario, y la única forma de llegar a él es por búsqueda simple, y probablemente descartado al no tener valoraciones en las cuales apoyarse.
- **Escasez de información:** Al existir pocos usuarios en el sistema, menor será la probabilidad de encontrar elementos recomendables ya que será menor la relación entre usuarios según sus intereses.

2.1.6.2. Filtrado basado en contenido

El sistema basado en contenido hace las recomendaciones en base a la información que ha entregado previamente el usuario ya sea de forma explícita o implícita. De esta forma a un usuario sólo se le recomendarán elementos que se relacionen a sus elecciones anteriores.

El sistema puede obtener información referente al usuario de forma explícita ya sea por formularios, o implícita mediante las preferencias que ha entregado al sistema, de esta forma el sistema tendrá conocimiento de qué elementos se repiten más en las preferencias del usuario y así recomendarle otros con características similares.

Para recomendar elementos relacionados al usuario, el sistema debe almacenar además las características de los ítems para así poder realizar las comparaciones entre ellos y luego generar la recomendación en base a los elementos con mayor similitud.

Características de sus algoritmos

Los algoritmos utilizados en los sistemas de recomendación basados en el contenido, tienen como objetivo procesar la información obtenida por el sistema para luego extraer las recomendaciones. Este algoritmo crea modelos de usuarios en base a sus intereses del usuario, y en base a estos predice si un ítem es o no recomendable para él.

Problemas comunes

Los sistemas de recomendación que utilizan en enfoque basado en el contenido, deberán luchar con los siguientes inconvenientes:

- **Limitación por el análisis del contenido:** Debido a que la calidad de las recomendaciones son dependientes de la cantidad de características que posea un ítem, es necesario tener las propiedades suficientes y correctas para que nuestro sistema funcione de forma óptima.
- **Sobre-especialización:** A un usuario sólo se le harán recomendaciones de elementos relacionados con los que ya valoró, por lo cual nunca se le recomendarán ítems relacionados con otros dominios que el usuario no ha visitado.
- **Usuario nuevo:** El usuario debe haber valorado algunos elementos del sistema para que éste pueda definir sus gustos, o haber entregado información suficiente para inferirla.

2.1.6.3. Filtrado basado en conocimiento

Los sistemas recomendadores basados en conocimiento realizan sus recomendaciones en base a inferencias de las necesidades y preferencias del usuario. Para lograr esto el sistema debe analizar cómo un ítem específico responde a una necesidad en particular de un usuario y, por lo tanto, la razón sobre la relación entre la necesidad y una posible recomendación.

Estos sistemas de recomendación a diferencia de los basados en contenido y de filtrado colaborativo, profundizan en el conocimiento sobre los usuarios e ítems para lograr recomendaciones más adecuadas a los requisitos del usuario, por esto se recomienda para sistemas con dominios complejos.

Estos sistemas consideran otros aspectos como la satisfacción del usuario y el porqué de su elección, logrando hacer una representación explícita sobre el ítem y de las oportunidades que lo envuelven, lo que permite una mayor calidad, confianza y soporte para los usuarios.

Características de sus algoritmos

Los algoritmos utilizados en los sistemas basados en conocimiento están fundados en el razonamiento basado en casos, es decir, en el proceso de solucionar nuevos problemas basándose en soluciones de problemas anteriores.

Problemas comunes

Debido a la naturaleza y complejidad de este enfoque es necesaria una ingeniería del conocimiento.

Las recomendaciones, al no ser basadas en las valoraciones que va entregando el usuario y no se alimenta de nueva información que se ingresa al sistema, sólo serán del tipo estático, ya que para las mismas necesidades encontrarán las mismas recomendaciones.

2.1.6.4. Híbrido

Este enfoque corresponde a la combinación de diferentes métodos de recomendación para hacer las predicciones, con el fin de generar recomendaciones más óptimas y eliminar problemas específicos de un enfoque en particular. El enfoque híbrido más comúnmente usado es la combinación del enfoque de filtrado colaborativo con algún otro enfoque, para así paliar con el problema del nuevo artículo. Existen siete formas diferentes en que se pueden combinar los sistemas de recomendación híbridos:

- Por pesos (*weighted*).
- Conmutados (*switching*).
- Mezclados (*mixed*).
- Combinación de propiedades (*feature combination*).
- En cascada (*cascade*).
- Meta niveles (*meta-level*).

Cada uno de los enfoques antes descritos poseen diversos problemas a la hora de implementarlos de forma aislada, al combinarlos es posible reducir el impacto o eliminar completamente estos problemas. Por ejemplo, el enfoque basado en contenido o el filtrado colaborativo poseen el problema del nuevo artículo, el cual puede ser eliminado haciendo uso de recomendaciones basadas en el conocimiento. Por ende la solución a este problema consistiría en realizar las primeras recomendaciones utilizando el enfoque basado en el conocimiento y a medida que se va obteniendo información, comenzar a aplicar, ya sea, el enfoque colaborativo o el basado en contenido.

Problemas comunes

Si bien la combinación de diferentes enfoques posee varias ventajas a la hora de implementar un sistema, tampoco está exento de inconvenientes. La implementación de los sistemas basados en este enfoque posee una complejidad mayor, y es necesaria la construcción de dos o más sistemas de recomendación los cuales deben ser interoperables.

2.1.7. Principales técnicas

Los métodos o técnicas más utilizadas en la actualidad son la *minería de datos*, las *lógicas difusas* y los de *agrupamiento genético*.

2.1.7.1. Minería de datos (*Data Mining*)

La minería de datos es una técnica de extracción de información implícita en una gran cantidad de datos para descubrir conocimiento, ya sean patrones ocultos, asociaciones o anomalías; es un campo interdisciplinar que busca predecir salidas y revelar relaciones entre los datos. La minería de datos se divide en tareas:

- **Clasificación:** Cataloga los datos en alguna clase categórica ya definida.
- **Regresión:** Este modelo corresponde a un dato con un valor real de una variable.
- **Clustering o agrupación:** Agrupa los registros similares; formando un cluster o grupo, que es un conjunto de registros similares entre sí y diferente a los registros de otro cluster.
- **Generación de reglas:** Se crean reglas de datos; refieren al descubrimiento de relaciones de asociación y dependencias funcionales entre atributos.

- **Resumen o sumariazión:** Proporciona una descripción breve y precisa de subconjuntos de datos.
- **Análisis de secuencias:** Se modelan patrones secuenciales, con el objetivo de modelar los estados del proceso, u obtener la desviación y tendencias en el tiempo.

2.1.7.2. Lógica difusa (*Fuzzy Logic*)

El concepto nace en 1965 en la *Universidad de California en Berkeley*, introducido por *Zadeh* [13]. Las lógicas difusas pretenden crear aproximaciones matemáticas en la resolución de ciertos tipos de problemas, a diferencia de la lógica clásica, que únicamente imponen a sus enunciados valores verdadero o falso; si bien una gran parte del razonamiento natural ha sido satisfecho por los valores de verdad, en razonamiento humano no utiliza necesariamente valores tan deterministas.

Se denominan difusas, porque los valores de verdad no-deterministas que poseen son por lo general connotaciones de incertidumbre, lo que no implica que los valores sean necesariamente desconocidos.

Las lógicas difusas han tenido relevantes aplicaciones en el procesamiento electrónico de datos. A los enunciados de ciertas áreas de conocimiento se les asigna un valor de verdad, que son grados de veracidad o falsedad. En un sistema recomendador, el objetivo es describir los grados de los enunciados de salida en términos de entrada, de manera de optimizar la generación de recomendaciones [14].

Las lógicas difusas son una tecnología importante en el área de la *Inteligencia Artificial*, y han originado sistemas expertos de tipo difuso y sistemas de control automático.

2.1.7.3. Agrupamiento genético

Utilizan funciones de orden aleatorio y otras funciones matemáticas para modelar procesos científicos, técnicos y empresariales. Resultan de gran utilidad en situaciones que pueden tener muchas soluciones, las que deben evaluarse para generar una solución óptima. Los algoritmos genéticos utilizan reglas de procesos matemáticos. Por lo general, el uso de estos algoritmos está dirigido a la búsqueda, optimización y la descripción.

Para un enfoque más práctico, se explicará a continuación un tipo de algoritmo genético en particular.

El agrupamiento genético para datos de formas no esféricas es un algoritmo creado por *Lin Yu Tseng* y *Shiueng Bien Yang*, en 1990; encuentra automáticamente la cantidad óptima de grupos y asigna los datos a ellos de manera simultánea, también puede generar grupos de formas no esféricas [15]. Se divide en dos etapas [16]:

Primera etapa: *Obtención de clusters*

- Calcular la matriz M , de tamaño $n \times n$, de distancia para los datos de un conjunto S ; donde n es la cantidad de sesiones. Para cada sesión se encuentra la distancia con su vecino más cercano, o sea la menor distancia calculada con otras sesiones.
- Calcular el promedio de las menores distancias encontradas.
- Construir la matriz de adyacencia A .
- Encontrar los componentes conectados de la matriz, cada componente se toma como un cluster.

Segunda etapa: *Aplicación del algoritmo genético*

- **Inicialización:** Se genera una población de N cadenas, de forma aleatoria. El tamaño de las cadenas será igual a la cantidad de grupos creados anteriormente, cada una representa un subconjunto de grupos. Luego se calculan las distancias tanto de cercanía como de separación entre clusters.
- **Reproducción:** Se calcula la función objetivo para cada cadena.
- **Cruce:** Se seleccionan dos puntos aleatoriamente, desde 1 hasta N . Para cada cadena se selecciona un padre, con él se intercambiarán los valores presentes entre los dos puntos.
- **Mutación:** Se selecciona de forma aleatoria un número de 1 hasta N , el que indica a la posición de cadena que le cambiarán su valor; esto es para cada cadena. Si el valor es 0, se le asigna un 1 o viceversa. Se define un valor de probabilidad de 0.1 para la asignación de 1s.
- **Unión de grupos:** Este paso se aplica después del cálculo de la función objetivo para cada cadena. Se ordenan las cadenas en orden decreciente de la función objetivo. La cadena con mayor valor se selecciona con el objetivo de unir los grupos que contiene. Posteriormente se toma la siguiente cadena y se verifica que ninguno de los grupos que en ella se encuentran, hayan sido unidos en el paso anterior; si un grupo ya fue unido, se descarta la cadena, de lo contrario se unen los grupos. Esta etapa termina cuando se recorre toda la lista.

La generación de recomendaciones comienza cuando el usuario entra al sistema, de manera automática se crea una sesión de usuario donde se guardan los elementos a los que accede el usuario. Cuando esto ocurre, se calcula la similitud entre la sesión y los centros de los grupos generados, con el fin de establecer qué grupo define mejor el comportamiento del usuario. Mientras mayor sea la cantidad de elementos accedidos, la vinculación del perfil será más precisa. Cuando el grupo se determina, se calculan los elementos más frecuentes [17]. Las recomendaciones que recibe el usuario son las primeras N páginas más visitadas.

2.2. Herramientas de desarrollo

El lenguaje de programación con el cual es creado un sistema es de gran importancia al momento de implementar las funciones necesarias para cumplir los requerimientos del usuario. Es por esto que se debe evaluar cada una de las características de los lenguajes posibles a utilizar y definir cuál es el que cumple mejor y más eficazmente los requerimientos. Además de ser agradable e intuitivo para el desarrollador.

Se debe considerar que el lenguaje a utilizar debe soportar gran cantidad de datos, responder rápidamente a las consultas realizadas por el usuario, y pueda ser accesible fácilmente por los usuarios a los cuáles está dirigido, por ello se utilizará el sistema gestor de bases de datos SQL Server, por lo que el lenguaje a escoger debe ser compatible.

Actualmente existen muchos lenguajes de programación, y se siguen desarrollando otros nuevos, pero se estima conveniente, enfocarse en aquellos de mayor demanda por los desarrolladores, los que se exponen a continuación.

2.2.1. PHP

PHP es un lenguaje de Script creado en el año 1995 por *Rasmus Lerdorf*, es decir, que acompaña un documento HTML o está contenido en su interior; fue diseñado para incrementar el dinamismo en las páginas web. Desde sus inicios hasta ahora sus características han ido aumentando hasta convertirse en un lenguaje de programación completo capaz de manejar entornos que albergan grandes bases de datos.

Este lenguaje es de uso general de código del lado del servidor, es decir, le permite a las páginas que incluyan PHP ser interpretadas en el servidor antes de ser enviada, para finalmente enviar la página que incluye sólo código HTML al usuario que la consulta.

Ventajas:

- Este lenguaje puede desplegarse en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin ningún costo.
- Es un lenguaje que se asemeja bastante a los lenguajes de programación estructurada C y Perl, lo que le permite a los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta.
- Es capaz de conectarse con la mayoría de los gestores de bases de datos existentes en la actualidad.

Desventajas:

- Debido a que este lenguaje se interpreta en tiempo de ejecución, puede ser una complicación el hecho de que en algunos casos el código fuente no se pueda ocultar.
- Al mezclar sentencias HTML y PHP, se puede ver perjudicada la lectura y comprensión del código.
- Todas las tareas se realizan en el servidor, por lo que a medida que aumenten las solicitudes la aplicación puede resultar ineficiente.
- Dificulta la modularización de la aplicación.

2.2.2. Java

El lenguaje de programación Java fue desarrollado en el año 1995 por Sun Microsystems, está orientado a objetos y se considera de propósito general.

Java es considerado único debido a sus características, entre ellas, la más importante, es que los programas ejecutables java semi-compilados, son independientes de la arquitectura, lo que permite que puedan ejecutarse en gran variedad de equipos y sistemas operativos con diferentes características. Esto gracias a que funciona con *JVM*, la máquina virtual de Java que se encuentra disponible en diversos sistemas operativos, ésta máquina interpreta el programa Java al lenguaje de computadora, para así poder ejecutar el programa.

Para todo aquel que conozca la programación orientada a objetos, el cambio a este lenguaje puede resultar sencillo, al igual que para los programadores de C++, ya que la sintaxis es similar entre ambos lenguajes.

Ventajas:

- Aprovecha características de la mayoría de los lenguajes modernos evitando sus inconvenientes.
- Sus librerías le permiten tener gran cantidad de funciones.
- Permite crear Applets, los cuales son programas de menor tamaño que se insertan en páginas HTML y ejecutarlos en el ordenador local.
- Es fácil de aprender y está bien estructurado.
- Ofrece una mejor seguridad al controlar el acceso a recursos del sistema, y ofrece, según Sun, una seguridad garantizada frente a virus a través de redes locales e Internet.

Desventajas:

- Puede resultar un programa de ejecución lenta debido al uso de la JVM.
- Algunas herramientas tienen un costo adicional.
- Para usuarios sin conocimientos en programación podría resultar difícil de aprender.

2.2.3. ASP.NET

ASP.NET es un framework de aplicaciones web desarrollado por Microsoft. Está construido sobre *CLR* (Common Language Runtime), el cual es un entorno de ejecución para los códigos de los programas soportados por la plataforma .NET. Entre algunos de los lenguajes desarrollados para este marco encontramos: *C#, Visual Basic .Net, Delphi, C++, F#, J#, Perl, Python*, entre otros.

ASP.NET está orientado a objetos, además posee controles que permiten un estilo de HTML, como de la programación declarativa que le permite crear grandes páginas con mucho menos código que con ASP clásico.

Este framework no es solamente una actualización de ASP, ya que combina la productividad del desarrollador con un rendimiento, fiabilidad e implementación rediseñando todo el proceso.

Ventajas:

- Esta framework ofrece más de 4500 clases que encapsulan gran cantidad de funciones.
- Es mucho más rápido, ya que posee compilación de ejecución, detectando automáticamente los cambios, compilar de forma dinámica los archivos si es necesario, y almacenar los resultados compilados para la reutilización de las solicitudes futuras.
- Es liviano, y se puede utilizar en cualquier computadora que tenga acceso a la red.
- Es muy fácil de programar y tiene herramientas que permiten un manejo rápido y cómodo de la base de datos.
- Soporta aplicaciones con componentes en múltiples lenguajes lo que permite que pueda ser manejado por desarrolladores de distintos perfiles.

Desventajas:

- No es multiplataforma, ya que sólo está disponible para Windows.
- Al no ser multiplataforma limita la portabilidad.

Al analizar estas tres opciones, y considerando las características y requisitos del sistema a desarrollar, se establece que el sistema se utilizará haciendo uso del Framework .NET, debido a sus múltiples ventajas y gracias a la posibilidad de trabajar directamente con SQL Server desde Visual Studio sin necesidad de tener ambas herramientas abiertas, permitiendo reducir el tiempo de trabajo y asegurando un correcto manejo de los datos, además el utilizar el componente LINQ de la plataforma .NET permite una mayor rapidez en las consultas a la base de datos y facilita la conexión a la misma. Por otra parte al ser multilenguaje permite escoger el lenguaje que se acomode al desarrollador, por lo que se optó por el lenguaje C#. Por esto a continuación se explicará con mayor detalle este marco de trabajo.

2.2.3.1. Elementos de la estructura .NET

Esta estructura está compuesta por tres elementos fundamentales:

- Entorno común de ejecución.
- Bibliotecas de clases .NET.
- Componentes de unificación.

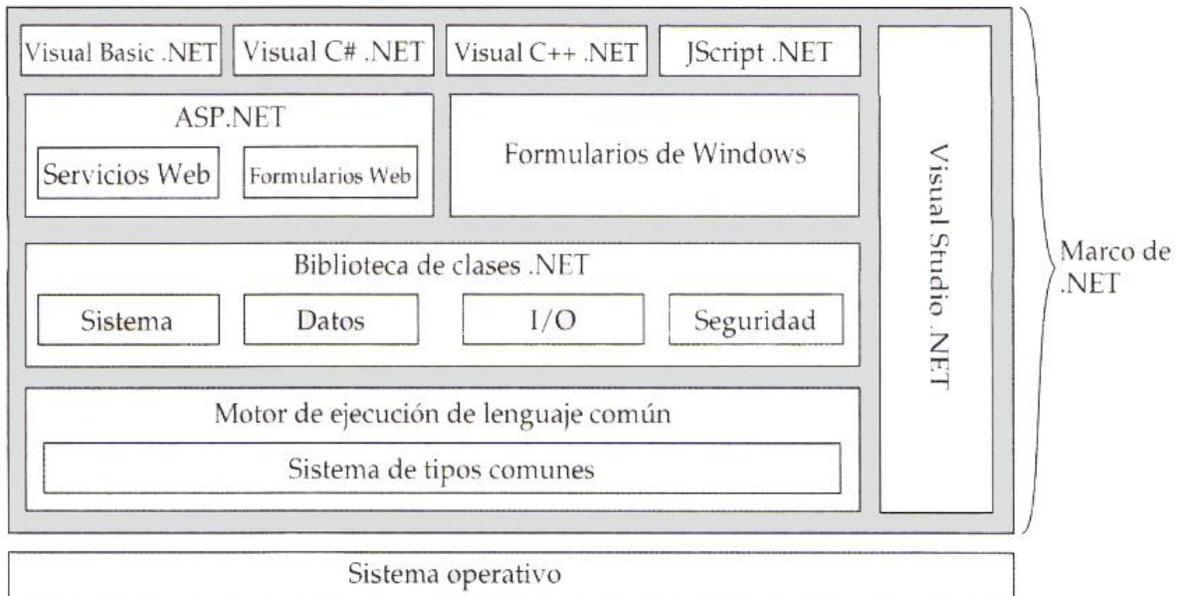


Ilustración 1: Estructura del framework .NET

Entorno común de ejecución

El entorno común de ejecución o CLR definido anteriormente, es el nivel situado entre una aplicación y el sistema operativo en el que se ejecuta. Este entorno ofrece una serie de servicios que permiten optimizar las aplicaciones, reduciendo la cantidad de código y gestionando la memoria entre otras acciones.

Este entorno es el encargado de compilar el código antes de ejecutarlo, independiente del lenguaje de programación utilizado, generando una representación en un lenguaje común a la estructura .NET conocido como *MSIL (Lenguaje Intermedio de Microsoft)* o simplemente IL.

CLR posee un sistema de tipos común (CTS), este componente ofrece un conjunto de tipos de datos con un sistema de comportamiento común a todos ellos, eliminando los problemas de compatibilidad que pueden generarse fuera de .NET.

Biblioteca de clases .NET

.NET contiene gran variedad de clases que permiten modelar el sistema y los servicios que ofrece. Para un mejor manejo, esta biblioteca se divide en espacios de nombre, donde la raíz de la biblioteca se denomina System, y contiene clases principales, tipos de datos, entre otras características.

Entre las ventajas que proporciona esta biblioteca, se puede mencionar el consistente conjunto de servicios disponibles para todos los lenguajes .NET.

2.2.3.2. Formularios Web

Los formularios Web simplifican el desarrollo del sistema, permitiendo que el desarrollador diseñe fácilmente la interfaz de un sistema, pudiendo incluso arrastrar y soltar controles para editar una página. La información puede ser validada tanto del lado del servidor como del lado del cliente.

2.2.3.3. Servicios Web

Estos servicios facilitan el diseño de aplicaciones que integran elementos de recursos remotos, consisten en una aplicación que puede ser utilizada por otras aplicaciones y componentes. Para esto es necesario invocar el método del servicio Web como si se tratase de un método de un componente instalado de un sistema.

3. DEFINICIÓN DEL PROYECTO

3.1. Objetivos del proyecto

Objetivo General

1. Desarrollar un software que permita recomendar asignaturas electivas a alumnos, aplicando algoritmos de recomendación considerando el rendimiento académico previo y preferencias de los estudiantes.

Objetivos Específicos

1. Realizar un estudio sobre los sistemas de recomendación, diferentes aplicaciones, algoritmos más utilizados, con el fin de definir los enfoques y técnicas más apropiadas para el desarrollo de un sistema recomendador.
2. Diseñar algoritmos que resuelvan específicamente la problemática identificada.
3. Implementar los algoritmos de recomendación diseñados y los diferentes módulos del sistema recomendador para sugerir asignaturas electivas a los estudiantes de Ingeniería Civil en Informática.
4. Validar los resultados entregados por el sistema, ya implementados y en marcha, mediante la utilización del sistema para recomendar electivos.
5. Desarrollar la documentación asociada al proyecto (Informe del Proyecto, manuales de usuario, otros).

3.2. Ambiente de Ingeniería de Software

Metodología de desarrollo: Modelo Cascada con reducción de riesgos

Este modelo corresponde a una modificación del modelo cascada, consistente en la incorporación de un espiral en la fase del desarrollo del proyecto que se considere con mayor riesgo. Generalmente los errores se producen a nivel de análisis de requerimientos por lo que se ubica mayormente el espiral en la parte superior del modelo. Se caracteriza por utilizar gran cantidad de documentación que es traspasada de etapa en etapa, en donde el documento obtenido en una de ellas sirve de base para el desarrollo de la etapa posterior. Este modelo está compuesto por

seis fases, en algunos casos se consideran cinco pero varía dependiendo del tipo de software que se está desarrollando. Cada una de estas etapas se realizan secuencialmente sin opción de volver a una etapa anterior, salvo en aquellas fases en donde se encuentre ubicado el espiral.

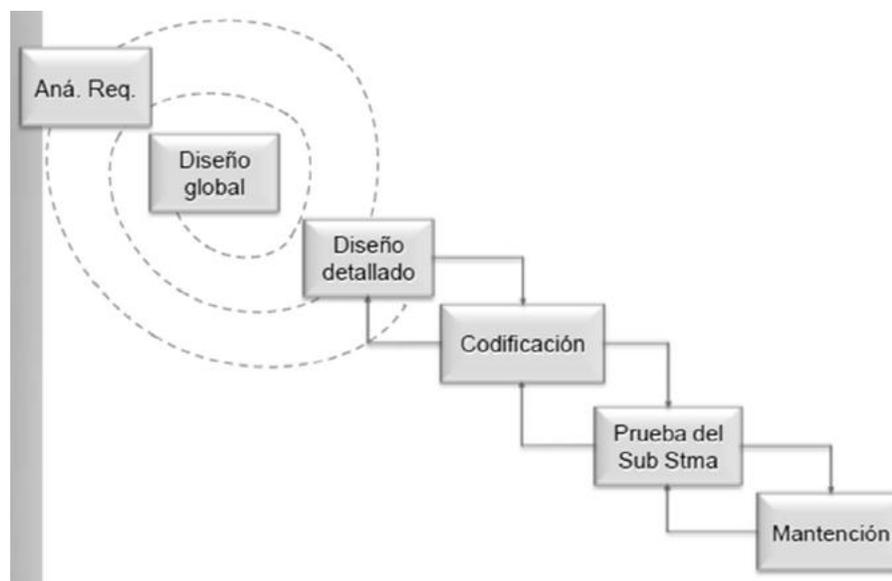


Ilustración 2: Representación gráfica del Modelo Cascada con reducción de riesgos

Etapas:

1. **Análisis y definición de requerimientos:** En esta fase se reúnen todas las necesidades y funcionalidades que el cliente desea implementar en el sistema a desarrollar. En esta fase se obtiene la definición de requerimientos del software, que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.
2. **Diseño global:** En esta fase se modela el sistema de forma tal que soporte todos los requisitos funcionales y no funcionales identificados en etapas anteriores. En esta etapa podemos obtener la descripción de los casos de uso.
3. **Diseño detallado:** En esta fase se desarrolla el sistema tras el análisis de las características del ambiente en donde se implantará, y se completa el sistema con los modelos de control, interfaz o comunicaciones, según corresponda. En esta etapa podemos obtener documentos como diagrama de clases, diagramas de secuencia, modelo relacional de la base de datos, modelo físico de la base de datos entre otros.
4. **Codificación:** En esta fase se implementa el sistema, codificándolo en un lenguaje de programación definido por el cliente.

5. **Prueba del sistema:** En esta etapa se realizan las evaluaciones pertinentes con el fin de encontrar la mayor cantidad de errores posibles para así corregirlos, esta prueba puede ser realizada por el equipo de desarrollo, o pruebas con el usuario, para así hacer la entrega de un producto de calidad.
6. **Mantención:** En esta etapa se examinan aquellos aspectos que se deberán examinar a futuro, las modificaciones y auditorías periódicas que se deberán llevar a cabo para asegura el correcto funcionamiento del sistema.

Ventajas:

- Es un modelo sencillo y fácil de llevar a cabo, por lo que facilita la planificación.
- Reduce la probabilidad de errores.
- Genera gran documentación en sus etapas por lo que se mantiene un mayor control en el proceso de desarrollo.
- Se obtiene un producto de gran calidad.

Desventajas:

- El cliente sólo valida sus requerimientos al momento de obtener el producto final, las entradas y salidas intermedias son documentos internos que no están pensadas para la validación del usuario.

Aplicabilidad:

- Proyectos de mediano tamaño, donde se tenga una noción sobre los requerimientos a implementar.
- Proyectos que utilicen tecnología nueva, que requieren mayor tiempo en el desarrollo del sistema.

Se escoge este método, debido que la aplicabilidad de éste concuerda con las características del proyecto, permitiendo utilizar un espiral para estudiar de mejor manera las tecnologías de los sistemas de recomendación.

Herramientas de apoyo al desarrollo de software a utilizar

Entorno de Desarrollo Integrado (IDE): Microsoft Visual Studio 2010

Permite crear aplicaciones y sitios web dinámicos, así como servicios en cualquier entorno que soporte la plataforma .NET.

Tanto la interfaz como la funcionalidad del sistema han sido desarrolladas en este IDE, utilizando el framework para aplicaciones web ASP.NET y el lenguaje de programación C#.

Sistema Gestor de Base de Datos (SGBD): Microsoft SQL Server 2005

Permite crear bases de datos, basadas en el modelo relacional; sus lenguajes de consultas son T-SQL y ANSI SQL; tiene la ventaja de ser compatible con Visual Studio.

4. CARACTERÍSTICAS DEL SOFTWARE

4.1. Alcances

El software:

- Mantiene información sobre el historial académico del estudiante.
- Almacena las preferencias ingresadas por el estudiante.
- Infiere qué electivos son más convenientes para el estudiante.
- Almacena qué electivos quiere inscribir el estudiante.
- Almacena información sobre los electivos y los estudiantes registrados en ellos.

No realizará:

1. Inscribir formalmente a los estudiantes de Ingeniería Civil en Informática en un electivo en la Intranet institucional.

4.2. Objetivo del software

4.2.1. Objetivo General

1. El sistema optimizará el proceso de inscripción de electivos para la carrera de Ingeniería Civil en Informática de la Universidad del Bío-Bío, facilitando la elección de los electivos de los estudiantes.

4.2.2. Objetivos Específicos

El sistema:

1. Mantendrá al estudiante informado sobre los electivos que se impartirán en el semestre y sus características, ya que podrá revisarlos en cualquier momento durante el periodo de preinscripción de electivos.
2. Mejorará la satisfacción y desempeño del estudiante ante un electivo, ya que recibirá recomendaciones en base a sus preferencias y en base a una estimación de notas a obtener considerando su rendimiento previo.
3. Disminuirá la tasa de deserción de alumnos en un electivo, ya que se mantendrán más informados sobre ellos antes de inscribirlos.

5. BASES DE LA RECOMENDACIÓN

5.1. Diseño de la recomendación

Considerando toda esta información, la propuesta es desarrollar un sistema que recomiende electivos a los alumnos de la carrera Ingeniería Civil en Informática, que tome en cuenta tanto las preferencias de los estudiantes, como su rendimiento a lo largo de la carrera. Por lo tanto, los usuarios del sistema deben interactuar primeramente con él para ingresar sus preferencias, pero el sistema también tendrá previamente almacenada información correspondiente al informe curricular del alumno; lo que implica que la propuesta tiene un enfoque basado en contenido, ya que el sistema inferirá los electivos que un estudiante debe inscribir en base a sus prioridades, información ingresada explícitamente, y sus calificaciones.

Los problemas que se presentan generalmente al utilizar el enfoque basado en contenido (Ver página 17) son enfrentados con el uso de una encuesta de preferencias, la que debe garantizar la obtención de datos suficientes y óptimos para la recomendación. Además dicha encuesta es obligatoria para obtener recomendaciones, por lo que asegura que el usuario valorará todos los electivos y así obtener recomendaciones de calidad. Por otra parte, las notas de los alumnos estarán previamente ingresadas en la base de datos, por lo que el sistema tendrá siempre los datos necesarios para realizar una buena recomendación.

Para el desarrollo de esta propuesta se utilizarán algoritmos y métodos propios, diseñados especialmente para resolver la problemática identificada, de manera que resulte sencilla y ad hoc.

Entonces, para explicar el diseño de la recomendación, se separarán en *recomendación basada en preferencias* y *recomendación basada en calificaciones*.

5.1.1. Recomendación basada en preferencias

La base de esta recomendación radica en la siguiente premisa “*La aceptación que un estudiante tiene sobre un electivo es equivalente a su aceptación promedio por las cualidades que caracterizan dicho electivo*”. Cada electivo tiene características que resultan agradables o desagradables para diversos tipos de alumnos, como su contenido, metodología de las clases, métodos de evaluación, etc.

La idea de esta recomendación es sugerir los electivos que resulten de mayor agrado para cada estudiante, para ello se propone utilizar una lista de afirmaciones que el alumno puntuará de acuerdo a su criterio. Estas aseveraciones deben estar relacionadas con cada uno de los electivos, afectándoles de manera positiva, negativa o simplemente no afecta. Las afirmaciones deben ser diseñadas de manera objetiva, responsable e idealmente, en conjunto con los profesores que impartirán electivos durante el semestre. Entonces, para establecer las bases de la recomendación de esta propuesta, se dividirá en: la forma de obtener un valor de recomendación y la forma en que se le presentará al estudiante.

5.1.1.1. Obtención de un valor de recomendación

La propuesta considera, para esta recomendación, establecer valoraciones que reflejen la aprobación del estudiante por cada una de las afirmaciones, y en base a ellas conseguir un valor de recomendación. Dichas valoraciones se representarán con la siguiente escala de aceptación.

- Muy en desacuerdo.
- En desacuerdo.
- Indiferente.
- De acuerdo.
- Muy de acuerdo.

Definida la escala de aceptación es necesario asociar a cada una con un valor numérico; estableciéndose lo siguiente.

Aceptación	Valor
Muy en desacuerdo	-2
En desacuerdo	-1
Indiferente	0
De acuerdo	1
Muy de acuerdo	2

Tabla 1: Aceptación/Valor

Cabe destacar, que cada afirmación se relaciona con cada electivo, por lo tanto un mismo valor de aceptación para una misma afirmación puede influir neutral o positivamente en algunos electivos y negativamente en otros, por ello se estableció un multiplicador, que será 1 si la afirmación afecta positivamente, -1 si afecta negativamente o 0 si no afecta al electivo; por ejemplo, la afirmación “*No me gustan las bases de datos*”, ante la respuesta “*Muy en desacuerdo*” afecta positivamente a los electivos que utilicen bases de datos, por lo que su multiplicador debe ser **-1**.

Con estas bases se diseñó la siguiente fórmula, cuyo resultado permitirá establecer que tan conveniente es recomendar qué electivos a qué estudiante:

$$\text{Valor de recomendación}_{1j} = \sum_{i=1}^n A_i * M_{ji}$$

Donde:

j : Es el j -ésimo electivo a evaluar.

n : Es el número total de afirmaciones.

A_i : El valor de aceptación del alumno por la *afirmación* _{i} .

M_{ji} : El multiplicador correspondiente a la *afirmación* _{i} respecto al electivo j .

Nota: El valor de recomendación calculado corresponde a sólo un electivo, por lo que se debe calcular para cada uno de ellos.

El valor obtenido representa la aceptación que se espera del alumno a un electivo, el cual implicará mayor probabilidad de aceptación a mayor valor. Las recomendaciones se harán considerando este valor, en otras palabras será más recomendable el electivo donde se espera se logrará mayor satisfacción del estudiante, y por consiguiente será menos recomendable el electivo donde se espere lograr la satisfacción más baja.

5.1.1.2. Presentación al estudiante

La propuesta de esta tesis estima prudente entregar al estudiante un valor de aceptación específico, de tipo porcentual, para mejorar su comprensión. Por lo tanto se debe transformar este valor utilizando la regla de tres simple:

$$\frac{A_j}{T_j} = \frac{x_j}{1}$$

Donde:

A_j : El valor de aceptación obtenido en la fórmula *Valor de recomendación*_{1j}.

T_j : El valor de aceptación que considera todas las afirmaciones como positivas, o sea, la cantidad total de afirmaciones no neutras, para el electivo j , multiplicado por 2, el mayor valor de aceptación por afirmación.

x_j : El valor incógnito equivalente al valor de recomendación para el electivo j en forma porcentual.

Nota: El valor de recomendación calculado corresponde a sólo un electivo, por lo que se debe calcular para cada uno de ellos.

Ejemplo 1: Recomendación basada en preferencias:

El electivo estará relacionado con cada afirmación presente en la base de datos de forma positiva o negativa, para este ejemplo se consideran dos electivos y sus respectivos multiplicadores para cada afirmación.

Considerar las siguientes respuestas para cada afirmación.

Afirmación	Aceptación	Valor
Afirmación 1	De acuerdo	1
Afirmación 2	Muy de acuerdo	2
Afirmación 3	En desacuerdo	-1
Afirmación 4	Muy de acuerdo	2
Afirmación 5	De acuerdo	1

Se analizan los valores totales obtenidos para cada uno de los electivos.

	Electivo 1	Electivo 2
	Multiplicador	Multiplicador
Afirmación 1	1	-1
Afirmación 2	1	-1
Afirmación 3	-1	1
Afirmación 4	1	0
Afirmación 5	1	1

Ya con estos datos, se procede a encontrar el valor de recomendación, mediante fórmula antes vista:

$$Valor\ de\ recomendación_{1j} = \sum_{i=1}^n A_i * M_{ji}$$

Recordar que:

- j : Es el j -ésimo electivo a evaluar.
- n : Es el número total de afirmaciones.
- A_i : El valor de aceptación del alumno por la afirmación i .
- M_i : El multiplicador correspondiente a la afirmación i respecto al electivo j .

Entonces reemplazando los valores se tiene:

Electivo 1

$$\text{Valor de recomendación}_1 = (1 * 1) + (2 * 1) + (-1 * -1) + (2 * 1) + (1 * 1) = 7$$

Para este electivo se obtuvo un total de 7 puntos, este valor se debe pasar a su equivalente porcentual para ser entregado al estudiante.

$$\frac{A_j}{T_j} = \frac{x_j}{1}$$

Recordar que:

- A_j : Valor de recomendación obtenido, en este caso 7.
- T_j : El valor de aceptación que considera todas las afirmaciones no neutras, para el electivo j , como positivas, en este caso el mayor valor es 10.
- x_j : El valor incógnito equivalente al valor de recomendación para el electivo j en forma porcentual.

$$x = \frac{7}{10} = 0,7$$

Por lo que se espera un 70% de aceptación del electivo por parte del alumno.

Electivo 2

$$\text{Valor de recomendación}_2 = (1 * -1) + (2 * -1) + (-1 * 1) + (2 * 0) + (1 * 1) = -3$$

Para este electivo se obtuvieron -3 puntos, en casos donde se obtienen valores negativos se define el valor de recomendación porcentual igual a 0%.

En resumen, se concluye que, para el estudiante en cuestión, es más recomendable el Electivo 1 ya que se obtuvo un valor recomendación mayor, es decir, que el Electivo 1 posee más características similares a las que el estudiante prefiere.

5.1.2. Recomendación basada en calificaciones

La base de esta recomendación radica en la siguiente premisa “*El desempeño de un estudiante en una asignatura será similar en otra asignatura con similares contenidos*”. La malla de la carrera Ingeniería Civil en Informática de la Universidad del Bío-Bío, define dos áreas de formación: *General* y *Especialidad*. En vista que las asignaturas electivas pertenecen al área de *Especialidad*, se ignora completamente el área *General* para el desarrollo de este proyecto.

En consiguiente a lo anterior, la malla considera tres principales sub áreas para la Especialidad: *Ingeniería de Software y Base de Datos*, *Sistemas Computacionales* y *Gestión en Informática*.

La idea de esta recomendación es sugerir los electivos donde se espera que el estudiante tenga mejores calificaciones. Entonces, para establecer las bases de la recomendación de esta propuesta, se dividirá en: la forma de obtener un valor de recomendación y la forma en que se le presentará al estudiante.

5.1.2.1. Obtención de un valor de recomendación

Basándose en la nueva malla de Ingeniería Civil en Informática, que comenzará en el año 2013, se tiene la siguiente tabla de asignatura/especialidad.

Asignatura	Especialidad
Primer semestre	
Introducción a la programación	Ingeniería de Software y Base de Datos
Segundo semestre	
Programación Orientada a Objetos	Ingeniería de Software y Base de Datos
Estructuras discretas para Cs. De la Comp.	Ingeniería de Software y Base de Datos
Tercer semestre	
Estructura de datos	Ingeniería de Software y Base de Datos
Administración general	Gestión en Informática

Cuarto semestre	
Fundamentos de Cs. De la Comp.	Ingeniería de Software y Base de Datos
Gestión contable	Gestión en Informática
Quinto semestre	
Modelamiento de procesos de información	Ingeniería de Software y Base de Datos
Sexto semestre	
Análisis y diseño de algoritmos	Ingeniería de Software y Base de Datos
Base de datos	Ingeniería de Software y Base de Datos
Economía	Gestión en Informática
Séptimo semestre	
Arquitectura de computadores	Sistemas Computacionales
Optimización y programación de Base de datos	Ingeniería de Software y Base de Datos
Sistemas de información	Gestión en Informática
Gestión presupuestaria y financiera	Gestión en Informática
Gestión estratégica	Gestión en Informática
Octavo semestre	
Sistemas operativos	Sistemas Computacionales
Inteligencia artificial	Ingeniería de Software y Base de Datos
Ingeniería de software	Ingeniería de Software y Base de Datos
Legislación	Gestión en Informática
Formulación y evaluación de proyectos	Gestión en Informática
Noveno semestre	
Comunicación de datos y redes	Sistemas Computacionales
Gestión proyecto de software	Ingeniería de Software y Base de Datos
Gestión de recursos humanos	Gestión en Informática
Décimo semestre	
Seguridad informática	Ingeniería de Software y Base de Datos

Tabla 2:Asignatura/Especialidad

Se debe tener en cuenta que se considerará que cada electivo pertenece a cada especialidad en diferentes grados porcentuales.

Con estas bases se diseñó la siguiente fórmula, cuyo resultado permitirá establecer que tan conveniente es recomendar qué electivos a qué estudiante:

$$\text{Valor de recomendación}_{2j} = \sum_{i=1}^n P_{ji} \left(\frac{A_{ji} + T_i}{2} \right)$$

Donde:

j : Es el j -ésimo electivo a evaluar.

n : Es el número total de especialidades, en este caso es 3.

P_{ji} : El porcentaje de pertenencia de la *especialidad* _{i} en el electivo j .

A_{ji} : El promedio de las asignaturas de la *especialidad* _{i} , requeridas en el electivo j .

T_i : El promedio de todas las asignaturas de la *especialidad* _{i} .

Nota: El valor de recomendación calculado corresponde a sólo un electivo, por lo que se debe calcular para cada uno de ellos.

El valor obtenido será equivalente a una calificación, la cual implicará mayor probabilidad de éxito a mayor nota. Las recomendaciones se harán considerando este valor; en otras palabras será más recomendable el electivo donde se espera se obtendrá la calificación más alta, y por consiguiente será menos recomendable el electivo donde se espera obtener la calificación más baja.

La fórmula pondera de la misma forma el desempeño promedio de las asignaturas que requiere explícitamente el electivo, por especialidad, como la tendencia promedio del estudiante en todas las asignaturas de la especialidad.

5.1.2.2. Presentación al estudiante

La propuesta de esta tesis estima prudente entregar al estudiante un valor de calificación específico, de tipo porcentual, para mejorar su comprensión. Por lo tanto se debe definir el desempeño de un estudiante considerando el promedio obtenido cada asignatura; utilizando la lógica tradicional, se establecen los siguientes niveles de desempeño:

Malo: El estudiante obtiene promedio entre 1.0 y 3.9.

Regular: El estudiante obtiene promedio entre 4.0 y 5.9.

Bueno: El estudiante obtiene promedio entre 6.0 y 7.0.

Sin embargo, un alumno regular de promedio 4.0 es casi igual de malo que un estudiante de promedio 3.9; así mismo, un alumno bueno de promedio 6.0 es muy similar a un alumno regular de promedio 5.9; es por ello que se usa lógica difusa, estableciendo la siguiente escala:

Malo: El estudiante obtiene promedio entre 1.0 y 4.5, sin embargo será 100% malo si su promedio está entre 1.0 y 2.9, y será 0% malo si su promedio es 4.5.

Regular: El estudiante obtiene promedio entre 3.9 y 6.0, sin embargo será 100% regular si su promedio es 5.0 y será 0% regular si su promedio es 3.9 o 6.0.

Bueno: El estudiante obtiene promedio entre 5.5 y 7.0, sin embargo será 100% bueno si su promedio es 7.0 y 0% bueno si su promedio es 5.5.

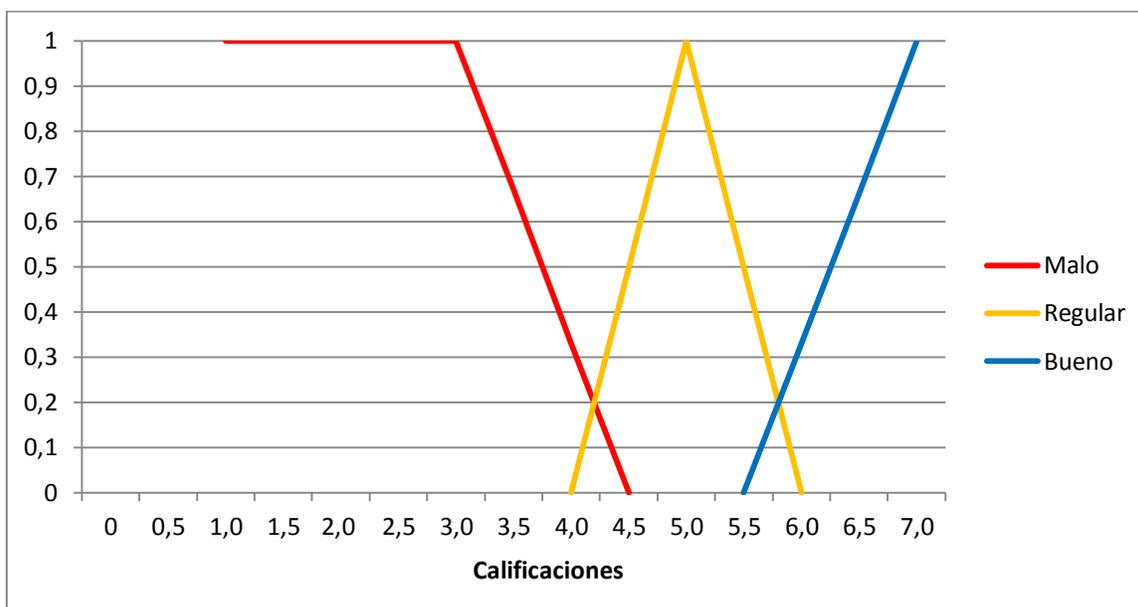


Ilustración 3: Gráfico del grado de desempeño en base a calificaciones promedio

Para identificar en qué grado el desempeño de un estudiante es malo, regular o bueno, se utilizará la ecuación de la recta:

$$y - y_1 = m(x - x_1)$$

Donde la pendiente m es:

$$m = \frac{(y - y_1)}{(x - x_1)}$$

Por lo tanto, para cada rango se tienen las siguientes coordenadas, pendientes y ecuación:

Malo

Coordenadas: (2,9; 1) (4,5; 0)

$$m = \frac{-1}{1,6}y - 1 = -0,625(x - 2,9)$$

$$y = -0,625x + 2,815$$

Regular

En este rango se observan dos pendientes, que se denotarán por Regular₁ y Regular₂.

Regular₁

Coordenadas: (3,9; 0) (5,0; 1)

$$m = \frac{1}{1,1}y - 0 = 0,909(x - 3,9)$$

$$y = 0,909x - 3,54$$

Regular₂

Coordenadas: (5,0; 1) (6,0; 0)

$$m = \frac{-1}{1}y - 1 = -1(x - 5,0)$$

$$y = 6,0 - x$$

Bueno

Coordenadas: (5,5; 0) (7,0; 1)

$$m = \frac{1}{1,5}y - 0 = 0,66(x - 5,5)$$

$$y = 0,66x - 3,63$$

Donde el valor de y, es el valor porcentual del desempeño en base a una nota x.

Nota: El valor de recomendación calculado corresponde a sólo al *j*-ésimo electivo, por lo que se debe calcular para cada uno de ellos.

Ejemplo 2: Recomendación basada en calificaciones:

Electivo 1

- Nombre: Desarrollo de aplicaciones Web utilizando ASP.NET.
- Contenido en base a Especialidad:
 - Ingeniería de Software y Base de Datos: 60%
 - Sistemas Computacionales: 40%
 - Gestión en Informática: 0%
- Asignaturas requeridas:
 - Introducción a la Programación.
 - Base de Datos.
 - Análisis y diseño de Algoritmos.

Electivo 2

- Nombre: Programación avanzada en SQL.
- Contenido en base a Especialidad:
 - Ingeniería de Software y Base de Datos: 80%
 - Sistemas Computacionales: 15%
 - Gestión en Informática: 5%
- Asignaturas requeridas:
 - Base de datos.
 - Optimización y programación avanzada de bases de datos.

Estudiante

- Notas para requisitos de Electivo 1:
 - Introducción a la Programación: 6,5
 - Base de Datos: 5,4
 - Análisis y diseño de Algoritmos: 5,9
- Tendencia promedio de notas para Electivo 1 en base a Especialidad:
 - Ingeniería de Software y Base de Datos: 5,6
 - Sistemas Computacionales: 6,0
 - Gestión en Informática: 6,5

- Notas para requisitos de Electivo 2:
 - Base de datos: 6,1
 - Optimización y programación avanzada de bases de datos: 5,4
- Tendencia promedio de notas para Electivo 2 en base a Especialidad:
 - Ingeniería de Software y Base de Datos: 6,5
 - Sistemas Computacionales: 6,3
 - Gestión en Informática: 5,9

Ya con estos datos, se procede a encontrar el valor de recomendación, mediante fórmula antes vista:

$$\text{Valor de recomendación}_{2j} = \sum_{i=1}^n P_{ji} \left(\frac{A_{ji} + T_i}{2} \right)$$

Recordar que:

- j : Es el j -ésimo electivo a evaluar.
- n : Es el número total de especialidades, en este caso es 3.
- P_{ji} : Porcentaje requerido para cada especialidad, las cuales serían Ingeniería de Software y Base de Datos, Sistemas Computacionales y Gestión en Informática, para el electivo j .
- A_{ji} : Promedio de notas de todos los requisitos del electivo j que pertenezcan a la *especialidad* _{i} , en caso de que no existan, sólo se tomará el valor de T_i .
- T_i : Nota promedio del alumno de todas las asignaturas pertenecientes a la *especialidad* _{i} , incluyendo las de A_i .

Entonces, reemplazando los valores se tiene:

Electivo 1

$P_1 = 0,6$	$P_2 = 0,4$	$P_3 = 0$
$A_1 = \frac{6,5+5,4+5,9}{3} = 5,933$	$A_2 = 0$	$A_3 = 0$
$T_1 = 5,6$	$T_2 = 6,0$	$T_3 = 6,5$

$$\text{Valor de recomendación } 1 = 0,6 * \frac{(5,933 + 5,6)}{2} + 0,4 * 6,0 + 0 * 6,5 = 5,85$$

Electivo 2

$P_1 = 0,8$	$P_2 = 0,15$	$P_3 = 0,05$
$A_1 = \frac{6,1 + 5,4}{2} = 5,75$	$A_2 = 0$	$A_3 = 0$
$T_1 = 6,5$	$T_2 = 6,3$	$T_3 = 5,9$

$$\text{Valor de recomendación 2} = 0,8 * \frac{(5,75 + 6,5)}{2} + 0,15 * 6,3 + 0,05 * 5,9 = 6,14$$

Como resultado se tiene que el Electivo 2 es más recomendable para el estudiante en cuestión, ya que se obtiene un grado mayor de recomendación, sin embargo para el Electivo 1, se obtuvo un resultado igualmente alto, por lo que resultaría ser un electivo recomendable pero en menor grado.

Finalmente, para traducir el valor de recomendación en una estimación de desempeño, se utiliza la gráfica antes vista, aplicando la función punto pendiente, dependiendo del área a la que pertenezca el resultado; en este caso el resultado es el siguiente:

Electivo 1

Como el valor de recomendación del Electivo 1 es 5,85 entra en el rango de desempeño Regular y Bueno.

Bueno

$$y = 0,66x - 3,63$$

$$y = 0,66 * 5,85 - 3,63$$

$$y = 0,23$$

Regular

$$y = 6,0 - x$$

$$y = 6,0 - 5,85$$

$$y = 0,15$$

Analizando los valores obtenidos, se observa que el estudiante, tendrá un desempeño Bueno de un 23% y un desempeño Regular de un 15%. Por lo tanto se espera que el estudiante tenga un desempeño Bueno, al tener mayor valor.

Electivo 2

Como el valor de recomendación del Electivo 2 es 6,14 entra en el rango de desempeño Bueno.

$$y = 0,66x - 3,63$$

$$y = 0,66 * 6,14 - 3,63$$

$$y = 0,4224$$

Por lo que se espera que el estudiante tenga un desempeño bueno del 42%.

En resumen, se concluye que, para el estudiante en cuestión, es más recomendable el Electivo 2 ya que se obtuvo un valor recomendación mayor, es decir, que en el Electivo 2 se espera mejor desempeño.

Nota: En base a estas condiciones, y estas fórmulas, se realizarán las recomendaciones, basadas en preferencias y en calificaciones, las cuales serán traspasadas a código.

6. ANÁLISIS

6.1. Diagrama de casos de uso

Para modelar los datos que conforman el sistema, y posteriormente traspasarlos al modelo físico, primero es necesario identificar cada uno de los actores que interactuarán en él, junto sus respectivas acciones y atributos.

6.1.1. Actores

El primer actor identificado son los estudiantes, quienes utilizarán el sistema para preinscribir los electivos a cursar durante el semestre. El estudiante también podrá obtener recomendaciones sobre qué asignaturas inscribir, al contestar una encuesta.

El segundo actor identificado son los profesores, quienes son quienes conocen las características sobre el o los electivos que cada uno impartirá durante el semestre. Los profesores serán quienes describan las características y definan los requisitos de sus respectivas asignaturas; serán quienes establezcan el grado de pertenencia de sus electivos a cada área de especialidad.

El tercer actor identificado es el jefe de carrera, quien ingresará los electivos, con sus respectivos profesores, al sistema, y podrá revisar los alumnos preinscritos en cada electivo, con el fin de inscribirlos formalmente más adelante.

Un cuarto actor identificado es el administrador, quien se ocupará de cargar la base de datos, creará usuarios, y administrará el sistema, valga la redundancia. Eventualmente, según como resulte la situación en la práctica, el administrador podría encargarse de las acciones delegadas al jefe de carrera.

6.1.2. Casos de uso y descripción

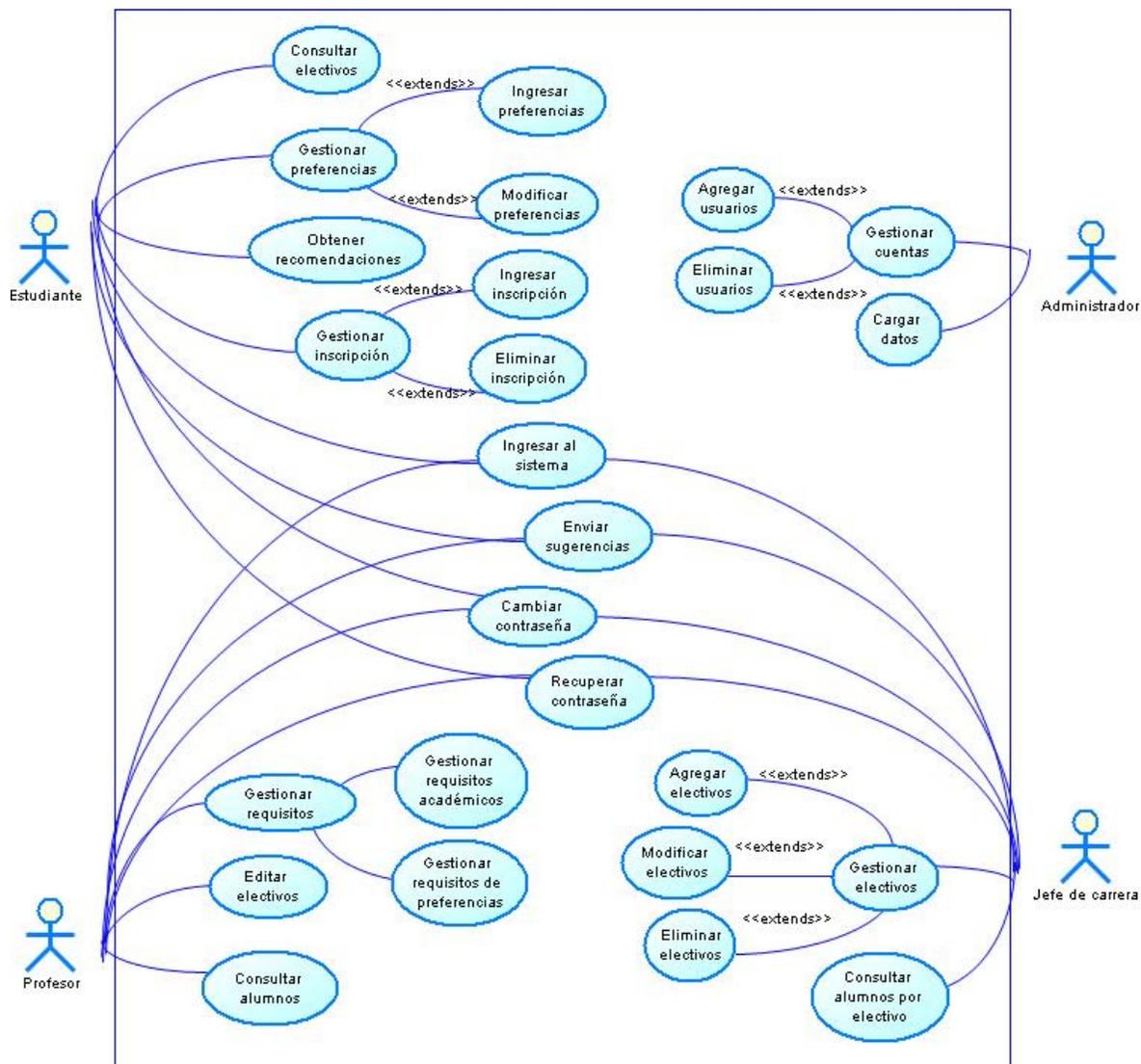


Ilustración 4: Diagrama de Casos de uso del sistema

6.1.3. Especificación de casos de uso

Caso de uso: Ingresar al sistema

- Descripción: Los tres tipos de usuario que tendrán interfaz gráfica (estudiante, profesor y jefe de carrera), deben ingresar al sistema para acceder a sus respectivas funciones.
- Pre-condiciones: Existe por lo menos un usuario de cada tipo registrado en la base de datos.
- Flujo de eventos básicos:

Estudiante/Profesor/Jefe de carrera	Sistema
1. Ingresa nombre de usuario y contraseña.	2(a). Valida y verifica nombre de usuario y contraseña. 3(a). Muestra la pantalla principal del sistema correspondiente al usuario.

- Flujo de eventos alternativos:

Estudiante/Profesor/Jefe de carrera	Sistema
	2(b). El usuario no se encuentra registrado o datos erróneos. 3(b). Muestra mensaje de error: Usuario no encontrado.

- Post-condiciones: Usuario validado en el sistema.

Caso de uso: Enviar sugerencias

- Descripción: Los tres tipos de usuario podrán enviar sugerencias o dudas al administrador.
- Pre-condiciones: El usuario está validado en el sistema.
- Flujo de eventos básicos:

Estudiante/Profesor/Jefe de carrera	Sistema
1. Ingresa asunto y mensaje.	2(a). Valida que se hayan ingresado datos. 3(a). Muestra mensaje de confirmación de envío del mensaje.

- Flujo de eventos alternativos:

Estudiante/Profesor/Jefe de carrera	Sistema
	2(b). El usuario no ha ingresado datos. 3(b). Muestra mensaje de error: Faltan datos.

- Post-condiciones: Se envía mensaje al administrador.

Caso de uso: Cambiar contraseña

- Descripción: Los tres tipos de usuario podrán modificar su contraseña cuando lo deseen.
- Pre-condiciones: El usuario está validado en el sistema.
- Flujo de eventos básicos:

Estudiante/Profesor/Jefe de carrera	Sistema
1. Ingresar contraseña actual, nueva contraseña y repite nueva contraseña.	2(a). Valida y verifica contraseñas ingresadas. 3(a). Muestra mensaje de éxito.

- Flujo de eventos alternativos:

Estudiante/Profesor/Jefe de carrera	Sistema
	2(b). Contraseña actual ingresada no coincide o contraseñas nuevas no coinciden. 3(b). Muestra mensaje de error: Contraseñas no coinciden.

- Post-condiciones: Contraseña modificada en la base de datos.

Caso de uso: Recuperar contraseña

- Descripción: Los tres tipos de usuario podrán recuperar sus datos de acceso en cualquier momento.
- Pre-condiciones: Existe por lo menos un usuario de cada tipo registrado en la base de datos.
- Flujo de eventos básicos:

Estudiante/Profesor/Jefe de carrera	Sistema
1. Ingresar correo electrónico.	2(a). Valida y verifica correo ingresado. 3(a). Muestra mensaje de confirmación de envío de mensaje.

- Flujo de eventos alternativos:

Estudiante/Profesor/Jefe de carrera	Sistema
	2(b). El usuario no se encuentra registrado o correo erróneo. 3(b). Muestra mensaje de error: Usuario no encontrado.

- Post-condiciones: Datos de accesos son enviados al usuario.

Gestionar preferencias

Caso de uso: Ingresar preferencias

- Descripción: El estudiante deberá gestionar sus preferencias de acuerdo a sus requerimientos.
- Pre-condiciones: El estudiante está validado en el sistema.
- Flujo de eventos básicos:

Estudiante	Sistema
2. Evalúa cada una de las afirmaciones presentadas.	1. Muestra formulario con las afirmaciones a evaluar por el estudiante. 3(a). Valida y verifica que se haya completado la encuesta. 4(a). Guarda las respuestas en la base de datos.

- Flujo de eventos alternativos:

Estudiante	Sistema
	3(b). El alumno no evalúa todas las afirmaciones. 4(b). Muestra mensaje de error: Debe responder toda la encuesta.

Caso de uso: Modificar preferencias

- Descripción: El estudiante puede cambiar sus preferencias ingresadas.
- Pre-condiciones: El estudiante está validado en el sistema.
- Flujo de eventos básicos:

Estudiante	Sistema
1. Evalúa cada una de las afirmaciones presentadas.	2(a). Valida y verifica que haya completado la encuesta. 3(a). Guarda las nuevas respuestas en la base de datos.

- Flujo de eventos alternativos:

Estudiante	Sistema
	2(b). El alumno no evalúa todas las afirmaciones. 3(b). Muestra mensaje de error: Debe responder toda la encuesta.

- Post-condiciones: Preferencias se actualizan en la base de datos.

Caso de uso: Obtener recomendaciones

- Descripción: El estudiante puede solicitar recomendaciones sobre qué electivos inscribir, en base a sus preferencias o calificaciones.
- Pre-condiciones: El estudiante está validado en el sistema.
- Flujo de eventos básicos:

Estudiante	Sistema
1. Consultar recomendaciones.	2(a). Verifica que se haya completado la encuesta. 3(a). Muestra la pantalla con las recomendaciones que satisfacen sus características.

- Flujo de eventos alternativos:

Estudiante	Sistema
	2(b). El alumno no ha respondido el cuestionario 3(b). Muestra mensaje de error: Debe completar el cuestionario.

- Post-condiciones: Se muestran recomendaciones de electivos al estudiante.

Gestionar inscripción

Caso de uso: Ingresar inscripción

- Descripción: El estudiante podrá realizar una preinscripción de electivos.
- Pre-condiciones: El estudiante está validado en el sistema.
- Flujo de eventos básicos:

Estudiante	Sistema
2. Selecciona electivos a preinscribir.	1. Muestra listado de electivos disponibles para realizar preinscripción. 3(a). Valida y verifica que se hayan seleccionado 1 a 4 electivos, cumpla con los requisitos y hayan cupos disponibles. 4(a). Se muestra mensaje de confirmación de inscripción.

- Flujo de eventos alternativos:

Estudiante	Sistema
	3(b). El alumno no cumple los requisitos, no quedan cupos disponibles o ya realizó el máximo de preinscripciones. 4(b). Muestra mensaje de error: No cumple requisitos.

- Post-condiciones: Electivos preinscritos son almacenados en la base de datos.

Caso de uso: Eliminar inscripción

- Descripción: El estudiante puede eliminar electivos preinscritos.
- Pre-condiciones: El estudiante está validado en el sistema.
- Flujo de eventos básicos:

Estudiante	Sistema
2. Selecciona el electivo del cual desea eliminar preinscripción.	1. Muestra listado de electivos preinscritos por el usuario. 3(a). Muestra mensaje de éxito.

- Post-condiciones: Preinscripción seleccionada se elimina de la base de datos.

Caso de uso: Editar electivos

- Descripción: El profesor puede y debe facilitar al sistema información relacionada al o los electivos que impartirá.
- Pre-condiciones: El profesor está validado en el sistema.
- Flujo de eventos básicos:

Profesor	Sistema
1. Seleccionar electivo a modificar.	3(a). Valida y verifica que se hayan ingresado datos correctos.
2. Ingresar horario en que se imparte el electivo seleccionado.	4(a). Guarda la información ingresada del electivo.

- Flujo de eventos alternativos:

Profesor	Sistema
	3(b). No se ingresan datos o se ingresan datos erróneos. 4(b). Muestra mensaje de error: Debe ingresar datos correctos.

- Post-condiciones: Información de electivos es actualizada en la base de datos.

Gestionar requisitos

Caso de uso: Gestionar requisitos académicos

- Descripción: El profesor puede y debe establecer y facilitar al sistema las asignaturas requisito para el o los electivos que impartirá en el semestre además de los grados de pertenencia a cada especialidad.
- Pre-condiciones: El profesor está validado en el sistema.
- Flujo de eventos básicos:

Profesor	Sistema
1. Seleccionar electivo a gestionar requisitos académicos.	2(a). Verifica que se haya seleccionado un electivo.
3. Seleccionar asignaturas que son requisitos para el electivo seleccionado y grados de pertenencia a cada especialidad.	4(a). Verifica que se haya seleccionado al menos una asignatura y se hayan ingresado grados de pertenencia correctos.
	5(a). Guarda requisitos académicos del electivo seleccionado.

- Flujo de eventos alternativos:

Profesor	Sistema
	2(b). No se seleccionan asignaturas u grados de pertenencia.
	4(b). Muestra mensaje de error: Debe completar datos.

- Post-condiciones: Requisitos académicos para el electivo seleccionado se guardan exitosamente.

Caso de uso: Gestionar requisitos de preferencias

- Descripción: El profesor puede y debe establecer el grado en que afecta cada afirmación a los electivos que imparte.
- Pre-condiciones: El profesor está validado en el sistema.
- Flujo de eventos básicos:

Profesor	Sistema
1. Seleccionar electivo a establecer requisitos de preferencias.	2(a). Verifica que se haya seleccionado un electivo.
3. Seleccionar grados en que afectan las afirmaciones al electivo seleccionado.	4(a). Validar y verificar grados ingresados.
	5(a). Guarda grados del electivo seleccionado.

- Flujo de eventos alternativos:

Profesor	Sistema
	2(b). No se completan todos los grados.
	4(b). Muestra mensaje de error: Debe completar los requisitos.

- Post-condiciones: Grados en que afectan las afirmaciones se guardan exitosamente.

Gestionar electivos

Caso de uso: Agregar electivos

- Descripción: El jefe de carrera puede y debe facilitar al sistema la lista de electivos que se impartirán en el semestre.
- Pre-condiciones: El jefe de carrera está validado en el sistema.
- Flujo de eventos básicos:

Jefe de carrera	Sistema
1. Ingresar datos correspondiente al electivo a ingresar.	2(a). Valida y verifica datos ingresados.
	3(a). Se guarda el electivo ingresado en la base de datos.

- Flujo de eventos alternativos:

Jefe de carrera	Sistema
	2(b). No se ingresan datos o se ingresan datos erróneos.
	3(b). Muestra mensaje de error: Datos incorrectos.

- Post-condiciones: Electivo guardado exitosamente.

Caso de uso: Modificar electivos

- Descripción: El jefe de carrera puede modificar alguno de los electivos que se impartirán en el semestre.
- Pre-condiciones: El jefe de carrera está validado en el sistema.
- Flujo de eventos básicos:

Jefe de carrera	Sistema
1. Selecciona el electivo que desea editar.	2(a). Se habilita edición de electivo seleccionado.
3. Se ingresan datos del electivo seleccionado para modificar.	4(a). Se actualizan datos del electivo en la base de datos.

- Flujo de eventos alternativos:

Jefe de carrera	Sistema
	2(b). No se ingresan datos o se ingresan datos erróneos.
	4(b). Muestra mensaje de error: Datos incorrectos.

- Post-condiciones: Datos del electivo actualizados correctamente.

Caso de uso: Eliminar electivos

- Descripción: El jefe de carrera puede y debe eliminar los electivos que ya no son requeridos.
- Pre-condiciones: El jefe de carrera está validado en el sistema.
- Flujo de eventos básicos:

Jefe de carrera	Sistema
1. Seleccionar electivo a eliminar.	2(a). Confirma el usuario la eliminación. 3(a). Se actualiza estado del electivo en la base de datos.

- Flujo de eventos alternativos:

Jefe de carrera	Sistema
	2(b). El usuario cancela la eliminación. 3(b). No se producen cambios en la base de datos.

- Post-condiciones: Se actualiza el estado del electivo en la base datos correctamente.

Caso de uso: Ver alumnos por electivo

- Descripción: El jefe de carrera y profesor pueden revisar la lista de alumnos inscritos en cada electivo accesible a ellos. El jefe de carrera podrá posteriormente matricular a cada estudiante en los electivos que tomó, en la Intranet de la Universidad del Bío-Bío (El proceso de registro de alumnos en electivos en la Intranet es externo al sistema desarrollado en este proyecto).
- Pre-condiciones: El profesor/jefe de carrera está validado en el sistema.
- Flujo de eventos básicos:

Profesor/Jefe de carrera	Sistema
1. Se selecciona electivo para consultar alumnos preinscritos.	2(a). Verifica que haya un electivo seleccionado. 3(a). Muestra al usuario la lista de alumnos preinscritos en el electivo seleccionado.

- Flujo de eventos alternativos:

Profesor/Jefe de carrera	Sistema
	2(a). No se ha seleccionado electivo. 3(a). No hay cambios.

- Post-condiciones: Se presenta lista de alumnos preinscritos en electivo seleccionado al jefe de carrera o profesor.

Gestionar cuentas

Los siguientes casos de uso corresponden a las acciones del administrador, el cuál no interactuará con el sistema en sí, sino lo administrará accediendo directamente al sistema gestor de base de datos, por lo que no tendrá interfaz gráfica.

Caso de uso: Agregar usuarios:

- Descripción: El administrador puede y debe ingresar en la base de datos del sistema los usuarios, nombre y clave, que utilizarán el sistema, es decir, estudiantes que puedan tomar electivos, y jefe de carrera.

Caso de uso: Eliminar usuarios:

- Descripción: El administrador puede y debe eliminar de la base de datos los usuarios que ya no son requeridos.

Caso de uso: Cargar datos:

- Descripción: El administrador debe ingresar en la base de datos del sistema todos los datos necesarios para funcionar.

6.2. Modelamiento de datos

Restricciones semánticas:

- Un profesor gestiona uno o más electivos, y un electivo es impartido por un profesor.
- Un electivo se relaciona con todas las afirmaciones y cada afirmación se relaciona con todos los electivos.
- A un estudiante se le recomienda uno o muchos electivos y un electivo puede ser recomendado a muchos estudiantes.
- Cada asignatura pertenece a un área, y a un área pertenecen muchas asignaturas.
- Un electivo pertenece a varias áreas, y a un área pertenecen muchos electivos.
- Un estudiante responde muchas afirmaciones y cada afirmación es respondida por muchos estudiantes.
- Un estudiante cursó muchas asignaturas y cada asignatura fue cursada por muchos estudiantes.

Identificación de entidades y atributos

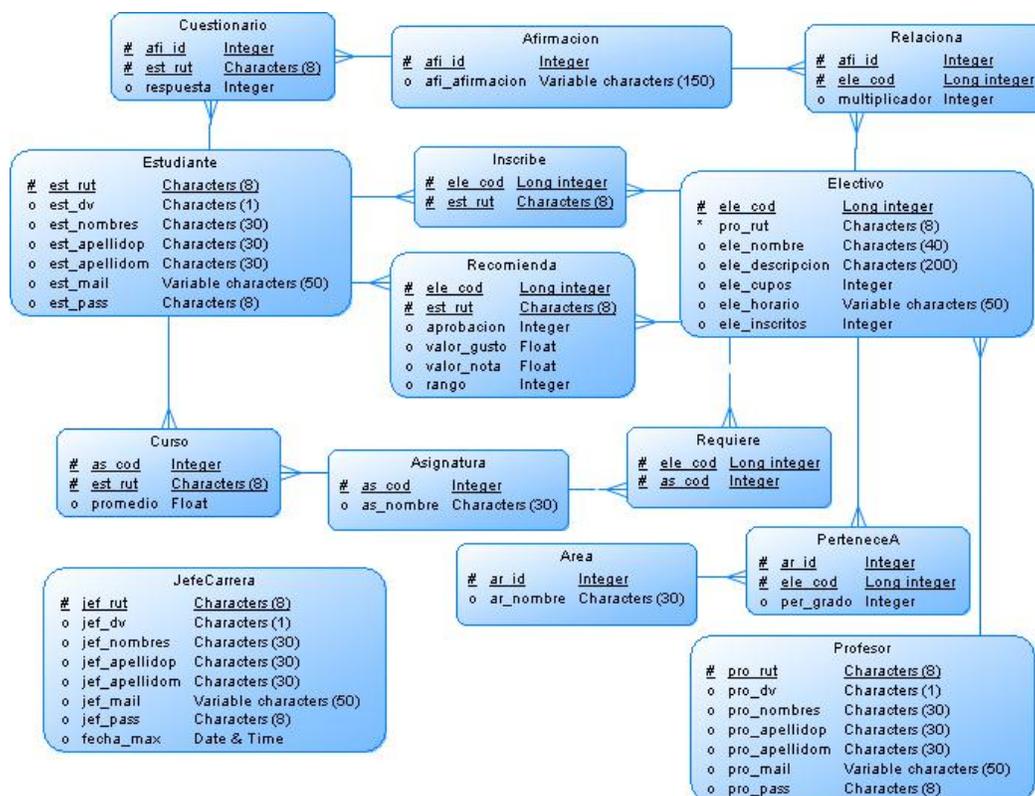


Ilustración 5: Modelo Entidad-Relación – Vista general

7. DISEÑO

7.1. Diseño físico de la base de datos

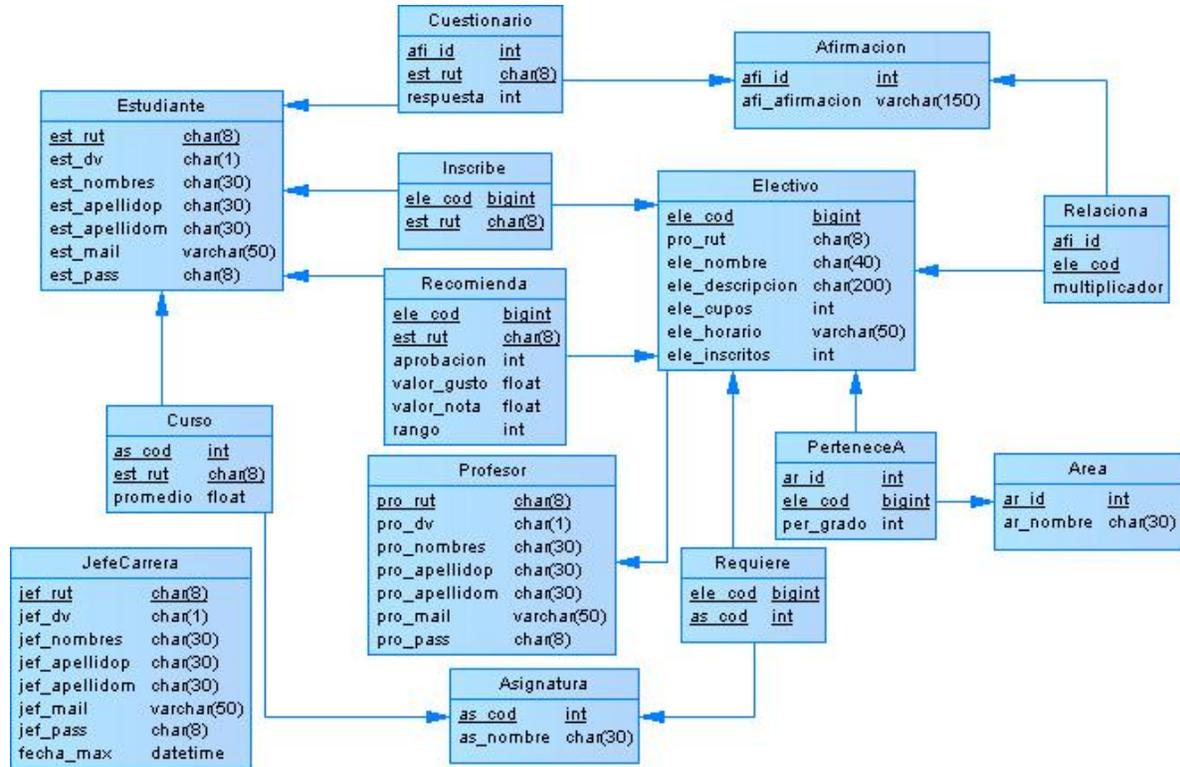


Ilustración 6: Modelo Relacional - Vista general

7.2. Diseño de arquitectura funcional

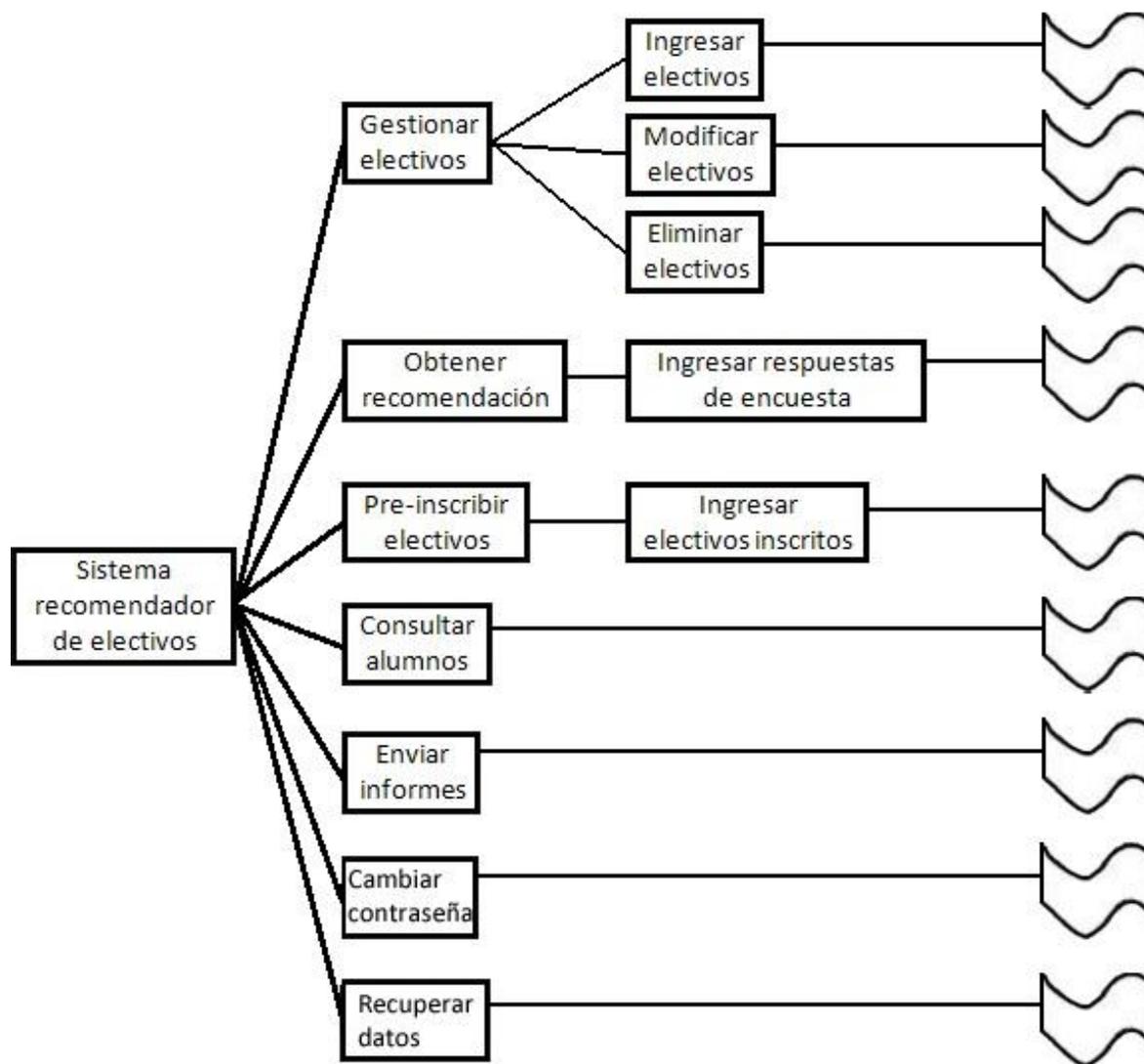


Ilustración 7: Diseño de arquitectura funcional del sistema

7.3. Diseño de interfaz y navegación

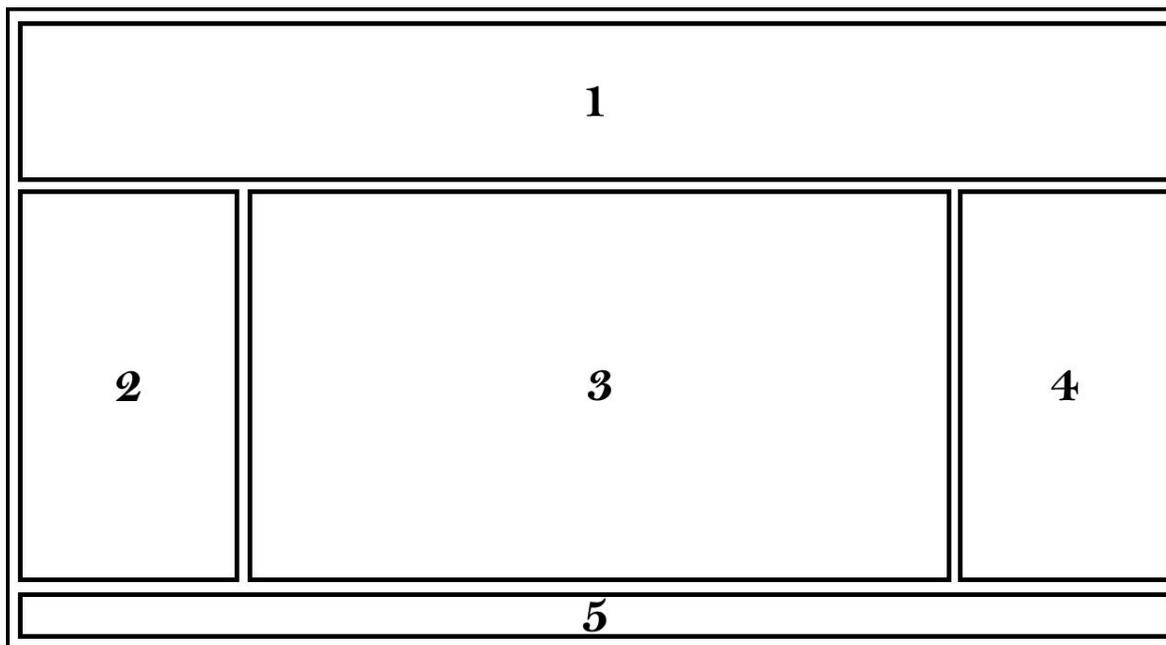


Ilustración 8: Diseño de interfaz del sistema

Área 1: Banner y Menú.

Posee un banner que representa a la universidad. Muestra un menú de opciones que variará dependiendo del usuario logueado en el sistema.

Área 2: Login, Datos del usuario y Calendario.

Contiene controles del Login para que el usuario ingrese al sistema. Letras color negro y fondo Azul.

Área 3: Contenedor principal.

En este módulo se presenta toda la información que el usuario solicita a través del menú superior. Títulos color azul, texto normal color negro y fondo blanco.

Área 4: Twitter.

Módulo que presenta los comentarios hechos por los usuarios en la cuenta de Twitter.

Área 5: Pie de página.

Módulo que muestra el pie de página del sistema. Letras color blanco y fondo azul oscuro.

8. IMPLEMENTACIÓN

8.1. Funcionamiento del sistema

A continuación se explica la interacción de cada uno de los usuarios con el sistema.



Ilustración 9: Interfaz general del sistema

Los tres tipos de usuarios tienen el mismo portal de ingreso, posteriormente serán redireccionados a su correspondiente sección. La estructura del sitio es la misma estando dentro del sistema. El *Usuario* corresponde al RUT del usuario, sin puntos, guion ni dígito verificador, además la *Contraseña*, en primera instancia es el mismo *Usuario*.

En el borde izquierdo se encuentra el área donde el usuario se identifica para entrar al sistema; puede solicitar una clave en caso que la haya olvidado. Considerando la influencia de las redes sociales, en el borde derecho se encuentra un espacio donde se pueden leer los comentarios, tanto del administrador como de los usuarios, utilizando una cuenta Twitter.

Para cada uno de los usuarios se tienen las siguientes funciones y características:

Jefe de carrera:

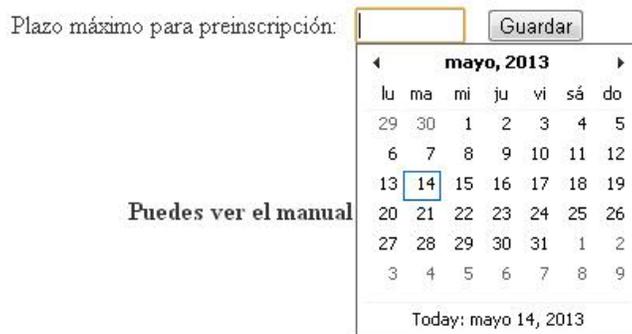


Ilustración 10: Establecer fecha máxima de preinscripción

El jefe de carrera debe establecer lo antes posible la fecha máxima para la preinscripción de electivos, los estudiantes podrán contestar el cuestionario, preinscribir y eliminar electivos hasta esta fecha.

Gestionar electivos (Gestión de Electivos)

Descripción: El jefe de carrera puede y debe agregar al sistema los electivos que se impartirán durante el semestre. También puede ver la lista de electivos registrados, modificarlos o eliminarlos.

Gestión de Electivos

Ingresar nuevo electivo ①

CODIGO	NOMBRE	CUPOS	PROFESOR		
620390	Fundamentos de programación en Java.	30	Meza		
620393	Desarrollo de aplicación usando ASP.NET	30	Perez		
620421	Taller de Programación para Dispositivos Móviles.	30	Ruiz		
620445	Administración y Manejo de GNU/Linux y software.	3	Ramirez		
620464	Programación avanzada en SQL.	30	Vidal		
620465	Desarrollo aplicación web.	30	Muñoz		

②

Ilustración 11: Gestión de electivos - Jefe de carrera

- 1: El jefe de carrera puede y debe ingresar nuevos electivos, con sus datos relacionados, al sistema.
- 2: Lista de electivos ingresado en el sistema, con la opción de Editar (ícono de lápiz) o Eliminar (ícono de cruz).

Funcionamiento

- Los datos de la lista son cargados automáticamente desde la tabla *ELECTIVOS*, mediante el uso de *LinqDataSource* de *LINQ*.
- Los cambios hechos en esta lista se efectuarán mediante un *UPDATE* en la tabla *ELECTIVOS* de la base de datos.
- La eliminación los electivos se efectuarán mediante un *UPDATE* en la tabla *ELECTIVOS* de la base de dato, dicha actualización cambiará el estado del electivo de activo a inactivo, por lo que se realizará una eliminación lógica.

Gestionar electivos (Ingresar nuevo Electivo)

Descripción: El jefe de carrera puede y debe ingresar información general relacionada al electivo que se quiera registrar.

Ingresar nuevo Electivo

Código:	<input type="text"/>	①
Nombre:	<input type="text"/>	②
Profesor:	<input type="text" value="Juan José Ramirez"/> ▼	③
Cupos:	<input type="text"/> ▲ ▼	④
<input type="button" value="Guardar"/>		⑤

Ilustración 12: Módulo de ingreso de nuevo electivo

- 1: Espacio donde el jefe de carrera ingresa el código del electivo a ingresar.
- 2: Espacio donde el jefe de carrera ingresa el nombre del electivo a ingresar.
- 3: Lista de profesores de donde el jefe de carrera escoge para asignar al electivo a ingresar.
- 4: Espacio donde el jefe de carrera ingresa la cantidad de cupos disponibles para el electivo.
- 5: Botón que permite al jefe de carrera guardar los datos ingresados.

Funcionamiento

- Los nombres de los profesores son cargados automáticamente desde la tabla *PROFESORES*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Los datos ingresados verificados y validados, para posteriormente guardarlos mediante un *INSERT* en la tabla *ELECTIVOS* de la base de datos.

Consultar alumnos

Descripción: El jefe de carrera puede revisar los alumnos registrado por cada electivo, para posteriormente registrarlos como inscritos en la Intranet de la universidad.

Listado de Alumnos por Electivo

Electivo: **1** 3 registros encontrados. **2**

RUT	NOMBRES	AP. PATERNO	AP. MATERNO
17212941-2	Teresa	Ulloa	Placencia
17538724-2	Marcelo	Torres	Ríos
15677441-3	Pablo	Leiva	Faundez

3

Ilustración 13: Lista de estudiantes pre-inscritos por electivo

- 1: Lista de electivos de donde el jefe de carrera escoge para ver su listado de alumnos inscritos.
- 2: Información sobre la cantidad de alumnos inscritos en el electivo seleccionado.
- 3: Lista de alumnos inscritos en el electivo seleccionado.

Funcionamiento

- Los nombres de los electivos son cargados automáticamente desde la tabla *ELECTIVOSS*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Los datos de la lista son cargados automáticamente desde la tabla *ESTUDIANTES*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*, donde la condición es que el estudiante esté inscrito en el electivo seleccionado.
- La cantidad de alumnos registrados por electivo se obtiene haciendo uso de la propiedad *Rows.Count* de la grilla.

Nota: El profesor también puede consultar alumnos, de la misma forma que el jefe de carrera, pero sólo con los electivos que éste imparta.

Profesor

Gestionar electivos

Descripción: El profesor puede agregar o modificar información relacionada a los electivos que impartirá en el semestre. Dicha información no es esencial para el funcionamiento de la recomendación, sin embargo resulta de utilidad para los estudiantes.

Gestión de Electivos

Lista de electivos a impartir

CODIGO	NOMBRE	HORARIO	CUPOS
620390	Fundamentos de programación en Java.	Martes 11:10 a 19:00	30

Edición de electivos

Electivo: Fundamentos de programación en Java. 2

Horario: Martes 11:10 a 19:00 3

4

Ilustración 14: Gestión d electivos - Profesor

Lista de electivos a impartir

1: Lista de todos los electivos que el profesor impartirá durante el semestre.

Edición de electivos

2: Lista de electivos de donde el profesor escoge para editar.

3: Espacio donde el profesor ingresa el horario del electivo seleccionado, o bien modificar dicha información, en caso que el horario esté establecido.

4: Botón que permite al profesor guardar los cambios efectuados en el electivo seleccionado.

Funcionamiento

- Los datos de la lista son cargados automáticamente desde la tabla *ELECTIVOS*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*, donde la condición es que el electivo lo imparta el profesor que esté utilizando el sistema.
- Los nombres de los electivos son cargados automáticamente desde la tabla *ELECTIVOSS*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Los datos ingresados verificados y validados, para posteriormente guardarlos mediante un *INSERT* en la tabla *ELECTIVOS* de la base de datos.

Gestionar requisitos

Descripción: El profesor puede y debe establecer las asignaturas requeridas para que un estudiante pueda tomar el o los electivos que impartirá en el semestre. También puede y debe establecer el grado de pertenencia a cada área de especialidad del o los electivos que impartirá en el semestre. Esta información es requerida para estimar el desempeño del estudiante en los electivos que tome.

Gestión de Requisitos Académicos

Seleccionar electivo: **1**

Asignación de porcentajes de área:

Ingeniería de Software y Bases de Datos: % **2**

Sistemas Computacionales: % **3**

Gestión Informática: % **4**

Seleccionar requisitos: **5**

Filtrar: Área Semestre **6**

Seleccionar asignaturas: **7**

- Introducción a la Programación
- Programación Orientada a Objetos
- Estructuras discretas para Cs. de la Co
- Estructuras de datos
- Administración general
- Fundamentos de Cs. de la Computación
- Gestión Contable
- Modelamiento de Procesos de Informac
- Análisis y diseño de algoritmos

9

Agregar >>

<< Quitar

10

Guardar **11**

8

Ilustración 15: Gestión de requisitos

1: Lista de electivos de donde el profesor escoge para establecer áreas y requerimientos.

Asignación de porcentajes de área

- 2: Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Ingeniería de Software y Base de Datos.
- 3: Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Sistemas Computacionales.
- 4: Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Gestión informática.

Nota: La suma de estos tres valores debe ser exactamente 100, ni más ni menos.

Requisitos

- 5: El profesor escoge si filtrar las asignaturas de la carrera por Área o por Semestre.
- 6: Si se escoge filtrar por Área, se podrán escoger las asignaturas que pertenezcan al área de especialidad seleccionada. Si se escoge filtrar por Semestre, se podrán escoger las asignaturas que pertenezcan, por malla, entre el primer y décimo semestre.
- 7: Lista de total de asignaturas, o bien las que satisfacen al filtro previamente ingresado. El profesor puede seleccionar aquellas que considere como requisito para que un estudiante curse el electivo seleccionado.
- 8: Lista de las asignaturas que el profesor considere como requisito para que un estudiante curse el electivo seleccionado.
- 9: Botón que permite al profesor agregar a la lista de requisitos un ramo de la lista de asignaturas.
- 10: Botón que permite al profesor quitar de la lista de requisitos una asignatura.
- 11: Botón que permite al profesor guardar los cambios efectuados en el electivo seleccionado.

Funcionamiento

- Los nombres de los electivos son cargados automáticamente desde la tabla *ELECTIVOS*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Los nombres de las asignaturas son cargados automáticamente desde la tabla *ASIGNATURAS*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*, teniendo la opción de filtrarlas ya sea por área o semestre.
- Los porcentajes de área ingresados son verificados y validados, para posteriormente guardarlos mediante un *INSERT* en la tabla *PERTENECE* de la base de datos.
- Las asignaturas requeridas ingresadas son verificados y validados, para posteriormente guardarlos mediante un *INSERT* en la tabla *REQUIERE* de la base de datos.

Gestionar encuesta

Descripción: El profesor puede y debe establecer si las afirmaciones de la encuesta afectan positivamente, negativamente o no afectan al o los electivos que imparta.

Gestión de Requisitos para Encuesta

Para que el sistema pueda ofrecer recomendaciones, debes establecer si cada afirmación afecta o no al electivo que impartes, por lo que te pedimos que respondas de forma objetiva y responsable, ya que no podrás modificarlo posteriormente.

Instrucciones: Marca la opción que corresponda considerando lo siguiente.

+: Afecta positivamente.

0: No afecta.

-: Afecta negativamente.

Seleccionar electivo:

Fundamentos de programación en Java. ▾

1

Nº	CUESTIONARIO	+	0	-
1	Me interesa conocer las virtudes ofrecidas por los software libres	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Tengo conocimiento sobre redes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
3	La seguridad informática no es mi fuerte	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
4	No tengo buen desempeño con las bases de datos	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5	Me es fácil encontrar soluciones óptimas	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Me es fácil modelar información	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Me interesa aprender más sobre SQL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
8	No tengo un dispositivo móvil moderno	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
9	Me interesa aprender a desarrollar aplicaciones móviles	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
10	Me gusta programar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

2

Ilustración 16: Lista de afirmaciones que influyen en un electivo

- 1: Lista de electivos de donde el profesor escoge para establecer como le afecta cada afirmación.
- 2: Lista de afirmaciones que el profesor debe evaluar como afectan al o los electivos que él imparta. Esta información es importante al momento de calcular las recomendaciones basadas en preferencias. Una vez que el profesor establezca estos requisitos no podrá volver a modificarlos.

Funcionamiento

- Los datos de la lista son cargados automáticamente desde la tabla *AFIRMACIONES*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Se verifica que se hayan puntuado todas las afirmaciones para posteriormente guardarlas mediante un *INSERT* en la tabla *RELACIONA* de la base de datos.

Estudiante

Recomendaciones (Cuestionario sin responder)

Descripción: El estudiante puede y debe, si quiere recibir recomendaciones, responder una serie de preguntas relacionadas con los electivos que se impartirán en el semestre; en base a estas respuestas el sistema obtendrá las recomendaciones óptimas para su caso.

Recomendaciones

Para que el sistema pueda ofrecer recomendaciones, debes responder la siguiente encuesta de opinión, por lo que te pedimos que respondas de forma objetiva y responsable.

Instrucciones: La pauta de evaluación debe ser completada marcando la opción que mejor describa tu apreciación en cada uno de los indicadores que se presentan.

Para ello considera la siguiente escala:

- 1: Muy de acuerdo.
- 2: De acuerdo.
- 3: Indiferente.
- 4: En desacuerdo.
- 5: Muy en desacuerdo.

1

Nº	AFIRMACIONES	1	2	3	4	5
1	Me interesa conocer las virtudes ofrecidas por los software libres	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
2	Tengo conocimiento sobre redes	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	La seguridad informática no es mi fuerte	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4	No tengo buen desempeño con las bases de datos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5	Me es fácil encontrar soluciones óptimas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
6	Me es fácil modelar información	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
7	Me interesa aprender más sobre SQL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
8	No tengo un dispositivo móvil moderno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
9	Me interesa aprender a desarrollar aplicaciones móviles	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Me gusta programar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[Ver recomendaciones](#) 2

Ilustración 17: Encuesta para establecer preferencias

- 1: Serie de preguntas que el estudiante debe puntuar con valores de 1 a 5, dependiendo de su situación. Se debe responder todas las preguntas.
- 2: Botón que permite al alumno guardar sus preferencias y ver las recomendaciones en base a ellas.

Funcionamiento

- Al cargar la página se verifica si el estudiante respondió el cuestionario, en caso de no haberlo hecho, ocurre lo siguiente.
- Los datos de la lista son cargados automáticamente desde la tabla *AFIRMACIONES*, con un *SELECT* haciendo uso de la clase *DataContext* de *LINQ*.
- Se verifica que se hayan puntuado todas las afirmaciones para posteriormente guardarlas mediante un *INSERT* en la tabla *CUESTIONARIO* de la base de datos.

Recomendaciones (Cuestionario respondido)

Descripción: En caso que el estudiante haya contestado la encuesta, podrá modificar las respuestas, o simplemente ir directo a las recomendaciones.

Recomendaciones

Ya has completado la encuesta.



Ilustración 18: Vista si la encuesta ya se respondió

- 1: Botón que permite al alumno ver inmediatamente las recomendaciones.
- 2: Botón que permite al alumno volver a contestar el cuestionario.

Funcionamiento

- Al cargar la página se verifica si el estudiante respondió el cuestionario, en caso de haberlo hecho, se le da la opción de responderlo nuevamente o ver las recomendaciones.

Recomendaciones (Lista de recomendaciones)

Descripción: Después que el estudiante conteste el cuestionario puede revisar los electivos que el sistema le sugiere, basándose tanto en sus preferencias como en su desempeño esperado.

Recomendaciones

Las siguientes recomendaciones están basadas en los resultados de la encuesta que respondiste. Sin embargo, se sugiere que consideres también el nivel de desempeño esperado.

CODIGO	NOMBRE	ACEPTACIÓN
620393	Desarrollo de aplicación usando ASP.NET	53%
620464	Programación avanzada en SQL.	41%
620465	Desarrollo aplicación web.	36%
620390	Fundamentos de programación en Java.	15%
620445	Administración y Manejo de GNU/Linux y software.	8%
620421	Taller de Programación para Dispositivos Móviles.	0%

1

Las siguientes recomendaciones están basadas en tu rendimiento académico durante todo el período académico.

CODIGO	NOMBRE	DESEMPEÑO
620464	Programación avanzada en SQL.	39%
620421	Taller de Programación para Dispositivos Móviles.	37%
620445	Administración y Manejo de GNU/Linux y software.	36%
620393	Desarrollo de aplicación usando ASP.NET	27%
620465	Desarrollo aplicación web.	20%
620390	Fundamentos de programación en Java.	20%

2

Ilustración 19: Lista de recomendaciones obtenidas

- 1: Lista de las cuatro mejores opciones de electivos a inscribir, basándose en las preferencias del estudiante.
- 2: Lista de las cuatro mejores opciones de electivos a inscribir, basándose en el desempeño esperado.

Funcionamiento

- Se obtienen los diferentes valores de recomendación para cada electivo, utilizando la fórmula de *Valor de recomendación₁* (Ver página 35), traspasada a código *C#* de *ASP.NET*.
- Se obtienen los diferentes valores de recomendación para cada electivo, utilizando la fórmula de *Valor de recomendación₂* (Ver página 40), traspasada a código *C#* de *ASP.NET*.
- Los resultados se guardan un *INSERT* en la tabla *RECOMIENDA* de la base de datos.
- Los datos de las listas son cargados automáticamente desde la tabla *RECOMIENDA*, haciendo uso de la clase *DataContext* de *LINQ*, se mostrarán los cuatro mayores valores, primero de aceptación y luego de desempeño.

Inscribir electivos

Descripción: El estudiante puede ver e inscribir los electivos donde cumpla los requisitos. El alumno puede o no tomar en cuenta las sugerencias de electivos que eventualmente se le presentaron.

Selección de Electivos

Selecciona los electivos que quieres inscribir este semestre.

1

CODIGO	NOMBRE	PROFESOR	INSCRITOS	HORARIO	DETALLES	
620445	Administración y Manejo de GNU/Linux y software.	Juan José Ramirez	3/3	Viernes 09:50 a 12:30. Viernes 14:10 a 15:30.		<input type="checkbox"/>
620421	Taller de Programación para Dispositivos Móviles.	Victor Ruiz	1/30	Sábado 09:40 a 12:30.		<input type="checkbox"/>
620390	Fundamentos de programación en Java.	Jazna Meza	1/30	Martes 11:10 a 19:00		<input type="checkbox"/>
620393	Desarrollo de aplicación usando ASP.NET	Gonzalo Perez	1/30	Lunes 17:50 a 20:00.		<input type="checkbox"/>
620465	Desarrollo aplicación web.	Jonathan Muñoz	0/30	Martes 18:10 a 21:00		<input type="checkbox"/>

Aceptar

2

Ilustración 20: Lista de electivos a impartir en el semestre

- Lista de electivos de donde el alumno puede escoger cuáles inscribir, con la libertad de considerar o no la recomendación entregada. Al hacer click en [Ver](#), se abrirá un documento PDF con información detallada sobre el electivo.
- Botón que permite al alumno inscribir los electivos seleccionados.

Funcionamiento

- Los datos de la lista son cargados automáticamente desde la tabla *ELECTIVOS*, mediante el uso de *LinqDataSource* de *LINQ*.
- El registro es validado, verificando que el estudiante no inscriba más de cuatro electivos, que cumpla los requisitos y que dichos electivos tengan cupos disponibles, para posteriormente ser guardado mediante un *INSERT* en la tabla *INSCRIBE*.

Eliminar inscripción

Descripción: El estudiante puede cancelar la preinscripción en caso que lo desee.

Eliminar Inscripción

Haz clic en el icono para eliminar un electivo.

CODIGO	NOMBRE	
620464	Programación avanzada en SQL.	

1

Ilustración 21: Lista de electivos que se puede cancelar su preinscripción

- 1: Lista de todos los electivos preinscritos por el estudiante, que puede cancelar con el ícono de cruz.

Funcionamiento

- Los datos de la lista son cargados automáticamente desde la tabla *INSCRIBE*, mediante el uso de *LinqDataSource* de *LINQ*.
- La eliminación se hace mediante un *DELETE* en la tabla *INSCRIBE*, posterior a la confirmación del alumno.

Recuperación de datos de ingreso

Descripción: Los tres tipos de usuarios pueden recuperar sus datos de ingreso en caso que los hayan olvidado.

¿Olvidaste tus datos de acceso?

Ingresar tu correo electrónico : ①

Enviar ②

The image shows a web form for password recovery. At the top, it asks '¿Olvidaste tus datos de acceso?'. Below this is a text input field with the label 'Ingresar tu correo electrónico :'. A circled number '1' is placed to the right of the input field. Below the input field is a button labeled 'Enviar', with a circled number '2' placed to its right.

Ilustración 22: Recuperación de datos de ingresos

- 2: Espacio donde el usuario ingresa su correo electrónico institucional.
- 3: Botón que permite al usuario enviar el correo ingresado para recuperar sus datos de ingreso.

Funcionamiento

- Se verifica que el correo corresponda a un usuario registrado, en caso de serlo se le envía un mensaje con sus datos de acceso.

Enviar informes y sugerencias

Descripción: Los tres tipos de usuarios que ingresan al sistema tienen la posibilidad de acceder a este módulo, donde podrán enviar sus dudas, sugerencias o problemas encontrados al momento de interactuar con el sistema.

Enviar Informes o Sugerencias

Asunto: ①

Mensaje: ②

③

Ilustración 23: Formulario para envío de Informes o sugerencias

- 1: Espacio donde el usuario ingresa el asunto de su mensaje.
- 2: Espacio donde se detalla el motivo del mensaje.
- 3: Botón que permite al usuario enviar el mensaje al administrador.

Funcionamiento

- El correo es obtenido directamente desde la base de datos, utilizando el identificador del usuario que envía el mensaje.

Cambiar contraseña

Descripción: Los tres tipos de usuarios que ingresan al sistema tienen la posibilidad de acceder a este módulo, donde podrán cambiar su contraseña en caso que lo deseen.

Cambiar Contraseña

Contraseña actual: ①

Nueva contraseña: ②

Repetir contraseña: ③

④

Ilustración 24: Formulario para cambio de contraseña

- 1: Espacio donde el usuario ingresa su contraseña actual.
- 2: Espacio donde el usuario ingresa la nueva contraseña.
- 3: Espacio donde el usuario repite la nueva contraseña, para confirmar que es la que desea.
- 4: Botón que permite al usuario cambiar su contraseña.

Funcionamiento

- Se verifica que la contraseña actual sea efectivamente la ingresada.
- Se verifica que el usuario ingrese una contraseña distinta a la actual, y dos veces para garantizar que está bien escrita.

9. CONCLUSIONES Y TRABAJO FUTURO

No cabe duda que los sistemas de recomendación han llegado para quedarse, sin embargo aún les queda mucho por mejorar.

Aunque en un comienzo su objetivo era mejorar las ventas de las tiendas online, al sugerir productos de interés para ciertos perfiles de clientes, hoy tienen muchas aplicaciones, ya que se pueden adaptar a cualquier área, siendo hoy muy utilizados en las redes sociales.

La utilización de estos sistemas en el ámbito educacional ha resultado de gran utilidad, ya que la gran cantidad de material académico de baja veracidad en la red hace que los sistemas de recomendación sean casi una necesidad.

Esta valoración sobre los sistemas de recomendación en el área académica motivó la búsqueda y posterior solución a algún problema dentro de la universidad, utilizando esta tecnología.

La inscripción de electivos, si bien no es un problema mayor, afecta a muchos estudiantes al momento de elegir y resulta una situación perfecta para aplicar las ventajas de los sistemas de recomendación.

En este sistema se emplearon algoritmos y técnicas especialmente diseñados para solucionar la problemática de inscripción de electivos.

Utilizar ASP.NET permite desarrollar un sitio web potente, de manera fácil, además de poder programar el comportamiento de los distintos controles de la página. La sencillez al interactuar directamente con la base de datos también es otra ventaja importante.

Los beneficios que se espera lograr con la propuesta de esta tesis es una mayor satisfacción y desempeño de los estudiantes al momento de cursar un electivo, manteniéndolos mejor informados y tomando en cuenta sus preferencias; esta situación conlleva, también, a una menor tasa de deserción.

Finalmente, se puede aportar que los sistemas de recomendación aún tienen muchas cosas que pulir, así como otros problemas que no dependen meramente de ellos.

Respecto a trabajos futuros, se espera mejorar el desempeño de la base de datos, para que responda con mayor rapidez, y para asegurar su funcionamiento ante solicitudes de múltiples usuarios a la vez, evitando posibles problemas de transacciones intercaladas como lectura de datos no comprometidos, lecturas no repetidas o sobre-escritura de datos no comprometidos.

Observar la aceptación de los usuarios por la interfaz del sistema y adecuarla si fuese necesario, también es una medida a tomar al corto plazo.

Otro trabajo que se espera desarrollar a futuro, es evaluar el desempeño del sistema de modo de observar que tanto, los estudiantes, aceptan las recomendaciones, y si los resultados son los esperados, para optimizar los algoritmos hasta lograr la mayor cantidad de expectativas cumplidas, y agregar un enfoque colaborativo, y relacionar estudiantes con características similares, y así lograr recomendaciones óptimas, cada vez más asertivas.

Referencias

1. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating Collaborative Filtering Recommender Systems. ACM, (2004)
2. Roberto, J., Martí, M., Rosso, P.: Sistemas de recomendación basados en lenguaje natural: Opiniones Vs. Valoraciones
3. Galán, S.: Filtrado colaborativo y Sistemas de recomendación, (2007)
4. Barroco, M., Pérez, L., Martínez, L.: Un sistema de recomendación basado en conocimiento con información lingüística multigranular, (2006)
5. Cabal, C., Martínez, F., Molina, V.: Sistemas de recomendación en el Comercio Electrónico y la E-educación, (2010)
6. Zaïane, O.: Building a Recommender Agent for e-Learning Systems. Proceedings of the International Conference on Computers in Education, (2002).
7. Richard M.: 5 Problems of Recommender Systems, http://www.readwriteweb.com/archives/5_problems_of_recommender_systems.php.
8. Han P., Xie B., Yang F., Shen R.: A scalable p2p recommender system based on distributed collaborative filtering, (2004).
9. Middleton S., Shadbolt N., De Roure D.: Ontological user profiling in recommender systems, (2004).
10. Adomavicius G., Tuzhilin A.: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, (2005).
11. Resnick, P., Varian H.: Recommender systems. Communications of the ACM, (1997)
12. Guo, X.: Personalized Government Online Services with Recommendation Techniques, (2006)
13. Zadeh, L.: Fuzzy logic, IEEE Computer, (1988)
14. Morales, G.: Introducción a la lógica difusa, (2002)
15. Tseng, L.: A genetic clustering algorithm for data with non-spherical-shape clusters. Pattern Recognition, (1999)
16. Tocarruncho, S., Aponte, F., Tocarruncho, A.: Extracción de perfiles basada en agrupamiento genético para recomendación de contenido, (2007)
17. Sarwar, B.: Analysis of Recommendation Algorithms for E-Commerce. Proceedings of the 2nd ACM conference on Electronic Commerce, (2000)

10. ANEXO

10.1. Detalle del Modelo Entidad-Relación.

Estudiante	
Atributo	Descripción
est_rut	RUT (Rol Único Tributario) del estudiante, sin su dígito verificador.
est_dv	Dígito verificador del RUT del estudiante.
est_nombres	Nombre o nombres del estudiante.
est_apellidop	Apellido paterno del estudiante.
est_apellidom	Apellido materno del estudiante.
est_mail	Correo electrónico institucional del estudiante.
est_pass	Contraseña del estudiante para ingresar al sistema.

Cursó	
Atributo	Descripción
promedio	Promedio del estudiante en una asignatura.

Asignatura	
Atributo	Descripción
as_cod	Código que identifica a la asignatura.
as_nombre	Nombre de la asignatura.

Recomienda	
Atributo	Descripción
aprobacion	Indica si el estudiante consideró o no la recomendación de un electivo.
valor_gusto	Valor de recomendación obtenido en base a las preferencias del estudiante.
valor_nota	Valor de recomendación obtenido en base a las notas del estudiante.
rango	Representa el rango de desempeño esperado para el estudiante.

Electivo	
Atributo	Descripción
ele_cod	Código que identifica al electivo.
ele_nombre	Nombre del electivo.
ele_descripcion	Descripción general del electivo.
ele_cupos	Cantidad de alumnos permitidos en el electivo.
ele_inscritos	Cantidad de alumnos inscritos en el electivo.
ele_horario	Horario en que se impartirá el electivo, en caso que esté definido.

Cuestionario	
Atributo	Descripción
respuesta	Respuesta del estudiante a una afirmación.

Afirmacion	
Atributo	Descripción
afi_id	Código que identifica a la afirmación.
afi_afirmacion	Afirmación sobre uno o algunos electivos.

Área	
Atributo	Descripción
ar_id	Código que identifica al área de especialidad.
ar_nombre	Nombre del área de especialidad.

PerteneceA	
Atributo	Descripción
per_grado	Grado de pertenencia del electivo a un área de especialidad.

Relaciona	
Atributo	Descripción
multiplicador	Afirmación sobre uno o algunos electivos.

Profesor	
Atributo	Descripción
pro_rut	RUT del profesor, sin su dígito verificador.
pro_dv	Dígito verificador del RUT del profesor.
pro_nombres	Nombre o nombres del profesor.
pro_apellidop	Apellido paterno del profesor.
pro_apellidom	Apellido materno del profesor.
pro_mail	Correo electrónico institucional del profesor.
pro_pass	Contraseña del profesor para ingresar al sistema.

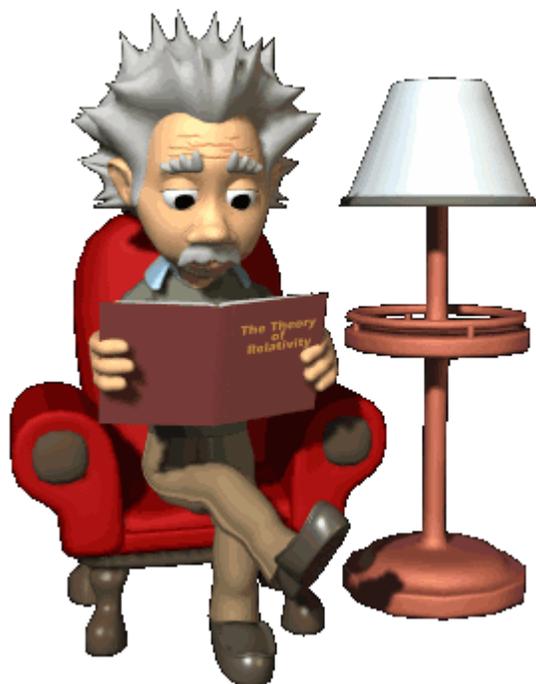
Jefe de carrera	
Atributo	Descripción
jef_rut	RUT del jefe de carrera, sin su dígito verificador.
jef_dv	Dígito verificador del RUT del jefe de carrera.
jef_nombres	Nombre o nombres del jefe de carrera.
jef_apellidop	Apellido paterno del jefe de carrera.
jef_apellidom	Apellido materno del jefe de carrera.
jef_mail	Correo electrónico institucional del jefe de carrera.
jef_pass	Contraseña del jefe de carrera para ingresar al sistema.
fecha_max	Fecha máxima para la preinscripción de electivos.

10.2. Manuales de usuario



Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática

Manual para el jefe de carrera



Contenido

Fijar fecha máxima	2
Gestionar electivos	3
Ingresar nuevo electivo	4
Consultar alumnos	4
Enviar sugerencias	6
Cambiar contraseña	7

PRIMEROS PASOS

Antes de usar lea atentamente el manual.

El Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática es un sistema que permite a los estudiantes de la carrera recibir recomendaciones en base a sus gustos y calificaciones. Además el sistema agiliza el proceso de inscripción de electivos mediante una preinscripción.

Recuerde que el sistema depende de su correcto uso, el jefe de carrera es el responsable de ingresar los electivos que se impartirán en el semestre y de la posterior inscripción formal en la Intranet Institucional de la Universidad del Bío-Bío.

Fijar fecha máxima

Al entrar al sistema, el jefe de carrera encontrará el espacio donde debe establecer la fecha máxima para la preinscripción de electivos.

Plazo máximo para preinscripción:

Puedes ver el manual

mayo, 2013						
lu	ma	mi	ju	vi	sá	do
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Today: mayo 14, 2013

Este será el plazo máximo para que los estudiantes puedan recibir recomendaciones, preinscribir o cancelar la preinscripción de electivos.

Gestionar electivos

El jefe de carrera debe gestionar los electivos que se impartirán durante el semestre.

Gestión de Electivos

Ingresar nuevo electivo  **1** **2**

CODIGO	NOMBRE	CUPOS	PROFESOR		
620390	Fundamentos de programación en Java.	30	Meza		
620393	Desarrollo de aplicación usando ASP.NET	30	Perez		
620421	Taller de Programación para Dispositivos Móviles.	30	Ruiz		
620445	Administración y Manejo de GNU/Linux y software.	3	Ramirez		
620464	Programación avanzada en SQL.	30	Vidal		
620465	Desarrollo aplicación web.	30	Muñoz		

1. Donde el jefe de carrera puede ingresar un nuevo electivo al pulsar  .
2. Lista de electivos ingresados por el jefe de carrera, puede modificarlos al pulsar  y eliminarlos al pulsar  .

Ingresar nuevo electivo

El jefe de carrera debe ingresar los detalles del nuevo electivo.

Ingresar nuevo Electivo

Código: ①

Nombre: ②

Profesor: ③

Cupos: ④

⑤

1. Espacio donde el jefe de carrera debe ingresar el código del electivo.
2. Espacio donde el jefe de carrera debe ingresar el nombre del electivo.
3. Lista de profesores, donde el jefe de carrera debe asignar al electivo.
4. Espacio donde el jefe de carrera debe ingresar los cupos para el electivo.
5. Botón que permite guardar el nuevo electivo.

Nota: El jefe de carrera no ingresa profesores al sistema; esta es tarea del Administrador, quien debe ingresarlos como usuario.

Nota: Todos los campos son obligatorios.

Consultar alumnos

El jefe de carrera puede revisar los alumnos registrados por electivo.

Listado de Alumnos por Electivo

Electivo: **1** 3 registros encontrados. **2**

RUT	NOMBRES	AP. PATERNO	AP. MATERNO
17212941-2	Teresa	Ulloa	Placencia
17538724-2	Marcelo	Torres	Ríos
15677441-3	Pablo	Leiva	Faundez

3

1. Lista de electivos, donde el jefe de carrera debe escoger para consultar alumnos.
2. Cantidad total de alumnos registrados en el electivo seleccionado.
3. Lista de alumnos registrados en el electivo seleccionado.

Enviar sugerencias

El jefe de carrera puede enviar sugerencias o consultas que surjan al interactuar con el sistema.

Enviar Informes o Sugerencias

Asunto: ①

Mensaje: ②

Enviar ③

1. Espacio donde el jefe de carrera debe ingresar el asunto del mensaje.
2. Espacio donde el jefe de carrera debe redactar el mensaje.
3. Botón que permite enviar el mensaje.

Cambiar contraseña

El jefe de carrera puede cambiar su contraseña en caso que lo desee.

Cambiar Contraseña

Contraseña actual: ①

Nueva contraseña: ②

Repetir contraseña: ③

④

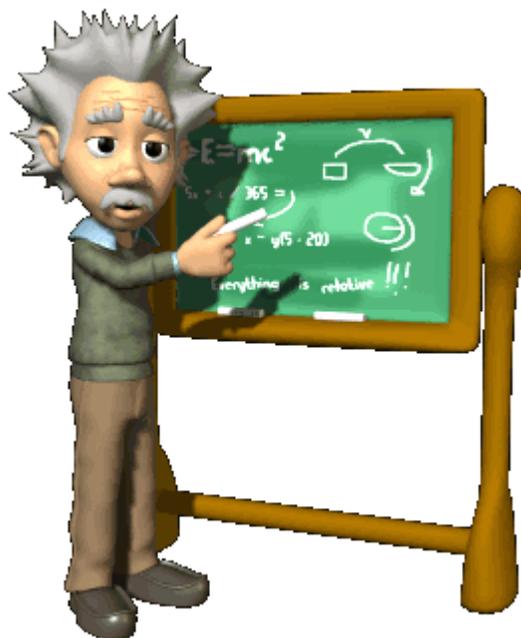
1. Espacio donde el jefe de carrera debe ingresar su contraseña actual, para confirmar que se trata del usuario legítimo.
2. Espacio donde el jefe de carrera debe ingresar su nueva contraseña.
3. Espacio donde el jefe de carrera debe repetir la nueva contraseña, para confirmar que no hay errores.
4. Botón que permite cambiar la contraseña.

Nota: La contraseña nueva debe ser diferente a la contraseña actual.



Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática

Manual para el profesor



Contenido

Gestionar electivos	2
Gestionar requisitos académicos	3-4
Gestionar requisitos de encuesta	5
Consultar alumnos	6
Enviar sugerencias	7
Cambiar contraseña	8

PRIMEROS PASOS

Antes de usar lea atentamente el manual.

El Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática es un sistema que permite a los estudiantes de la carrera recibir recomendaciones en base a sus gustos y calificaciones. Además el sistema agiliza el proceso de inscripción de electivos mediante una preinscripción.

Recuerde que el sistema depende de su correcto uso, el profesor es el responsable de ingresar los detalles de los electivos que impartirá en el semestre, información primordial para la generación de recomendaciones.

La malla de la carrera define tres áreas de formación en la especialidad, cada electivo pertenece en diferentes grados a cada una de ellas.

Gestionar electivos

El profesor puede gestionar los electivos que impartirá durante el semestre.

Gestión de Electivos

Lista de electivos a impartir

CODIGO	NOMBRE	HORARIO	CUPOS
620390	Fundamentos de programación en Java.	Martes 11:10 a 19:00	30

Edición de electivos

Electivo: **2**

Horario: **3**

4

1. Lista de electivos que el profesor impartirá durante el semestre.
2. Lista de electivos, donde el profesor debe escoger para modificar.
3. Espacio donde el profesor puede ingresar el horario del electivo seleccionado, en caso que esté definido.

Gestionar requisitos académicos

El profesor debe gestionar los requisitos académicos de los electivos que impartirá. Esta información es importante al momento de generar recomendaciones.

Gestión de Requisitos Académicos

Seleccionar electivo: **1**

Asignación de porcentajes de área:

Ingeniería de Software y Bases de Datos: % **2**

Sistemas Computacionales: % **3**

Gestión Informática: % **4**

Seleccionar requisitos: **5**

Filtrar: Área Semestre

6

Seleccionar asignaturas: **7**

- Introducción a la Programación
- Programación Orientada a Objetos
- Estructuras discretas para Cs. de la Co
- Estructuras de datos
- Administración general
- Fundamentos de Cs. de la Computación
- Gestión Contable
- Modelamiento de Procesos de Informac
- Análisis y diseño de algoritmos

9

10

11

8

- 1.** Lista de electivos, donde el profesor debe escoger para establecer áreas y requerimientos.
- 2.** Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Ingeniería de Software y Base de Datos.
- 3.** Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Sistemas Computacionales.
- 4.** Espacio donde el profesor asigna el valor porcentual de pertenencia del electivo seleccionado al área de especialidad Gestión informática.

Nota: La suma de estos tres valores debe ser exactamente 100, ni más ni menos.

- 5.** El profesor escoge si filtrar las asignaturas de la carrera por Área o por Semestre.
- 6.** Si se escoge filtrar por Área, se podrán escoger las asignaturas que pertenezcan al área de especialidad seleccionada. Si se escoge filtrar por Semestre, se podrán escoger las asignaturas que pertenezcan, por malla, entre el primer y décimo semestre.
- 7.** Lista de total de asignaturas, o bien las que satisfacen al filtro previamente ingresado. El profesor puede seleccionar aquellas que considere como requisito para que un estudiante curse el electivo seleccionado.
- 8.** Lista de las asignaturas que el profesor considere como requisito para que un estudiante curse el electivo seleccionado.
- 9.** Botón que permite al profesor agregar a la lista de requisitos un ramo de la lista de asignaturas.
- 10.** Botón que permite al profesor quitar de la lista de requisitos una asignatura.
- 11.** Botón que permite al profesor guardar los cambios efectuados en el electivo seleccionado.

Gestionar requisitos de encuesta

El profesor debe establecer si cada una de las afirmaciones, de la encuesta presentada al alumno, afectan positivamente, negativamente o no afectan a cada electivo que impartirá.

Gestión de Requisitos para Encuesta

Para que el sistema pueda ofrecer recomendaciones, debes establecer si cada afirmación afecta o no al electivo que impartes, por lo que te pedimos que respondas de forma objetiva y responsable, ya que no podrás modificarlo posteriormente.

Instrucciones: Marca la opción que corresponda considerando lo siguiente.

+ : Afecta positivamente.

0 : No afecta.

- : Afecta negativamente.

Seleccionar electivo: 1

Nº	CUESTIONARIO	+	0	-
1	Me interesa conocer las virtudes ofrecidas por los software libres	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	Tengo conocimiento sobre redes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
3	La seguridad informática no es mi fuerte	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
4	No tengo buen desempeño con las bases de datos	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5	Me es fácil encontrar soluciones óptimas	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6	Me es fácil modelar información	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7	Me interesa aprender más sobre SQL	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
8	No tengo un dispositivo móvil moderno	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
9	Me interesa aprender a desarrollar aplicaciones móviles	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
10	Me gusta programar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Lista de electivos, donde el profesor debe escoger para gestionar.

7. Lista de afirmaciones que el profesor debe puntuar.

Nota: Una vez establecidos los requisitos de encuesta no se pueden modificar posteriormente.

Consultar alumnos

El profesor puede revisar los alumnos registrados en los electivos que impartirá.

Listado de Alumnos por Electivo

Electivo: **1** 3 registros encontrados. **2**

RUT	NOMBRES	AP. PATERNO	AP. MATERNO
17212941-2	Teresa	Ulloa	Placencia
17538724-2	Marcelo	Torres	Ríos
15677441-3	Pablo	Leiva	Faundez

3

- 4.** Lista de electivos, donde el profesor debe escoger para consultar alumnos.
- 5.** Cantidad total de alumnos registrados en el electivo seleccionado.
- 6.** Lista de alumnos registrados en el electivo seleccionado.

Enviar sugerencias

El profesor puede enviar sugerencias o consultas que surjan al interactuar con el sistema.

Enviar Informes o Sugerencias

Asunto: ①

Mensaje: ②

③

4. Espacio donde el profesor debe ingresar el asunto del mensaje.
5. Espacio donde el profesor debe redactar el mensaje.
6. Botón que permite enviar el mensaje.

Cambiar contraseña

El profesor puede cambiar su contraseña en caso que lo desee.

Cambiar Contraseña

Contraseña actual: ①

Nueva contraseña: ②

Repetir contraseña: ③

④

- 5. Espacio donde el profesor debe ingresar su contraseña actual, para confirmar que se trata del usuario legítimo.
- 6. Espacio donde el profesor debe ingresar su nueva contraseña.
- 7. Espacio donde el profesor debe repetir la nueva contraseña, para confirmar que no hay errores.
- 8. Botón que permite cambiar la contraseña.

Nota: La contraseña nueva debe ser diferente a la contraseña actual.



Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática

Manual para el estudiante



Contenido

Recibir recomendaciones

Encuesta	2
Encuesta contestada	3
Recomendaciones	4-5

Inscripción de electivos

Inscribir electivos	6
Eliminar inscripción de electivos	7
Enviar sugerencias	8
Cambiar contraseña	9

PRIMEROS PASOS

Antes de usar lea atentamente el manual.

El Sistema recomendador de electivos para estudiantes de Ingeniería Civil en Informática es un sistema que permite a los estudiantes de la carrera recibir recomendaciones en base a sus gustos y calificaciones. Además el sistema agiliza el proceso de inscripción de electivos mediante una preinscripción.

En este sistema podrá preinscribir electivos, tendrá la lista completa de electivos y sus características. Además podrá recibir recomendaciones, en base a sus preferencias y calificaciones, para ello debe responder una encuesta. Podrá hacer esto las veces que quiera, hasta la fecha máxima establecida. Recuerde que la inscripción formal de electivo en la Intranet Institucional de la Universidad del Bío-Bío se realizará cuando terminen las preinscripciones.

Encuesta

Si el estudiante desea recibir recomendaciones de electivos debe contestar una encuesta.

Recomendaciones

Para que el sistema pueda ofrecer recomendaciones, debes responder la siguiente encuesta de opinión, por lo que te pedimos que respondas de forma objetiva y responsable.

Instrucciones: La pauta de evaluación debe ser completada marcando la opción que mejor describa tu apreciación en cada uno de los indicadores que se presentan.

Para ello considera la siguiente escala:

- 1: Muy de acuerdo.
- 2: De acuerdo.
- 3: Indiferente.
- 4: En desacuerdo.
- 5: Muy en desacuerdo.

Nº	AFIRMACIONES	1	2	3	4	5
1	Me interesa conocer las virtudes ofrecidas por los software libres	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
2	Tengo conocimiento sobre redes	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	La seguridad informática no es mi fuerte	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4	No tengo buen desempeño con las bases de datos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5	Me es fácil encontrar soluciones óptimas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
6	Me es fácil modelar información	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
7	Me interesa aprender más sobre SQL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
8	No tengo un dispositivo móvil moderno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
9	Me interesa aprender a desarrollar aplicaciones móviles	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10	Me gusta programar	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[Ver recomendaciones](#)

1. Lista de afirmaciones que el estudiante debe puntuar de acuerdo a su criterio, considerando la escala establecida.
2. Botón que permite guardar la encuesta y recibir recomendaciones.

Encuesta contestada

Esto ocurre en caso que el estudiante dese responder nuevamente la encuesta, o ver otra vez las recomendaciones después de responder la encuesta.

Recomendaciones

Ya has completado la encuesta.



3. Botón que permite al estudiante ver las recomendaciones considerando las últimas respuestas de la encuesta.
4. Botón que permite al estudiante contestar la encuesta nuevamente.

Recomendaciones

Una vez el estudiante responde el cuestionario recibirá recomendaciones basadas tanto en sus preferencias obtenidas de la encuesta, como basada en calificaciones obtenidas de su informe curricular. Se entregan valores porcentuales de desempeño esperado.

Recomendaciones

Las siguientes recomendaciones están basadas en los resultados de la encuesta que respondiste. Sin embargo, se sugiere que consideres también el nivel de desempeño esperado.

CODIGO	NOMBRE	ACEPTACIÓN
620393	Desarrollo de aplicación usando ASP.NET	53%
620464	Programación avanzada en SQL.	41%
620465	Desarrollo aplicación web.	36%
620390	Fundamentos de programación en Java.	15%
620445	Administración y Manejo de GNU/Linux y software.	8%
620421	Taller de Programación para Dispositivos Móviles.	0%

Las siguientes recomendaciones están basadas en tu rendimiento académico durante todo el período académico.

CODIGO	NOMBRE	DESEMPEÑO
620464	Programación avanzada en SQL.	39%
620421	Taller de Programación para Dispositivos Móviles.	37%
620445	Administración y Manejo de GNU/Linux y software.	36%
620393	Desarrollo de aplicación usando ASP.NET	27%
620465	Desarrollo aplicación web.	20%
620390	Fundamentos de programación en Java.	20%

8. Lista de electivos recomendados en base a las preferencias del estudiante, ordenados descendientemente.
9. Lista de electivos recomendados en base a las calificaciones del estudiante, ordenados descendientemente.

Nota: La recomendación basada en calificaciones considera cuatro niveles de desempeño, representados por un color.

Nivel	Color
Malo	Rojo
Regular malo	Amarillo
Regular bueno	Verde claro
Bueno	Verde oscuro

Nota: El estudiante debe considerar tanto el porcentaje mostrado como el color, por ejemplo es mejor **25%** que **90%**.

Inscribir electivos

El jefe de carrera puede revisar los alumnos registrados por electivo.

Selección de Electivos

Selecciona los electivos que quieres inscribir este semestre.

1

CODIGO	NOMBRE	PROFESOR	INSCRITOS	HORARIO	DETALLES	
620445	Administración y Manejo de GNU/Linux y software.	Juan José Ramirez	3/3	Viernes 09:50 a 12:30. Viernes 14:10 a 15:30.		<input type="checkbox"/>
620421	Taller de Programación para Dispositivos Móviles.	Victor Ruiz	1/30	Sábado 09:40 a 12:30.		<input type="checkbox"/>
620390	Fundamentos de programación en Java.	Jazna Meza	1/30	Martes 11:10 a 19:00		<input type="checkbox"/>
620393	Desarrollo de aplicación usando ASP.NET	Gonzalo Perez	1/30	Lunes 17:50 a 20:00.		<input type="checkbox"/>
620465	Desarrollo aplicación web.	Jonathan Muñoz	0/30	Martes 18:10 a 21:00		<input type="checkbox"/>

Aceptar

2

7. Lista de electivos que se impartirán durante el semestre.
8. Botón que permite inscribir los electivos seleccionados.

Nota: Puede ver los detalles del electivo al pulsar .

Nota: El estudiante puede inscribir hasta un máximo de cuatro electivos.

Nota: El estudiante sólo puede inscribir los electivos que tengan cupos disponibles y donde cumpla los requisitos.

Eliminar inscripción de electivos

El estudiante puede cancelar la inscripción de electivos si lo desea.

Eliminar Inscripción

Haz clic en el icono para eliminar un electivo.

CODIGO	NOMBRE	
620464	Programación avanzada en SQL.	

1

1. Lista de electivos, inscritos por el estudiante, que puede eliminar al pulsar  .

Enviar sugerencias

El estudiante puede enviar sugerencias o consultas que surjan al interactuar con el sistema.

Enviar Informes o Sugerencias

Asunto: ①

Mensaje: ②

③

- 7. Espacio donde el estudiante debe ingresar el asunto del mensaje.
- 8. Espacio donde el estudiante debe redactar el mensaje.
- 9. Botón que permite enviar el mensaje.

Cambiar contraseña

El estudiante puede cambiar su contraseña en caso que lo desee.

Cambiar Contraseña

Contraseña actual: ①

Nueva contraseña: ②

Repetir contraseña: ③

Aceptar ④

- 9.** Espacio donde el estudiante debe ingresar su contraseña actual, para confirmar que se trata del usuario legítimo.
- 10.** Espacio donde el estudiante debe ingresar su nueva contraseña.
- 11.** Espacio donde el estudiante debe repetir la nueva contraseña, para confirmar que no hay errores.
- 12.** Botón que permite cambiar la contraseña.

Nota: La contraseña nueva debe ser diferente a la contraseña actual.