



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

GESTOR DE INFORMACIÓN SOBRE ENSEÑANZA DE LA INGENIERÍA DE  
SOFTWARE EN CHILE

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN COMPUTACIÓN

JAVIER ANDRÉS ESPINOZA VILLARROEL

PROFESOR GUÍA:  
SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:  
JOSÉ BENGURIA DONOSO  
ALEXANDRE BERGEL  
JOCELYN SIMMONDS WAGEMANN

SANTIAGO DE CHILE  
2019

# Resumen

La enseñanza de la ingeniería de software en Chile es un desafío muy grande, dada a la cantidad de áreas de conocimiento que se deben abordar en los cursos de esta disciplina. Muchas carreras de computación e informática abordan la enseñanza de la ingeniería de software de manera ad hoc, generando así una brecha entre lo que se recomienda enseñar y lo que efectivamente se enseña en este ámbito. Para identificar el tamaño de esta brecha y las áreas de conocimiento en la que ésta aparece, académicos del Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile tomaron el currículum de referencia para carreras de ingeniería de software, propuesto por la Asociación de Maquinaria Computacional (ACM) y el Instituto de Ingeniería Eléctrica y Electrónica (IEEE) en el 2014, y en base a ella recopilamos información de la enseñanza de la ingeniería de software en Chile mediante análisis de páginas web de las universidades en Chile y entrevistas telefónicas. Dado el alto costo de este levantamiento de información, estudiantes de Ingeniería Civil en Computación de la misma universidad desarrollaron un sistema Web que permitía aplicar una encuesta en línea a profesores de cursos de ingeniería de software, recolectando así los datos del proceso anterior. Sin embargo, el sistema requería bastantes mejoras para ser utilizado en la práctica; dichas mejoras apuntaban a temas de usabilidad, escalabilidad, utilidad y robustez.

En este trabajo de memoria se realizaron las mejoras requeridas, tanto en el sistema para aplicar la encuesta, como también en el procesamiento y visualización de la información recolectada.

Para llegar a ese resultado se analizó el sistema de encuesta anterior, y también el currículum elaborado por la ACM e IEEE usado como referencia. Con esa información, se propuso un nuevo modelo de datos que aborda los problemas de escalabilidad de la solución anterior. Se diseñaron canales para comunicarle, a los profesores de ingeniería de software en Chile, el agregado de la información ingresada a través de reportes automáticos, mostrando información a nivel nacional, por institución y por carrera. Además, el sistema entrega diversas visualizaciones que complementan la información mostrada en los informes.

El nuevo sistema fue evaluado con profesores del área de ingeniería de software del Departamento de Ciencias de la Computación de la Universidad de Chile, y la retroalimentación obtenida sirvió para la mejora evolutiva del sistema. Con esto, se espera que sea lanzado a producción prontamente.

La nueva funcionalidad, así como la mejora de aquella implementada en la versión anterior del sistema, permitieron que el sistema se vuelva más usable, útil y robusto. Ahora existe una herramienta que provee información útil acerca de la enseñanza de la ingeniería de software en distintas universidades de Chile, lo que permite realizar un diagnóstico periódico (se espera que sea anual) e identificar los ámbitos donde aparece la brecha entre lo que las universidades enseñan, y lo que las recomendaciones internacionales proponen enseñar.



*Dedico este trabajo a toda la gente que conocí en mi camino.*



# Agradecimientos

Agradezco a mi familia por todo el apoyo que me dieron a través de este difícil periodo. Gracias a mis dos hermanos, Ignacio y Eduardo, a quienes podría escribirles un ensayo de por qué son tan importantes en mi vida, y gracias a mi madre y padre, Patricia y Eduardo.

A mis amigos de mi ex-colegio que siempre estuvieron ahí para apoyarme y hacerme reír, especialmente Sebastián Montaña, Diego Garay, Diego Abarca, y Diego Rebolledo.

A mis nuevos amigos que hice a través de este viaje de 7 años. En especial gracias a Joaquín Torres, Nicolás Bravo, Sergio Peñafiel, Belisario Panay, Américo Ferrada, Gabriel Sanhueza y Juan Rojas por haberme adoptado en su grupo de amigos cuando yo estaba solo en el DCC.

Muchas gracias a mi profesor guía, Sergio Ochoa, por siempre motivarme a aspirar a ser alguien mejor después de cada reunión semanal que teníamos. También a la profesora co-guía Jocelyn Simmonds por las risas y motivación que siempre me daba.

Por último, agradezco a Francisco Gutiérrez por haber prendido una llama en mí que pensé que había perdido luego del 2017.

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Motivación . . . . .	3
1.3. Objetivos de la Memoria . . . . .	4
1.4. Metodología de Trabajo . . . . .	4
1.5. Estructura de la Memoria . . . . .	5
<b>2. Marco teórico</b>	<b>6</b>
2.1. Software Engineering Currículum Guidelines 2014 . . . . .	6
2.2. Gestor de recolección de datos . . . . .	8
2.2.1. El proceso de encuesta . . . . .	8
2.2.2. Arquitectura física del sistema . . . . .	9
2.2.3. Arquitectura lógica . . . . .	9
2.2.4. Modelo de datos . . . . .	11
2.2.5. Encuesta . . . . .	12
2.2.6. Gráficos e informes generados por el sistema . . . . .	13
2.2.7. Otros . . . . .	15
2.2.8. Estado de la segunda versión sistema . . . . .	15
<b>3. Análisis y rediseño del sistema legado</b>	<b>18</b>
3.1. Análisis de la usabilidad y utilidad del sistema legado . . . . .	18
3.1.1. Focus group . . . . .	18
3.1.2. Lluvia de ideas acerca de los análisis . . . . .	21
3.2. Análisis detallado del sistema legado . . . . .	22
3.2.1. Arquitectura física . . . . .	22
3.2.2. Arquitectura lógica . . . . .	23
3.2.3. Modelo de datos . . . . .	24
3.2.4. Interfaces del sistema . . . . .	26
3.3. Requisitos de la solución . . . . .	28
3.3.1. Requisitos funcionales . . . . .	28
3.3.2. Requisitos no funcionales . . . . .	29
3.4. Rediseño realizado . . . . .	29
3.4.1. Arquitectura lógica . . . . .	29
3.4.2. Modelo de datos . . . . .	31
3.5. Resumen . . . . .	32

<b>4. Solución desarrollada</b>	<b>34</b>
4.1. Análisis implementadas . . . . .	34
4.1.1. Análisis de la lluvia de ideas . . . . .	34
4.1.2. Revisión de librerías . . . . .	35
4.1.3. Visualizaciones a usar . . . . .	35
4.1.4. Implementación de la visualizaciones . . . . .	40
4.2. Reestructuración del reporte . . . . .	41
4.2.1. Aspectos generales del reporte . . . . .	41
4.2.2. Estructura del reporte nuevo . . . . .	42
4.2.3. Generación de plantillas parametrizadas . . . . .	43
4.2.4. Implementación del reporte . . . . .	44
4.3. Cambios a las interfaces del sistema . . . . .	47
4.3.1. Encuesta . . . . .	48
4.3.2. Profesores participantes . . . . .	48
4.3.3. Reportes por institución . . . . .	49
4.4. Otros cambios realizados . . . . .	49
4.5. Resumen . . . . .	51
<b>5. Validación</b>	<b>52</b>
5.1. Validación de usabilidad y utilidad del sistema . . . . .	52
5.1.1. Protocolo de la validación . . . . .	52
5.1.2. Piloto de la encuesta . . . . .	53
5.1.3. Primera ciclo de entrevistas . . . . .	53
5.1.4. Discusión de los resultados . . . . .	57
5.1.5. Segundo ciclo de entrevistas . . . . .	60
5.1.6. Discusión de los resultados . . . . .	62
5.2. Resumen . . . . .	63
<b>6. Conclusiones</b>	<b>65</b>
<b>Bibliografía</b>	<b>68</b>
<b>A. Apéndice</b>	<b>70</b>
A.1. Reporte generado para el periodo de piloto . . . . .	70
A.2. Ejemplos de visualizaciones generadas por la segunda versión del sistema . . . . .	70
A.3. Interactividad de las visualizaciones . . . . .	86

# Índice de Ilustraciones

1.1. Metodología del trabajo a llevarse a cabo. . . . .	5
2.1. Pauta propuesta por el Currículum de Referencia de ACM/IEEE . . . . .	7
2.2. Extracto de la encuesta presentada por la segunda versión de la aplicación Web. . . . .	9
2.3. Arquitectura física del sistema. . . . .	9
2.4. Arquitectura lógica del sistema basada en patrón de tres capas. . . . .	10
2.5. Modelo de datos del sistema. . . . .	12
2.6. Extracto de los términos y condiciones que el usuario debe aceptar para comenzar la encuesta. . . . .	14
2.7. Extracto de interfaz de la primera parte de la encuesta. . . . .	16
2.8. Extracto de interfaz de la segunda parte de la encuesta. . . . .	17
2.9. Extracto de interfaz de la tercera parte de la encuesta. . . . .	17
3.1. Resumen de un periodo en específico. . . . .	19
3.2. Visualizaciones para el área “Fundamentos de la Computación”. . . . .	20
3.3. Visualizaciones para el área “Práctica Profesional”. . . . .	20
3.4. Interfaz de usuario del administrador, que muestra los periodos completados y activos. . . . .	26
3.5. Interfaz de usuario que muestra las carreras ingresadas en el sistema. . . . .	26
3.6. Interfaz que muestra los profesores ingresados en el sistema. . . . .	27
3.7. Arquitectura lógica del nuevo sistema. Los rectángulos en rojo muestran los módulos agregados en la solución. . . . .	30
3.8. Nuevo modelo de datos de la solución. . . . .	32
4.1. Ejemplo de visualización usando Chart.js. . . . .	36
4.2. Visualización del análisis <b>M1</b> implementada en el sistema. . . . .	38
4.3. Visualización del análisis <b>M2</b> . . . . .	38
4.4. Visualización del análisis <b>M3</b> . . . . .	39
4.5. Visualización del análisis <b>M4</b> . . . . .	39
4.6. Extracto de código que asocia una carrera con un periodo, creando así un vínculo específico para los resultados de esa carrera. . . . .	40
4.7. Interfaz que permite descargar o enviar reportes a los profesores. El cuadrado rojo corresponde a la opción de mandar los reportes a los profesores. El cuadrado verde indica la opción de descargar el informe de la encuesta en específico. . . . .	41
4.8. Tabla generada al compilar el archivo del listado 2. . . . .	46
4.9. Extracto de la primera parte de la encuesta. . . . .	47

4.10.	Extracto mejorado de los términos y condiciones de participación en la encuesta	48
4.11.	Extracto de interfaz mejorada de la segunda parte de la encuesta. . . . .	49
4.12.	Extracto de interfaz mejorada de los profesores que han dado su autorización para mostrar su nombre como participantes de la encuesta. . . . .	50
4.13.	Botón que redirige al usuario a la interfaz para descargar reportes de institución.	50
4.14.	Interfaz que permite descargar reportes por institución. . . . .	51
5.1.	Primera visualización generada para el periodo del piloto, que corresponde al análisis <b>M1</b> . . . . .	54
5.2.	Segunda visualización generada para el periodo del piloto, que corresponde al análisis <b>M2</b> . . . . .	54
5.3.	Tercera visualización generada para el periodo del piloto, que corresponde al análisis <b>M3</b> . . . . .	55
5.4.	Cuarta visualización generada para el periodo del piloto, que corresponde al análisis <b>M4</b> . . . . .	56
5.5.	Primer extracto de la página de contexto entregada a los profesores. . . . .	59
5.6.	Segundo extracto de la página de contexto entregada a los profesores. . . . .	59
5.7.	En este gráfico, se compara la cantidad de alumnos egresados de una carrera en específico, con la cantidad de alumnos egresados en promedio a nivel de las carreras que participaron en la encuesta. . . . .	60
5.8.	En este gráfico, se muestra la evolución de las horas de docencia semanales teóricas y prácticas a través de los semestres de una carrera. . . . .	61
5.9.	En este gráfico, se comparan las horas totales dedicadas a la enseñanza de las áreas de conocimiento de una carrera, con las horas recomendadas por el documento de la ACM. . . . .	62
5.10.	En este gráfico, se comparan las horas totales dedicadas a la enseñanza de las áreas de conocimiento de una carrera, con el promedio de horas totales a nivel de las carreras que participaron en la encuesta. . . . .	63
A.1.	Gráfico de barras que cuenta la cantidad de programas que no enseñan una unidad de conocimiento. . . . .	70
A.2.	Gráfico de barras que compara el promedio de cantidad de horas totales a nivel de país con las horas recomendadas por el documento de la ACM. . . . .	86
A.3.	Información mostrada al usuario al momento de mover el mouse cerca de las barras de la visualización. . . . .	87
A.4.	Información mostrada al usuario al momento de mover el mouse cerca de las barras de la visualización. . . . .	87
A.5.	Información mostrada al usuario al momento de mover el mouse cerca de los puntos de la visualización. . . . .	88
A.6.	Información mostrada al usuario al momento de mover el mouse cerca de los puntos de la visualización. . . . .	89



# Capítulo 1

## Introducción

### 1.1. Antecedentes

Actualmente en Chile existen tres tipos de carreras relacionadas con Computación e Informática: Ingeniería en Computación, Ingeniería Civil en Computación y/o Informática, e Ingeniería de Ejecución en Computación y/o Informática. En estas carreras se ven los fundamentos de ciencias de la computación, abordando distintas subdisciplinas y viendo la aplicación práctica de estos a la ingeniería. Una de esas áreas es la ingeniería de software, la cual tiene un impacto importante en la industria del software, pues, según la acreditación de la carrera Ingeniería Civil en Computación de la Universidad de Chile llevada a cabo el 2014, muchos de los egresados de las carreras antes mencionadas desarrollan su profesión como ingenieros de software, por lo que la enseñanza de la ingeniería de software se vuelve un tema importante.

La enseñanza de esta disciplina es un desafío muy grande debido, entre otras cosas, a la gran cantidad de conocimiento que se debe entregar a los estudiantes para que luego puedan desempeñarse profesionalmente en el mundo laboral [1, 2]. Estos conocimientos le permiten a los futuros ingenieros abordar diversas actividades, como por ejemplo, idear un nuevo producto de software, estudiar los procesos de software asociados al desarrollo de productos, realizar el análisis y diseño de una aplicación, o implementar y evaluar su usabilidad y utilidad.

Por otra parte, se presenta la necesidad de enseñar la ingeniería de software de forma teórica-práctica, debido a la naturaleza del conocimiento a entregar [3]. Dado que en las carreras de computación hay muchos ámbitos en los cuales se le debe entregar conocimiento a los alumnos, donde ingeniería de software es uno de ellos, existen fuertes restricciones de tiempo (y también en las mallas de los programas) para poder dictar cursos de este ámbito. Esto significa que en las carreras de computación e informática en Chile usualmente hay poco espacio (1 a 3 cursos de esta área) para entregarles a los estudiantes conocimiento teórico-práctico sobre una gran cantidad de tópicos de ingeniería de software. Dado que el conocimiento se construye a partir de los aspectos más fundamentales, los cursos de ingeniería de software en estas carreras parten enfocándose en dichos aspectos, sin poder ahondar muchas veces en conocimientos específicos, lo cual es usualmente requerido por los nuevos profesionales que

se insertan en la industria. Esta situación crea una brecha entre lo que necesita la industria del software y lo que enseñan las universidades [4, 5]. Debido a eso, los ingenieros recién egresados tendrán vacíos de conocimiento que tendrán que suplir durante su vida profesional, y mientras más vacíos tengan estas personas, menor será el impacto de las universidades sobre la industria del software. Es por eso que es importante para las universidades y para la industria (y por lo tanto, para el país) diagnosticar el estado en que se encuentra la formación de sus ingenieros de software al momento de egresar de las carreras, identificando posibles falencias en la formación de estos profesionales de cara a su incorporación a la industria.

Para hacer frente a este problema, académicos del Departamento de Ciencias de la Computación (DCC) de la Universidad de Chile llevaron a cabo una serie de entrevistas en el 2015, con el fin de realizar un diagnóstico e identificar las eventuales falencias antes mencionadas. Esta encuesta involucró a 20 universidades a lo largo del país, y en primer lugar buscó determinar cómo se impartía el conocimiento en el ámbito de la ingeniería de software en Chile, y qué conocimiento se les entregaba a los alumnos [6]. Dicha encuesta utilizó como base el currículum de referencia para carreras de ingeniería de software, desarrollado por la Asociación de Maquinaria Computacional (ACM) y el Instituto de Ingeniería Eléctrica y Electrónica (IEEE) publicado el 2014 [7]. Este currículum de referencia considera las necesidades de la industria a nivel internacional. Por lo tanto, si se identifica la brecha entre lo que propone esta recomendación y lo que efectivamente se enseña en la universidades, entonces las universidades tienen la posibilidad de determinar la brecha entre lo que ellas enseñan y lo que requiere la industria. De todas maneras, se reconoce que las recomendaciones internacionales no necesariamente van a estar completamente alineadas con lo que requiere la industria chilena de software.

El currículum de referencia propuesto por la ACM e IEEE entrega recomendaciones acerca del conocimiento de ingeniería de software que se les debería entregar a los alumnos, y la cantidad de horas de docencia totales que debería asignarse a cada tópico. Dado el alto nivel de aceptación que esta guía de currículum tiene entre las universidades, se la tomó como referencia al momento de desarrollar dicha investigación. La encuesta realizada en el 2015 se llevó a cabo llamando por teléfono, e involucró (como encuestados) a docentes de distintas universidades que impartían cursos en el área. Los resultados eran ingresados a una planilla Excel que contenía información acerca de las unidades de conocimiento impartidas en los cursos correspondientes, la forma en que se evaluaba a los alumnos, si los cursos tenían o no proyectos, entre otras preguntas.

Este proceso demostró ser muy costoso, en términos de tiempo y esfuerzo, además de ser propenso a errores. Por lo tanto se vio la necesidad de contar con un sistema Web que pudiera recolectar, procesar y representar los datos de las encuestas de manera fácil y cómoda.

Si bien se podrían haber usado sistemas de gestores de encuestas como Google Forms para elaborar esta encuesta, esto haría que el procesamiento de los datos ingresados fuese manual, lo que involucra una gran cantidad de trabajo y tiempo. Tomando en cuenta, además, que esta encuesta se quiere llevar a cabo cada cierto tiempo (anualmente o cada dos años), era mejor crear un sistema que pudiese procesar esos datos de manera automática.

Por lo tanto, estudiantes de la carrera de Ingeniería Civil en Computación de la Universidad de Chile, durante el final de 2017 (primera versión) [8] y principios del 2018 (segunda

versión) [9], desarrollaron un gestor de datos para aplicar una encuesta sobre la enseñanza ingeniería de software en Chile, y así, recolectar los datos que fueron preguntados en el proceso anterior. Sin embargo, para poder ponerse en producción este sistema requería diversas mejoras, por ejemplo, en términos de procesamiento automático de los datos recolectados recolectados, y de entrega de resultados. Por otra parte, la usabilidad del sistema no había sido evaluada con usuarios reales, lo cual generaba incertidumbre al momento de llevar a cabo la encuesta con docentes de las diversas carreras que enseñan ingeniería de software. En el desarrollo de la primera versión de dicho sistema participó el autor de esta memoria como jefe de proyecto.

## 1.2. Motivación

El problema general que se abordó en este trabajo de memoria, es la falta de un diagnóstico acerca cómo se enseña la ingeniería de software en Chile actualmente, en comparación con la recomendación de la ACM/IEEE [7]. Luego, en base a dicha comparación se buscó identificar falencias en la formación de los futuros ingenieros de software de Chile. Como se mencionó antes, se creó un sistema recolector de datos para tratar de abordar este desafío, donde uno de los objetivos de este sistema fue recolectar datos pertinentes para responder las siguientes preguntas:

1. ¿Cómo se enseña ingeniería de software en Chile?
2. ¿Qué tan lejos está lo que se enseña en Chile con lo que recomienda el currículum de la ACM/IEEE?
3. ¿Qué cosas se deberían agregar a los programas de ingeniería en computación en Chile para acercarse a la recomendación de la ACM/IEEE?

Las respuestas a estas preguntas no podía obtenerse con ninguna de las dos versiones del sistema, ni permitía hacer un diagnóstico detallado de la situación. Dichas versiones sólo entregaban un archivo que resumía las respuestas de los docentes que respondieron la encuesta. Se hipotetizó que está información no era suficiente para motivar a un profesor de un curso de ingeniería de software, a responder anualmente la encuesta. Eso disminuía la probabilidad de que este estudio pudiera repetirse periódicamente, y se volviera valioso para las instituciones educativas, los docentes y la industria.

Había cuatro tópicos importantes que abordar en la segunda versión del sistema (y por lo tanto, en esta memoria), y de esa manera poder obtener un sistema estable que pudiera ponerse en producción y así ayudar a responder las tres preguntas ya mencionadas. Estos tópicos eran los siguientes:

- La usabilidad de la última versión, es decir, la segunda, no había sido evaluada por usuarios finales. Tampoco se había evaluado la intuitividad de los formularios de ingreso de datos. Esta situación incrementaba la probabilidad de recolectar datos sin sentido al momento de comenzar el período de encuestas, pues los usuarios podrían ingresar datos erróneos debido a la falta de claridad de las preguntas de la encuesta, y la falta de guía de las interfaces del sistema; particularmente en los formularios de ingreso de información.
- No estaba claro si la motivación para responder la encuesta era lo suficientemente alta,

como para que los docentes quieran responderla todos los años. De la misma manera, tampoco se sabía cuánto era el esfuerzo que ellos tenía que hacer para completar la encuesta.

- El procesamiento de datos debía ser lo más automático posible, para poder así llevar a cabo anualmente la encuesta, invirtiendo un esfuerzo razonable que permitiera su repetición. Este procesamiento automático de datos debía además entregar información de manera más rápida a los docentes, universidades e industria.
- El sistema no era lo suficientemente robusto como para poder llevar a cabo las operaciones ya mencionadas, pues aún tenía algunos errores del back-end y front-end que impedían su puesta en producción.
- La información que entregaba el sistema, como resultado de la encuesta, era muy pobre como para llevar a cabo un análisis de la brecha entre lo que recomienda la ACM/IEEE, y lo que hacen las universidades.

Resolviendo estos problemas, se busca que el software sea lo suficientemente estable, usable y útil como para poder llevar a cabo periodos de tiempo donde los profesores de ingeniería de software puedan responder la encuesta. Esto permitiría identificar el estado actual de la educación de la ingeniería de software en Chile, y mejorarla a lo largo del tiempo.

### 1.3. Objetivos de la Memoria

El objetivo general de esta memoria es mejorar la usabilidad, utilidad y robustez del sistema legado, automatizando tanto como sea posible el procesamiento de los resultados de las encuestas, y entregando información valiosa para los docentes y las instituciones a las que ellos pertenecen. Para poder lograr el objetivo general, se requiere cumplir los siguientes objetivos específicos:

1. Mejorar la utilidad del software desarrollado, y de los datos capturados por dicho sistema.
2. Implementar mecanismos que automaticen el procesamiento de los datos de las encuestas.
3. Diseñar e implementar análisis de datos, visualizaciones o estadísticas apropiadas y valiosas, que permitan motivar a los docentes a responder la encuesta anualmente.
4. Validar la utilidad del nuevo sistema, escogiendo un grupo de docentes para que respondan la encuesta.

### 1.4. Metodología de Trabajo

Para desarrollar el trabajo de memoria se utilizó una metodología iterativa-incremental, pues ciertos hitos de la memoria, como el mejoramiento del procesamiento de datos, requiere de un estudio de usabilidad para ser desarrollado. Los más importantes fueron los siguientes:

- Lo primero que se hizo fue analizar el código realizado por los alumnos del curso Ingeniería de Software II para el semestre Otoño del año 2018. En este hito se analizaron los módulos creados, la base de datos que el sistema utilizaba, y la arquitectura física del sistema.

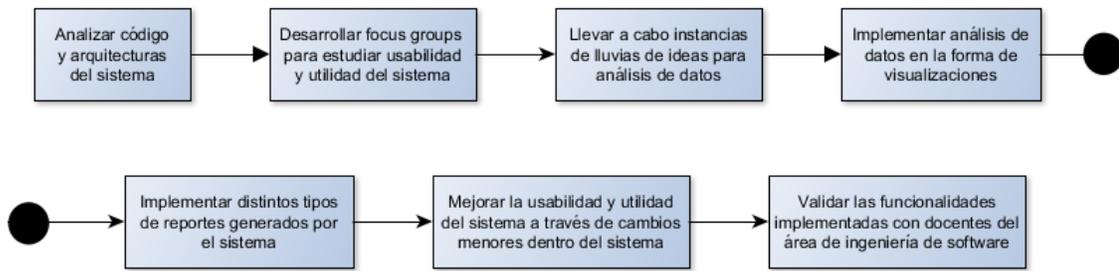


Figura 1.1: Metodología del trabajo a llevarse a cabo.

- Luego de llevar a cabo este análisis, se desarrolló un focus group con 5 profesores de ingeniería de software. En esta instancia, se estudió la usabilidad y utilidad del software y los resultados que éste generaba. Además, se generó una lluvia de ideas para posibles análisis de datos que el sistema podría llevar a cabo.
- En el primer tercio del semestre Otoño del año 2019, se diseñaron e implementaron análisis de datos y visualizaciones propuestas en el hito anterior, para motivar a los docentes a responder la encuesta.
- En el segundo tercio del mismo semestre, se estudió y desarrolló otra manera de entregar información procesada a los docentes. Esto dio como resultado un reporte que extendía el reporte generado por la segunda versión del sistema.
- En el último tercio del mismo semestre, se mejoró la funcionalidad y los productos obtenidos en las dos iteraciones anteriores a través de cambios menores como mejoramiento de las rutas (URLs) del sistema, rediseño de interfaces, y optimización o refactorización de código.
- Por último, se validaron las funcionalidades implementadas con tres profesores de la rama de la ingeniería de software, obteniendo así un input de los usuarios finales (y expertos) del sistema, que sirvieron para cambiar ciertos aspectos de los análisis de datos implementados por el memorista.

La figura 1.1 muestra el diagrama de flujo de esta metodología de trabajo.

## 1.5. Estructura de la Memoria

El resto del documento se estructura en seis capítulos. El Capítulo 2 habla acerca del estado del sistema recolector de datos (sistema legado) y del currículum elaborado por la ACM/IEEE. El Capítulo 3 presenta los requisitos del nuevo sistema y la propuesta de la solución. El Capítulo 4 explica la implementación de los requisitos indicados en el capítulo anterior, mientras que en el Capítulo 5 muestra los resultados de la validación realizada. En el Capítulo 6 se muestran las conclusiones del trabajo realizado y posibles líneas de trabajo para mejoras del sistema desarrollado.

# Capítulo 2

## Marco teórico

En este capítulo se explican los términos que serán usados a lo largo de este trabajo de memoria. Estos conceptos son cruciales para entender el trabajo realizado. En la sección 2.1 se explica qué es el Currículum Guideline 2014 publicado por la ACM/IEEE, se detalla su contenido, y qué términos de éste son importantes. La sección 2.2 describe el Gestos de Recolección de datos, que es el sistema legado por los estudiantes de Ingeniería de Software II. Allí se explica el flujo de trabajo (workflow) del proceso de encuesta, la arquitectura física, lógica y el modelo de datos previo al trabajo realizado en la memoria.

### 2.1. Software Engineering Currículum Guidelines 2014

El currículum de referencia 2014 para carreras de Ingeniería de Software (o Software Engineering Currículum Guidelines 2014), propuesto por la ACM/IEEE [7], es un documento que establece pautas de lo que se debería enseñar sobre esta temática en programas de pregrado en informática o computación. Este documento aborda distintos temas, como por ejemplo: la definición de la ingeniería de software, su evolución a través de los años, y qué es lo que se espera de un alumno de pregrado que pasa por cursos de ingeniería de software, entre otros. Una de las partes más importantes del documento habla acerca de los tres niveles de abstracción en los que se puede describir las temáticas asociadas a la enseñanza de la ingeniería de software: áreas de conocimiento, unidades de conocimiento y tópicos de conocimiento.

- Área de conocimiento (AC): Este es el nivel más alto de la jerarquía, representando una subdisciplina de la ingeniería de software que es reconocida como una parte significativa del conocimiento que un alumno de pregrado debería saber. En el documento, existen 10 áreas de conocimiento. Éstas son:
  1. Fundamentos de la Computación.
  2. Fundamentos de la Matemática e Ingeniería.
  3. Práctica Profesional.
  4. Modelamiento y Análisis de Software.
  5. Análisis y Especificación de Requerimientos.
  6. Diseño de Software.

KA/KU	Title	Hours	KA/KU	Title	Hours
<b>CMP</b>	<b>Computing essentials</b>	<b>152</b>	<b>DES</b>	<b>Software design</b>	<b>48</b>
CMP.cf	Computer science foundations	120	DES.con	Design concepts	3
CMP.ct	Construction technologies	20	DES.str	Design strategies	6
CMP.tl	Construction tools	12	DES.ar	Architectural design	12
			DES.hci	Human-computer interaction design	10
			DES.dd	Detailed design	14
			DES.ev	Design evaluation	3
<b>FND</b>	<b>Mathematical and engineering fundamentals</b>	<b>80</b>	<b>VAV</b>	<b>Software verification and validation</b>	<b>37</b>
FND.mf	Mathematical foundations	50	VAV.fnd	V&V terminology and foundations	5
FND.ef	Engineering foundations for software	22	VAV.rev	Reviews and static analysis	9
FND.ec	Engineering economics for software	8	VAV.tst	Testing	18
			VAV.par	Problem analysis and reporting	5
<b>PRF</b>	<b>Professional practice</b>	<b>29</b>	<b>PRO</b>	<b>Software process</b>	<b>33</b>
PRF.psy	Group dynamics and psychology	8	PRO.con	Process concepts	3
PRF.com	Communications skills (specific to SE)	15	PRO.imp	Process implementation	8
PRF.pr	Professionalism	6	PRO.pp	Project planning and tracking	8
			PRO.cm	Software configuration management	6
			PRO.evo	Evolution processes and activities	8
<b>MAA</b>	<b>Software modeling and analysis</b>	<b>28</b>	<b>QUA</b>	<b>Software quality</b>	<b>10</b>
MAA.md	Modeling foundations	8	QUA.cc	Software quality concepts and culture	2
MAA.tm	Types of models	12	QUA.pca	Process assurance	4
MAA.af	Analysis fundamentals	8	QUA.pda	Product assurance	4
<b>REQ</b>	<b>Requirements analysis and specification</b>	<b>30</b>	<b>SEC</b>	<b>Security</b>	<b>20</b>
REQ.rfd	Requirements fundamentals	6	SEC.sfd	Security fundamentals	4
REQ.er	Eliciting requirements	10	SEC.net	Computer and network security	8
REQ.rsd	Requirements specification and documentation	10	SEC.dev	Developing secure software	8
REQ.rv	Requirements validation	4			

Figura 2.1: Pauta propuesta por el Currículum de Referencia de ACM/IEEE

7. Validación y Verificación de Software.
8. Proceso de Software.
9. Calidad de Software.
10. Seguridad.

El documento de la ACM e IEEE, para cada una de estas áreas, recomienda una cantidad de horas pedagógicas que deberían ser dedicadas a la enseñanza de estas áreas. Cada AC puede tener asociada múltiples unidades de conocimiento.

- Unidades de conocimiento: Son módulos temáticos dentro de un área. Cada unidad posee distintos tópicos de conocimiento. Por ejemplo, en la figura 2.1, el área Computing essentials (o Fundamentos de la Computación) posee 3 unidades: *Fundamentos de la ciencia de la computación*, *Tecnologías de construcción*, y *Herramientas de construcción*.
- Tópicos de conocimiento: El tópico es el nivel más bajo de la jerarquía, usualmente se refiere a las actividades o conocimientos que se engloban en una unidad. Por ejemplo, la unidad *Herramientas de construcción* posee 4 tópicos: *Ambientes de desarrollo*, *Frameworks de interface*, *Unit testing*, y *Herramientas de análisis de performance*.

Este documento guía, si bien posee una estructura bien detallada, es difícil de entender y usarlo para una evaluación manual, como la experiencia reportada en [6], la cual buscó determinar la brecha entre lo que se enseña en Chile y lo que establece el currículum de referencia. La figura 2.1 muestra la división de áreas en unidades, detallando además las horas que se

recomienda dedicar a cada una de ellas. Por ejemplo, se ve que para *Computing essentials* o CMP se recomienda que la actividad de enseñanza aprendizaje involucre 152 horas, donde 120 corresponden a *Computer science foundations*, 20 a *Construction technologies* y 12 a *Construction tools*. Esto mismo es indicado para el resto de las áreas de conocimiento.

## 2.2. Gestor de recolección de datos

Como respuesta a la necesidad de obtener datos acerca de los cursos de ingeniería de software que se impartían en el marco de programas de computación e informática en Chile, se creó el sistema de recolección de datos, que recopila información de estos cursos (y de los programas) a través de una encuesta Web. Este sistema fue desarrollado durante el año 2017 y 2018 por alumnos de la Universidad de Chile, en el curso Ingeniería de Software II (CC5401), involucrando dos iteraciones distintas. El autor de esta memoria fue el líder del equipo desarrollador encargado de la primera iteración de dicho proyecto.

En la primera iteración se desarrolló el prototipo inicial del software, definiendo el workflow de la encuesta, una lógica base e interfaces, mientras que la segunda iteración amplió la funcionalidad y el alcance del software, mejorando la captura de datos, creando una motivación para que los docentes respondieran la encuesta. Esta motivación tenía que ver principalmente con la entrega de un informe con visualizaciones comparativas, donde el docente podía ver cómo se enseñan ciertos tópicos en las distintas universidades participantes. En ambas iteraciones se usó Python y Django [10] para el back-end y front-end, pues eran tecnologías que los estudiantes ya habían usado para previos proyectos, y porque ofrece protección contra inyecciones SQL por defecto.

La figura 2.2 muestra un extracto de la encuesta que los docentes tenían que responder al final de la segunda iteración del proyecto. En la tabla de esta interfaz, se muestra una pequeña explicación de las unidades de cada área, y al lado se encuentran los campos que deben llenar los docentes respecto a cada unidad. Estos campos corresponden al enfoque teórico y práctico que los profesores le dedican a la unidad de conocimiento.

### 2.2.1. El proceso de encuesta

El proceso de encuesta involucra los siguientes pasos:

1. Un administrador inicia un periodo de tiempo donde los profesores pueden contestar la encuesta.
2. Un administrador invita a participar a los docentes que enseñan cursos de ingeniería de software (en distintas universidades chilenas); esto se hace a través de correos que son enviados por el sistema. Dentro del correo de invitación se encuentra un hipervínculo que lo lleva a la encuesta correspondiente, para que le sea fácil al profesor responder dicha encuesta.
3. El docente acepta los términos y condiciones para participar en la encuesta. Luego de llenar ésta, ve un resumen de sus respuestas antes de enviar sus respuestas.
4. Se envía un informe al docente que detalla sus respuestas, además de algunos gráficos de barra que comparan la información que ellos llenaron en la encuesta, con la entregada por otros colegas del mismo programa.

EDUCA-IS: Encuesta sobre la Enseñanza de Ingeniería de Software

## Encuesta de curso: Ingeniería de Requisitos

Volver a: Área: Fundamentos IR Área: Fundamentos de la Computación

En esta sección se considera como enfoque la combinación de tiempo dedicado e importancia dada en evaluaciones a cada área. Considerando lo anterior se pide que responda cualitativamente, de acuerdo lo que usted considere pertinente qué tanto enfoque se usa, tanto teórica y como prácticamente, en la enseñanza de cada unidad dentro del área.

Nombre y descripción de la Unidad	NO enseño esta unidad	Enfoque Teórico	Enfoque Práctico
<b>Fundamentos de Ciencias de la Computación</b> Esta unidad incluye los siguientes temas: fundamentos de programación; algoritmos, estructura de datos y complejidad; técnicas de resolución de Problemas; uso y creación de abstracciones; la organización de un computador; E/S, mensajes de error, robustez; estructura y legibilidad de código; fundamentos de lenguajes de programación, sistemas operativos, bases de datos y redes de comunicación.	<input type="checkbox"/>	Ninguno  Mucho	Ninguno  Mucho
<b>Tecnologías para la Construcción</b> Esta unidad incluye los siguientes temas: diseño y uso de APIs; reuso de código y librerías; fundamentos de orientación a objetos; uso de aserciones; diseño por contrato; programación a la defensiva; manejo de los errores y excepciones; tolerancia a las fallas; técnicas de construcción basadas en estado y en tablas; configuración en tiempo de ejecución e internacionalización; parsing; primitivas para concurrencia; construcción de...	<input type="checkbox"/>	Ninguno  Mucho	Ninguno  Mucho

Figura 2.2: Extracto de la encuesta presentada por la segunda versión de la aplicación Web.



Figura 2.3: Arquitectura física del sistema.

A continuación se detallan varios elementos importantes del sistema, que se documentaron a partir del análisis detallado de dicha aplicación.

### 2.2.2. Arquitectura física del sistema

La arquitectura física del sistema se describe de la siguiente forma: un usuario (tanto administrador como docentes) accede al gestor a través de un navegador web, donde la aplicación corre en un servidor del DCC llamado Gestión Experimental de la Mejora del Software (GEMS). Luego de que se hagan ciertas operaciones dentro de la aplicación, ésta guardará información correspondiente en la base de datos alojada en el mismo servidor.

La figura 2.3 muestra un diagrama simple de la arquitectura física. La base de datos que la aplicación utiliza es un sistema de gestión de datos llamado MariaDB.

### 2.2.3. Arquitectura lógica

La implementación de la arquitectura del sistema se hizo utilizando el framework Django. En la figura 2.4 se presenta la arquitectura lógica de la segunda versión del software,

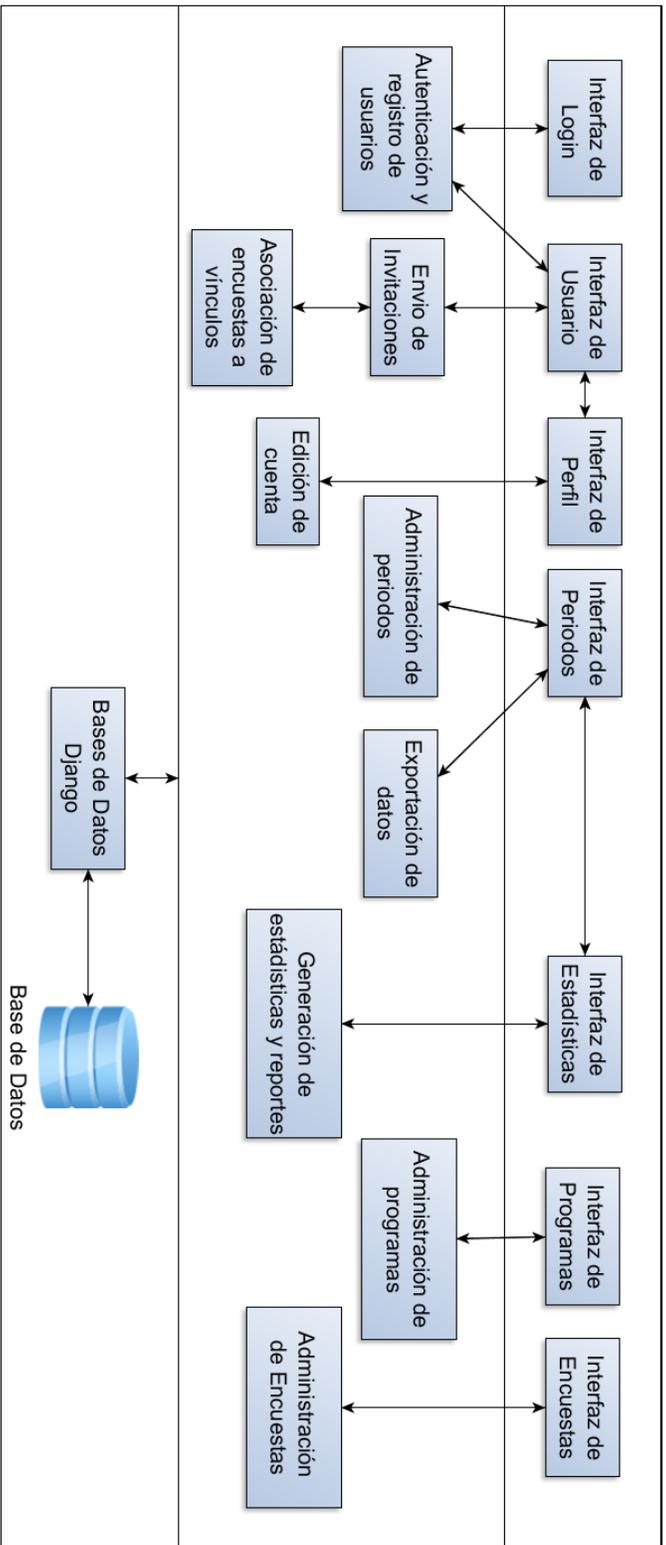


Figura 2.4: Arquitectura lógica del sistema basada en patrón de tres capas.

desarrollada por el grupo de Ingeniería de Software II del semestre Otoño del 2018.

La arquitectura lógica está basada en un modelo de tres capas, el cual consiste en separar las componentes importantes de un sistema de software en tres conjuntos jerárquicos: una capa de presentación o interfaz, otra que se encargue de la lógica de negocios, y por último, una capa donde residen los datos. Esto permite mantener un sistema estable, donde se pueden agregar diferentes interfaces, programas de lógica o datos, sin afectar a las capas restantes.

En la capa de lógica, los módulos (que pueden corresponder a un archivo o más dentro del sistema) más importantes de la arquitectura lógica son los de *Envío de Invitaciones*, *Asociación de encuestas a vínculos*, *Generación de Estadísticas y Reportes*, *Administración de Encuestas* y *Administración de periodos*. El primer módulo se encarga de enviar los enlaces a los profesores, usando el segundo módulo para asociar los vínculos entre cada respuesta a la encuesta y el profesor que la respondió. El tercer módulo se encarga de generar las estadísticas e informes para los profesores, mientras que el cuarto módulo se encarga de administrar todos los aspectos de la encuesta, ya sea guardar los datos en la base de datos, mostrarle información respectiva al docente, cambiar el flujo de la encuesta (workflow) según las áreas de conocimiento escogidas, entre otros. Por último, el quinto módulo se encarga de empezar, terminar, y revisar información acerca de los periodos donde se pueden responder las encuestas.

Las dependencias que se muestran en la figura 2.4, representadas por las flechas, fueron extraídas por los alumnos encargados de la segunda versión del software. Esto fue hecho a través de un análisis del código y una inspección de las declaraciones *import* en los archivos de código. Algunas dependencias entre vistas o interfaces corresponden a interfaces que están conectadas entre ellas. Es decir, una página muestra la otra, o viceversa. En el capítulo siguiente se mostrará una nueva versión de la arquitectura lógica, la que busca solucionar los problemas identificados en este sistema.

Los otros módulos se encargan de gestionar información de los administradores del sistema y de los programas ingresados a éste, exportar los datos brutos de un periodo, y manejar la autenticación de los administradores.

#### 2.2.4. Modelo de datos

Para poder guardar los datos que maneja el sistema, el equipo de desarrollo optó por una base de datos relacional con 12 tablas. En éste se guardan las diferentes entidades, tales como los profesores, las universidades, las carreras, y los resultados de cada encuesta, entre otros. Las tablas que se encuentran en la base de datos son las siguientes:

- Institución.
- Ciudad.
- Carrera.
- Curso.
- Profesor.
- Encuesta.
- Periodo.
- Estándar.
- Área Conocimiento.
- Unidad Conocimiento.
- Detalle Horas Área.
- Detalle Horas Unidad.

Las relaciones más importantes dentro del modelo de datos son las siguientes:

- Una institución puede poseer varias carreras asociadas.

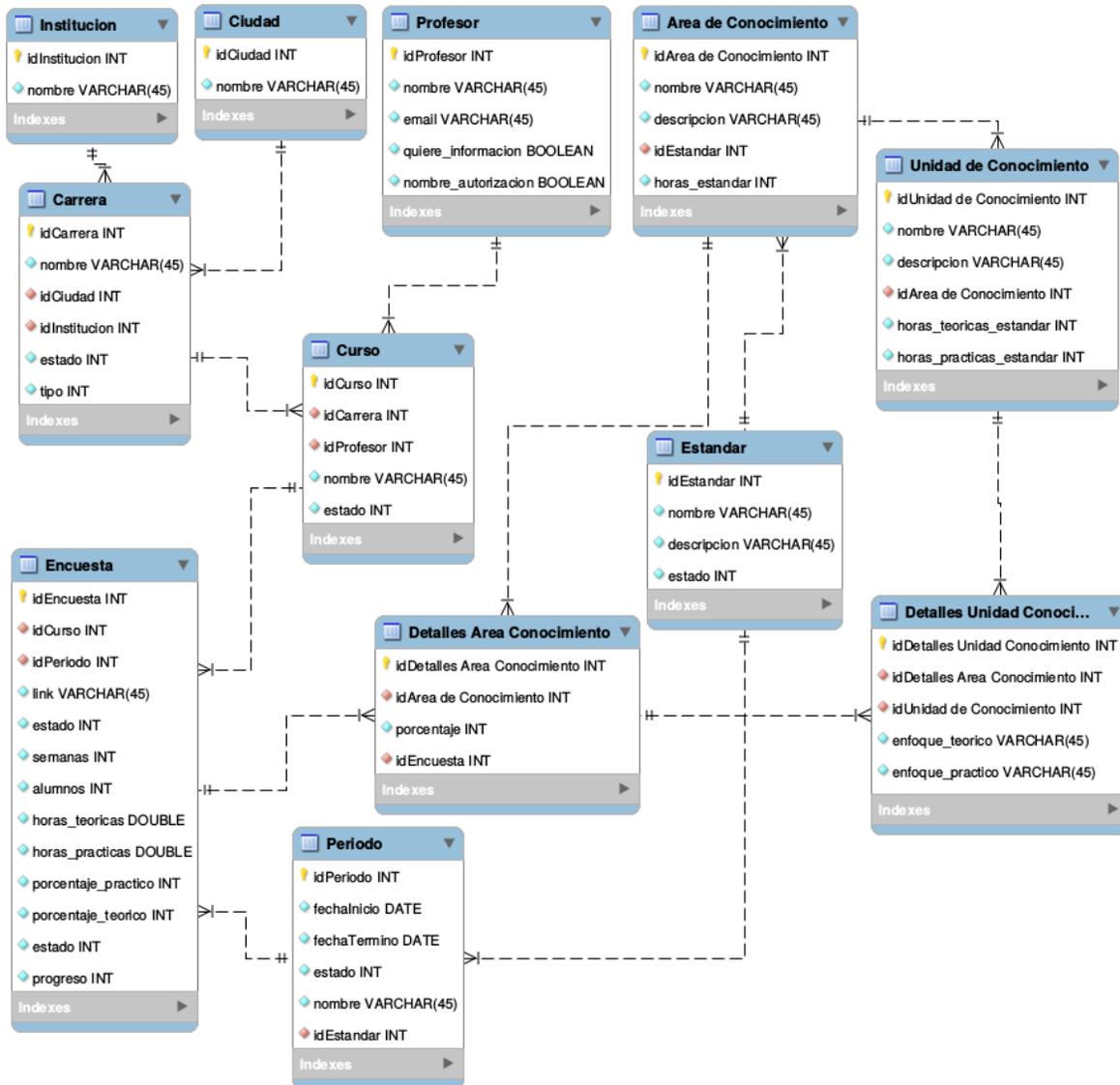


Figura 2.5: Modelo de datos del sistema.

- Una carrera puede poseer varios cursos asociados.
- Un profesor puede tener varios cursos asociados.
- Un curso puede tener varias encuestas asociadas (pero una por instancia de encuesta).

En la figura 2.5 se muestra un esquema de las tablas con sus filas correspondientes.

## 2.2.5. Encuesta

Para obtener la información necesaria, se diseñó, en las versiones pasadas, una encuesta que se asemejaba a la de las entrevistas llevadas a cabo por los académicos del DCC.

Para poder responder la encuesta, primero se debe crear un periodo de encuestas. Éste corresponde a un rango de fechas donde se pueden invitar a distintos docentes a que respondan la encuesta del sistema. Cuando los profesores son invitados, reciben un correo electrónico que posee un vínculo donde pueden ingresar a la encuesta de su curso para un periodo en

particular.

## Interfaces de la encuesta

Al momento de hacer click en el vínculo enviado a través del email al docente, el docente es redirigido a la encuesta de su curso correspondiente. Antes de llenar datos, al usuario se le presenta una pantalla con términos y condiciones que debe leer y aceptar para poder ingresar a la encuesta. En la figura 2.6 se muestra un extracto de esta información.

Luego de aceptar los términos, el docente es redirigido a la primera parte de la encuesta; la figura 2.7 muestra su interfaz correspondiente. En ésta se preguntan datos como la cantidad de semanas de clase, y la cantidad de horas teóricas y prácticas que se imparten a la semana, entre otros datos. Al ingresar esos datos, se muestra la segunda pantalla de la encuesta donde se indican las áreas que un profesor imparte en un curso, y también cuánto tiempo invierte la persona en esas áreas, en base a porcentajes. La figura 2.8 muestra un extracto de la segunda interfaz. Cabe destacar que la tabla mostrada tiene diez filas, una por cada área de conocimiento del Currículo de Referencia de la ACM/IEEE.

Después de detallar las áreas que el docente enseña, el sistema le muestra a éste una interfaz por cada área. Esta interfaz contiene las distintas unidades que un área posee, y el usuario debe declarar la dedicación de tiempo semanal de docencia directa que imparte, y el enfoque (teórico, teórico-práctico, o práctico) que le da a cada unidad. En la figura 2.9 se muestra un ejemplo de ésta.

El docente, si lo desea, puede navegar a través de las tres partes de la encuesta para corregir algún dato que ingresó mal o eliminar un área que pensaba que enseñaba.

### 2.2.6. Gráficos e informes generados por el sistema

Luego de finalizar un periodo de encuesta, el sistema generaba una serie de gráficos que servían como información para motivar a los docentes a seguir respondiendo la encuesta. Estos gráficos se guardaban como imágenes dentro de una carpeta en el sistema, que después eran cargadas dentro de la página. En la sección A.2 se encuentran dos ejemplos de estos gráficos.

Estos gráficos eran los siguientes:

- 10 gráficos que mostraban, por cada área de conocimiento, la cantidad de carreras que no enseñaban una cierta unidad de conocimiento.
- Otro gráfico representaba la cantidad de horas totales que el país le dedicaba a la enseñanza de las áreas de conocimiento, versus las horas totales que recomendaba el currículum de la ACM/IEEE.
- Por último, se encontraban 10 gráficos donde se mostraba la distribución de las horas dedicadas a las áreas de conocimiento por programa.

Por otro lado, el sistema generaba, usando un template de LaTeX, un reporte PDF para cada docente que respondió una encuesta, que tenía como objetivo mostrar información agregada y útil acerca de la enseñanza de la ingeniería de software. Sin embargo, lo único

## Encuesta de curso: Ingeniería de Software III

### Acuerdo de Confidencialidad

Al contestar esta encuesta, se atiene al acuerdo de confidencialidad siguiente:

Acuerdo de Confidencialidad y No Divulgación de información, que celebran por una parte EDUCAS quién en lo sucesivo se denominará "EL RECEPTOR", representado en este acto por Jocelyn Simmonds y por la otra el usuario que ingresa al sistema, quién a lo sucesivo se le denominará "EL DIVULGANTE", en su carácter de representante legal, al temor de las declaraciones y cláusulas siguientes:

**PRIMERA.-** Objeto. El presente Acuerdo se refiere a la información que EL DIVULGANTE proporcione al RECEPTOR, ya sea de forma oral, gráfica o escrita y, en estos dos últimos casos, ya esté contenida en cualquier tipo de documento, para ejecutar los estudios sobre Ingeniería de Software que se están desarrollando / que se van a desarrollar.

**SEGUNDA.-** 1. EL RECEPTOR únicamente utilizará la información facilitada por EL DIVULGANTE para el fin mencionado en la Estipulación anterior, comprometiéndose EL RECEPTOR a mantener la más estricta confidencialidad respecto de dicha información, advirtiendo de dicho deber de confidencialidad y secreto a sus empleados, asociados y a cualquier persona que, por su relación con EL RECEPTOR, deba tener acceso a dicha información para el correcto cumplimiento de las obligaciones del RECEPTOR para con EL DIVULGANTE.

2. EL RECEPTOR o las personas mencionadas en el párrafo anterior no podrán reproducir, modificar, hacer pública o divulgar a terceros de manera directa la información objeto del presente Acuerdo sin previa autorización escrita y expresa del DIVULGANTE.

3. De igual forma, EL RECEPTOR adoptará respecto de la información objeto de este Acuerdo las mismas medidas de seguridad que adoptaría normalmente respecto a la información confidencial de su propia Empresa, evitando en la medida de lo posible su pérdida, robo o sustracción.

Figura 2.6: Extracto de los términos y condiciones que el usuario debe aceptar para comenzar la encuesta.

que mostraba este informe era:

- Los datos que fueron ingresados por el docente en cada pregunta de la encuesta.
- Los gráficos mencionados anteriormente.

### 2.2.7. Otros

La seguridad del sistema es manejada a través de Django, el framework en el cual la aplicación está hecha. Si bien la aplicación utiliza HTTPS para manejar parte de su seguridad, Django maneja problemas como inyecciones SQL pues sus consultas (queries) son construidas usando parametrización. El código de una consulta SQL es definido separadamente de los parámetros de la consulta, y ya que los parámetros son dados por el usuario, éstos son escapados de inmediato.

Otros problemas de seguridad que Django maneja inherentemente son problemas *Cross-site scripting* (XSS), y *Cross site request forgery* (CRSF).

### 2.2.8. Estado de la segunda versión sistema

El sistema legado estuvo cerca de cumplir con la funcionalidad mínima requerida para llevar a cabo la encuesta. Esta funcionalidad corresponde a iniciar un periodo de encuestas, almacenar los datos ingresados, y entregar información según los datos de las encuestas. Sin embargo, había varios puntos que impedían su puesta en producción. Por ejemplo, no era robusto debido al acotado tiempo de testing y ajuste que tuvieron los desarrolladores. Además, gran parte del procesamiento de los resultados debía hacerse de forma manual a través del procesamiento de un archivo de datos en bruto, pues la funcionalidad para ello no estaba disponible en la segunda versión del sistema. Debido a eso, el procesamiento de datos automatizado no podía ser llevado a cabo. También, dado a la falta de validación de los formularios de entrada, era factible el ingreso involuntario de información sin sentido (o errónea). Esto podía ocurrir ya sea por falta de claridad de los mensajes mostrados en los formularios de encuesta, o simplemente un error humano al ingresar datos en los formularios del sistema.

Esta segunda versión generaba archivos de imágenes para cada gráfico incluido en el reporte que éste generaba, el cual era muy básico y no entregaba información agregada a los profesores. Dado a que se quiere utilizar este software cada 1 o 2 años, generar una serie de imágenes para cada visualización podía eventualmente producir problemas de escalabilidad, especialmente si se implementaban análisis de datos o visualizaciones por carrera o curso. Un ejemplo de una visualización generada por el sistema legado se puede ver en la sección A.2 del anexo.

Adicionalmente, la usabilidad y utilidad de la aplicación no había sido evaluada por los usuarios finales, y tampoco había garantías de que la información que le entregaba el sistema al docente fuera suficientemente valiosa, como para motivarlos a completar la encuesta una vez al año. Esto último, es decir, la repetitividad de la encuesta, es algo que se pretende lograr para poder analizar la evolución de la enseñanza de la ingeniería de software a lo largo del tiempo.

## Encuesta de curso: Ingeniería de Software III

### Información General del Curso

Barra de Progreso:



Responda esta sección de acuerdo a lo indicado en el programa del curso que se puede, considerando solo las horas de clase presencial. Puede responder con decimales, en cuyo caso utilice como delimitador decimal un punto. Las horas teóricas corresponden a la cantidad de horas de clases expositivas que posee el curso y las horas prácticas corresponden a la cantidad de horas de clases dedicadas a actividades de aplicación del conocimiento. Considere los promedios de ambas a lo largo del semestre.

?Cuántas semanas posee este curso?

Cantidad de semanas:

?Cuántas horas semanales de docencia directa posee este curso?

Horas Teóricas (ej: 2.XX):

Horas Prácticas (ej: 2.XX):

Figura 2.7: Extracto de interfaz de la primera parte de la encuesta.

## Encuesta de curso: Ingeniería de Software V

*Ir a:* Áreas de Enseñanza ▼ IR

**Áreas de Enseñanza**

**Barra de Progreso:**

Indique las áreas de la Ingeniería de Software abordadas por este curso. Además, indique el porcentaje del curso que se dedica a cada área y es necesario que los porcentajes sumen 100% para proseguir con la encuesta.

Nombre del Área	Descripción del Área	Porcentaje del curso
Fundamentos de la Computación	Esta área incluye los fundamentos de ciencias de la computación que apoyan el diseño e implementación de productos de software. Esto incluye el conocimiento necesario para convertir un diseño en una implementación, así como las técnicas y herramientas utilizadas durante este proceso.	% del Curso: _____ <input type="checkbox"/> No enseño esta área
Fundamentos de Ingeniería y Matemática	Esta área incluye los fundamentos matemáticos y de la ingeniería para la construcción de software de acuerdo a ciertos criterios deseados. Estos fundamentos ayudan a establecer un proceso de diseño predecible, siguiendo un proceso iterativo para modelar y razonar acerca de productos y las relaciones entre estos, haciendo un uso eficiente de los recursos disponibles.	% del Curso: _____ <input type="checkbox"/> No enseño esta área

Figura 2.8: Extracto de interfaz de la segunda parte de la encuesta.

## Encuesta de curso: Ingeniería de Software III

*Ir a:* Área: Fundamentos de la Com ▼ IR

**Área: Fundamentos de la Computación**

**Barra de Progreso:**

En esta sección, se debe responder, de acuerdo a lo que usted considere pertinente, cuánto enfoque le da a cierta unidad, tomando en cuenta los aspectos teóricos y prácticos.

Nombre y descripción de la Unidad	NO enseño esta unidad	Enfoque Teórico	Enfoque Práctico
<b>Fundamentos de Ciencias de la Computación</b> Incluye: Fundamentos de programación – Algoritmos, Estructura de Datos y Complejidad – Técnicas de Resolución de Problemas – Uso y Creación de Abstracciones – La Organización de un Computador – E/S, Mensajes de Error, Robustez – Estructura y Legibilidad de Código – Fundamentos de Lenguajes de Programación – Sistemas operativos – Bases de Datos – Redes de Comunicación.	<input type="checkbox"/>	Ninguno ○ ○ ○ ○ ○ Mucho	Ninguno ○ ○ ○ ○ ○ Mucho

Figura 2.9: Extracto de interfaz de la tercera parte de la encuesta.

# Capítulo 3

## Análisis y rediseño del sistema legado

Tomando los conceptos introducidos en el capítulo 2, y teniendo en cuenta los objetivos descritos en el capítulo 1, se llevaron a cabo diversas actividades para determinar las falencias del sistema legado, que impedían su puesta en producción y el logro de los objetivos planteados. Estas actividades fueron principalmente una evaluación de la usabilidad y utilidad de la aplicación a través de un focus group, una lluvia de ideas para identificar nuevos análisis a incorporar y sus visualizaciones correspondientes, y finalmente una revisión exhaustiva de toda la funcionalidad implementada.

La sección 3.1 muestra los resultados de la evaluación de la usabilidad y utilidad del sistema, la cual se hizo con académicos de la Universidad de Chile. Además, se detalla una primera exploración de los análisis a obtener a partir de los datos recuperados en la encuesta. El cálculo automático y representación apropiada de estos análisis es un requisito para posteriormente poder diseñar e implementar visualizaciones que motiven a los docentes a responder la encuesta. La sección 3.2 muestra los resultados de la exploración detallada de la solución actual, detallando las limitaciones de la arquitectura física, arquitectura lógica y del modelo de datos. Luego, se detallan los problemas más importantes para cada tópico, y se define el diseño (o rediseño) para solucionar cada uno de éstos. La sección 3.3 define el problema a abordar a través de los requisitos propuestos, mientras que en la sección 3.4 se presenta la nueva solución, es decir, la reingeniería del sistema legado.

### 3.1. Análisis de la usabilidad y utilidad del sistema legado

En esta sección se muestran los resultados del análisis de la usabilidad y utilidad del sistema legado, empezando con una presentación de la información obtenida a través del focus group.

#### 3.1.1. Focus group

Para estudiar la utilidad y usabilidad del sistema, se llevó a cabo un focus group con 5 académicos de la Universidad de Chile, que hacen docencia en el área de Ingeniería de Software:

Periodo 01/08/18 - 30/11/18	
Fecha de inicio	01/08/18
Fecha de termino	30/11/18
Numero de programas participantes	3
Numero de cursos participantes	7

Figura 3.1: Resumen de un periodo en específico.

Cecilia Bastarrica, Francisco Gutiérrez, Maíra Marques<sup>1</sup>, Sergio Ochoa y Jocelyn Simmonds. La dinámica de la actividad consistió en mostrar el sistema a los docentes y explicar cómo funciona, partiendo desde las visualizaciones que el sistema entrega luego de un periodo de encuesta, hasta las interfaces de usuario a las que acceden los profesores participantes. Por cada interfaz, los docentes dieron comentarios acerca del diseño, usabilidad y funcionalidad del sistema. A continuación, se presentan las principales críticas y/o sugerencias dadas por los académicos:

### Revisión de visualizaciones y estadísticas

Al ver las visualizaciones que se le entregan a los profesores, los académicos criticaron la utilidad de éstas, mencionando que no proveían suficiente información, ni motivación como para hacer que ellos respondieran la encuesta anualmente. La siguiente lista muestra varias sugerencias específicas acerca de las interfaces relacionadas a esta subsección:

- Mostrar nombres de programas y/o cursos que participaron en un periodo de encuestas en la vista de estadísticas.
- Además de tener las visualizaciones en el informe, dejar disponible la información en el sistema Web.
- Cambiar las leyendas y escalas de visualizaciones entregadas al docente, para que no hayan inconsistencias entre las visualizaciones, haciéndolas más legibles.

La figura 3.1 muestra un pequeño resumen de un periodo cualquiera, detallando la cantidad de programas y cursos que participaron en éste. Aquí se presenta el problema asociado a la primera sugerencia mencionada.

Las figuras 3.2 y 3.3 presentan dos ejemplos de las visualizaciones mostradas a los docentes. Los cuadrados azules contienen las leyendas de las visualizaciones, mientras que los cuadrados rojos destacan la escala de la segunda visualización. Aquí se presenta el problema asociado

<sup>1</sup>Al momento de escribir este documento, Maíra Marques ya no pertenece al DCC.

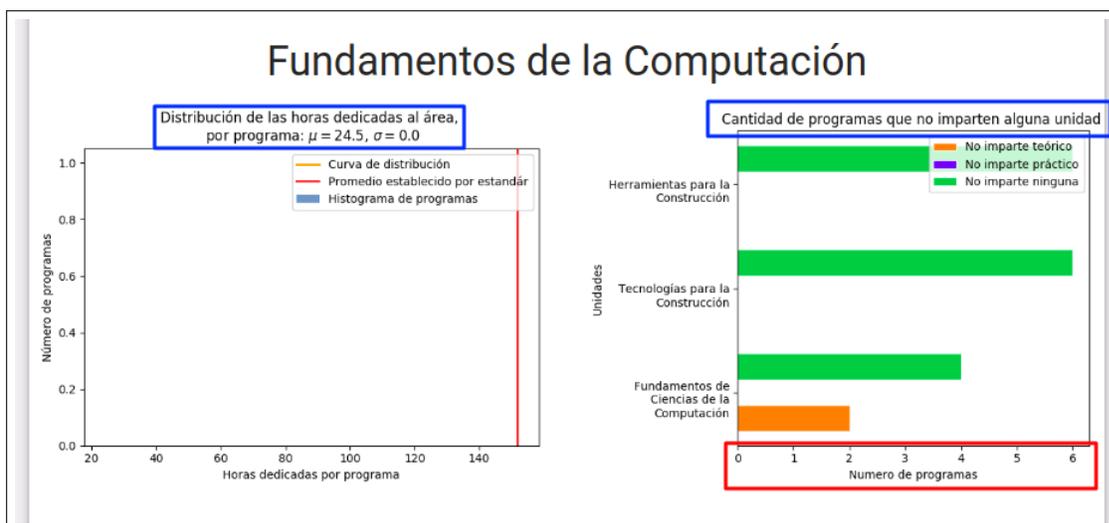


Figura 3.2: Visualizaciones para el área “Fundamentos de la Computación”.



Figura 3.3: Visualizaciones para el área “Práctica Profesional”.

a la tercera sugerencia mencionada.

## Revisión de las interfaces de la encuesta

Habiendo terminado la revisión de las visualizaciones y estadísticas, se continuó con el análisis de las interfaces de la encuesta. Las principales críticas a esta parte del sistema fueron: la longitud de la encuesta, la enorme cantidad de información que un docente debe leer para poder responderla (ver figura 2.9), y la ambigüedad de ciertos términos (o frases) que aparecen en la encuesta y que pueden inducir a interpretaciones erróneas. A continuación, se presentan las sugerencias que los académicos presentaron:

- Preguntar por la cantidad de alumnos promedio en un curso, en vez de la cantidad de alumnos anuales que pasan por dicho curso.
- Añadir un área “Otros” en el caso de cursos que también aborden tópicos/unidades que no estén incluidas en las que declara la ACM/IEEE.

- Enumerar las descripciones de áreas en la segunda y tercera parte de la encuesta, en vez de dejarlo en texto plano.
- Cambiar (o hacer más claro) el input requerido para el aspecto “enfoque teórico-práctico”, solicitado en la tercera parte de la encuesta.
- Agregar validaciones a la encuesta, tales como verificar que todos los campos sean respondidos, asegurándose que el docente siempre ingrese datos, y que dichos datos cumplan con validaciones mínimas.
- Dejar más claro que un docente puede navegar (hacia adelante y hacia atrás), y también editar las respuestas que ha ingresado en la encuesta.
- Cambiar la estética de ciertas partes de la encuesta para que se mantenga un estándar dentro del sistema.
- Evaluar la opción de separar el ítem que solicita la “cantidad de alumnos anualmente”, a la “cantidad de alumnos varones y mujeres”.
- Agregar pregunta acerca de los métodos o elementos de apoyo que usa un profesor para enseñar en sus clases; por ejemplo: uso de diapositivas, vídeos, entre otros.

### Otras sugerencias

Las siguientes sugerencias surgieron a medida que se exploraban las diversas secciones del sistema, las cuales no necesariamente están relacionadas a las anteriormente mencionadas:

- Mostrar la afiliación de los profesores en la interfaz de profesores asociados.
- Cambiar el formato del correo electrónico enviado a cada docente, al momento de invitarlos para que respondan la encuesta.
- Aclarar algunos conceptos que no son fáciles de entender, como por ejemplo “horas académicas” y “enfoque teórico-práctico”.
- Ver una opción de evitar el uso del *scroll* en el proceso de respuesta a la encuesta.

### 3.1.2. Lluvia de ideas acerca de los análisis

Finalizando el focus group, se desarrolló una lluvia de ideas, con los profesores Sergio Ochoa y Jocelyn Simmonds, con respecto a los análisis e información relevante que el sistema le debería proveer al docente que responde la encuesta. Estos análisis se dividieron en 3 categorías: análisis por curso, para programas, y para instituciones. A continuación, se muestran los análisis de datos discutidos en la lluvia de ideas. Para ejemplificar y ayudar a entender los análisis, se usarán los cursos Metodologías y Diseño de Programación (CC3002), Ingeniería de Software I (CC4401), Ingeniería de Software II (CC5401), y Proyecto de Software (CC5402):

#### Por curso:

- C1** - Comparación con otros cursos del mismo programa en términos de horas e información ingresada. Por ejemplo, se puede comparar el número de horas dedicadas a una área específica de conocimiento, por CC3002, CC4401, CC5401 y CC5402.
- C2** - Listado de cursos de otros programas parecidos al de un docente en particular. Por ejemplo, una lista de cursos de otros programas que sean parecidos a CC3002 en términos de áreas y unidades impartidas.

### **Por programa:**

- P1** - Contribución de áreas por cada curso del programa. Por ejemplo, con cuántas horas de docencia contribuyen los cursos CC3002, CC4401, CC5401 y CC5402 a los tópicos Requisitos y Testing.
- P2** - Comparación de un programa con las horas sugeridas por el Currículum Guideline de la ACM/IEEE. Esto consiste en sumar la cantidad de horas ingresadas por área y por cada curso de un programa, y luego compararlas con las horas sugeridas por lo propuesto en el Currículum Guideline de la ACM/IEEE.
- P3** - Progresión de horas teóricas y prácticas a través de un programa. Con esto se puede observar cómo las horas dedicadas a teoría y práctica evolucionan a medida que se sigue una línea temporal dentro del programa.

### **Por institución:**

- I1** - Cantidad de programas que imparten una cierta área de conocimiento, ya sea de manera teórica y/o práctica.
- I2** - Número de cursos de ingeniería de software obligatorios y electivos que posee cada programa.
- I3** - Distribución de todos los programas con respecto al Currículum Guideline de la ACM/IEEE, y a los programas de otras instituciones chilenas. Esto se refiere a ver aquellos programas que estén más cerca de lo recomendado por el Currículum Guideline, en términos de horas de docencia directa y tópicos impartidos por áreas.

Ciertos análisis requieren añadir un campo a la base de datos, y por tanto, a las interfaces de la encuesta: por ejemplo, que pregunten si el curso es obligatorio o no, y otro donde se especifique el semestre en el cual se dicta dicho curso según la malla del programa. Para los cursos electivos se pretende que la respuesta a esta pregunta sea el semestre donde los alumnos pueden empezar a tomar el curso. Esto se deberá incluir expresamente en el formulario de ingreso de datos.

## **3.2. Análisis detallado del sistema legado**

Habiendo realizado las actividades previamente mencionadas, se llevó a cabo una exploración más profunda de toda la funcionalidad implementada en el sistema legado. Cada apartado de esta sección presenta una descripción de los principales componentes del sistema, y se analizan sus limitaciones.

### **3.2.1. Arquitectura física**

Como se mencionó anteriormente, un usuario accede al sistema a través de un navegador web, donde la página corre en el servidor GEMS. Éste posee Debian 8 como sistema operativo, una arquitectura x86\_64, y el servidor de base de datos corre en Maria DB. La arquitectura física de la solución actual no presenta limitaciones que perjudiquen el desarrollo de nuevas

funcionalidades, debido a que sólo se está trabajando con universidades, programas, y cursos en Chile, los cuales se estiman que son 20 universidades, 25 programas, y 40 cursos según los datos recolectados en las entrevistas llevadas a cabo por los académicos del DCC. [6] Esto significa que el volumen de datos no es suficientemente grande como para requerir un cambio de arquitectura. Si es que se quiere extender la encuesta al extranjero, entonces es necesario cambiar la arquitectura para que el sistema pueda ser accedido simultáneamente por un número importante de personas. Sin embargo, el abordar este problema no es parte del alcance de esta memoria.

### 3.2.2. Arquitectura lógica

En la siguiente lista se detallan cada uno de los módulos de la capa de lógica presentados en la figura 2.4:

**MD1** - *Autenticación y Registro de Usuarios*: Este módulo se encarga de manejar el inicio y cierre de sesión de un administrador, del proceso de recuperación de contraseña si es que algún administrador se olvidó de ésta, y del registro de nuevos administradores.

**MD2** - *Envío de Invitaciones*: Éste es responsable de enviar los correos electrónicos a cada profesor que se le indique; el correo incluye los vínculos para responder sus encuestas correspondientes. Para ello, el sistema le provee al administrador una lista de profesores previamente invitados, de la cual puede escoger la cantidad de profesores que considere necesario.

**MD3** - *Asociación de Encuestas a Vínculos*: Este módulo es utilizado por el anterior para asignar a cada curso, en un periodo en específico, un enlace que le permite al profesor (o cualquier persona que tenga el enlace) ingresar a la encuesta para responderla. Este enlace se genera usando los datos del curso, profesor y periodo, lo que significa que cada enlace es único.

**MD4** - *Edición de Cuenta*: Este componente permite a los administradores editar información acerca de su cuenta. Esta información puede ser el correo electrónico o contraseña que usan para iniciar sesión dentro del sistema.

**MD5** - *Generación de Estadísticas y Reportes*: Este módulo se encarga de dos procesos muy importantes. El primero es la generación de visualizaciones según los datos de las encuestas de un periodo. Entre estas visualizaciones están las siguientes: los gráficos de barras que muestran la cantidad de programas que no dedican tiempo a la enseñanza de un área de conocimiento, y los gráficos de distribución de horas de docencia, según el número de programas [9]. El segundo proceso corresponde a la creación de un reporte que incluye las visualizaciones generadas anteriormente, y que es enviado a cada profesor que completó una encuesta con información acerca del curso correspondiente.

**MD6** - *Administración de Periodos*: Cada encuesta tiene asociado un periodo de tiempo en el cual se puede ser respondida. Este módulo se dedica a manejar y administrar estos periodos de tiempo, ofreciendo interfaces para comenzar y terminar un periodo. Además, se encarga de mostrar los resultados de las todas las encuestas de un periodo.

**MD7 - Exportación de Datos:** Este módulo se encarga de generar un archivo de *comma separated values* (CSV) que contiene la información bruta de todas las encuestas que fueron respondidas en su totalidad, en un periodo en específico.

**MD8 - Administración de Programas:** Permite al administrador ver, modificar, agregar o eliminar programas dentro del sistema. Cada carrera tiene asociada una ciudad, una institución y cursos específicos.

**MD9 - Administración de Encuestas:** Se encarga de mostrarle al docente la encuesta de su curso, asegurarse que las respuestas sean válidas, guardar los datos en caso de que el profesor no pueda seguir respondiendo la encuesta, y modificar la encuesta según las áreas de conocimiento escogidas por un profesor.

### Análisis de limitaciones

De todos estos módulos mencionados, el que presenta la mayor cantidad de limitaciones es el **MD5**. Esto se debe principalmente a dos razones. Primero, la librería usada para crear las visualizaciones, que en este caso es Matplotlib [11], exporta las imágenes de las visualizaciones y las guarda en una carpeta asociada a un periodo que llega a pesar 1MB. Esto significa que, si se quisiera añadir visualizaciones específicas para cada institución, carrera o curso que participaron en un periodo, se tendrían que añadir carpetas para cada una de estas entidades, lo que implica un problema de escalabilidad a largo plazo.

En segundo lugar, el reporte generado por este módulo no muestra información agregada de las encuestas, sino que sólo le muestra al docente la información que él o ella ingresó en la encuesta, más las visualizaciones mencionadas anteriormente. Esto no permite desarrollar un estudio para ver dónde se puede mejorar la enseñanza de la ingeniería de software de manera nacional, institucional, o por carrera, pues no se le está entregando al docente información que le puede ser útil. Dado eso, se vuelve necesario mejorar este módulo del sistema, de manera tal que entregue un informe donde muestren resultados procesados y útiles, y genere visualizaciones que no se tengan que exportar a una carpeta en particular.

### 3.2.3. Modelo de datos

En la siguiente lista se detallan cada una de las entidades del modelo de datos mostrado en la figura 2.5:

- *Institución:* Corresponde a las instituciones de educación superior de Chile. Incluye tanto establecimientos públicos como privados. Dentro de esta tabla se guarda el nombre de la organización correspondiente. Una institución puede tener varias carreras.
- *Ciudad:* Cada fila de esta tabla representa una ciudad dentro de Chile. En esta tabla se guarda el nombre de la ciudad, y al igual que una institución, esta tabla puede tener varias carreras o programas asociados.
- *Carrera:* Representa un programa o un conjunto de estudios, repartidos en cursos, que capacitan para ejercer una profesión de computación. Una carrera pertenece a una sola institución y ciudad, a la vez de tener una cantidad determinada de cursos. En el campo se guarda el nombre del programa, la institución y ciudad relacionada, si está activo o no, y además el tipo de carrera. Sin embargo, no se guarda información acerca de la

relación entre cursos de una carrera.

- *Curso*: Un curso representa un conjunto de clases sobre algún tópico en específico de la ingeniería de software que sigue una estructura fija, especialmente dentro de una carrera u programa. Un curso tiene un profesor encargado de responder la encuesta, y puede tener varias encuestas asociadas. Dentro de este campo se guarda el profesor, la carrera asociada, el nombre del curso y su estado.
- *Profesor*: Esta es la persona encargada de enseñar un curso asociado al ámbito de ingeniería de software en el marco de una carrera. Un profesor puede tener asociado varios cursos, pues éste puede dictar más de uno en la misma institución. En esta tabla se guarda el nombre y correo electrónico del profesor.
- *Encuesta*: En esta tabla se guarda parte de la información ingresada en la encuesta. Datos como las semanas de clase, las horas semanales teóricas y prácticas, entre otros, son almacenados aquí para luego ser procesados por el sistema. Solamente puede existir una encuesta por curso en un periodo definido, y ésta puede poseer varias áreas detalladas en la segunda parte de la encuesta.
- *Periodo*: Un periodo es un intervalo de tiempo, con fecha de inicio y término, donde se puede responder una encuesta. Luego de que finaliza el intervalo, éste deja de estar activo. Por último, un periodo tiene asociado un estándar, que en este caso es el Currículum Guidelines 2014 elaborado por la ACM/IEEE.
- *Estándar ACM*: Esta tabla almacena el Currículum Guidelines 2014 elaborado por la ACM/IEEE, particularmente las áreas y unidades de conocimiento propuestas en el documento. Un estándar o guía de referencia puede tener varios periodos de encuesta asociados.
- *Área Conocimiento*: Corresponde a una subdisciplina de la ingeniería de software detallada en el Currículum Guidelines 2014. Esta área posee una cantidad de horas específica que corresponde a las horas detalladas por el estándar correspondiente. Un área tiene varias unidades relacionadas, jerarquía que se explicó anteriormente.
- *Unidad Conocimiento*: Son los módulos temáticos en los que se separa un área de conocimiento. Al igual que éstas, poseen una cantidad de horas prácticas y teóricas que un docente debería dedicar a esas unidades.
- *Detalle Horas Área*: Corresponde a las áreas de conocimiento que fueron indicadas por un profesor en una encuesta en particular. Aquí se detalla el porcentaje que un profesor dedica de su curso en las áreas detalladas en el Curriculum Guideline 2014 elaborado por la ACM/IEEE. Un área detallada puede poseer múltiples unidades de conocimiento.
- *Detalle Horas Unidad*: Aquí se guardan los datos ingresados en la tercera parte de la encuesta, almacenando el enfoque teórico y práctico que los profesores le dan a sus cursos.

## Análisis de limitaciones

Una de las cosas que no se consideraron al momento de desarrollar el proyecto durante el curso de Ingeniería de Software II, fue agregar un campo que guarde las regiones a las que cada ciudad pertenece. Esto sirve para poder hacer estudios a nivel de región, ya sea la cantidad de horas que una región le dedica a la enseñanza de un área de conocimiento, o la cantidad de alumnos que egresan de carreras de ingeniería de software de esa región.

Periodos		Programas		Invitaciones
INICIAR NUEVO PERIODO ⊕		AÑADIR ESTÁNDAR ⊕		
Inicio	Término	Detalles	Estadísticas	Exportar
17/04/2019	20/04/2019			
01/04/2019	10/04/2019			
05/12/2018	01/04/2019			

Figura 3.4: Interfaz de usuario del administrador, que muestra los periodos completados y activos.

Periodos		Programas		Invitaciones
<input type="text" value="Buscar Programa"/>		Todas las Instituciones ▼		AÑADIR PROGRAMA ⊕
Editar	Nombre Programa	Ciudad	Institución	Cursos
	Ingeniería Civil en Informática	Valdivia	Universidad Austral de Chile	Ingeniería de Requisitos Ingeniería de Software Métodos y Modelos de Ingeniería de Software
	Ingeniería Civil en Informática	Concepción	Universidad del Bío-Bío	Sistemas de Información Ingeniería de Software Gestión de Proyectos de Software

Figura 3.5: Interfaz de usuario que muestra las carreras ingresadas en el sistema.

Otro problema que nace del modelo de datos del sistema legado, es la falta de entidades que guarden los datos necesarios para llevar a cabo estudios pertinentes. Por ejemplo, no existe una tabla que permita ordenar los cursos de una carrera por orden de semestre. Otro ejemplo es que, a pesar que la encuesta está pensada para cualquier curso de ingeniería de software, incluso los cursos que no pertenecen a la malla (electivos), actualmente no hay una forma de diferenciar entre cursos obligatorios y electivos.

### 3.2.4. Interfaces del sistema

Dentro del sistema, se encuentran 34 interfaces de usuario distintas, pero las que se muestran en esta sección son las relevantes para el trabajo de la memoria. En esta sección se hablarán de las limitaciones interfaces explicadas en la sección 2.2.5, y de las interfaces del administrador que serán explicadas en la siguiente subsección.

Periodos		Programas			Invitaciones	
<input type="text" value="Buscar Docente"/>		Todas los programas <span>▼</span>			<input type="button" value="AÑADIR DOCENTE"/>	
					<input type="button" value="AÑADIR ADMINISTRADOR"/>	
					<input type="button" value="MODIFICAR MIS DATOS"/>	
<input type="checkbox"/>	Nombre Docente	Universidad	Programa y curso	E-Mail	Invitado a periodo actual	Editar usuario
<input type="checkbox"/>	Juan Álvarez	Universidad Austral de Chile	Ingeniería Civil en Informática - Ingeniería de Requisitos	jalva@educais.tk	NO	<input type="button" value="✎"/>
<input type="checkbox"/>	Iván Torres	Universidad Austral de Chile	Ingeniería Civil en Informática - Ingeniería de Software	torres.ivan@educais.tk	NO	<input type="button" value="✎"/>
<input type="checkbox"/>	Felipe Labarca	Universidad Austral de Chile	Ingeniería Civil en Informática - Métodos y Modelos de Ingeniería de Software	alexis.espinoza@ug.uchile.cl	NO	<input type="button" value="✎"/>

Figura 3.6: Interfaz que muestra los profesores ingresados en el sistema.

## Interfaces de usuario del administrador

En la figura 3.4 se muestra una de las interfaces principales de los administradores. Cuando un administrador inicia sesión en el sistema, se le presenta una tabla que contiene todos los periodos que se han creado. Para cada periodo se muestra su fecha de inicio y término (o un botón que termina el periodo), los detalles de ese periodo, las estadísticas o datos agregados, y un botón que permite descargar los datos en bruto del periodo.

La figura 3.5 muestra la segunda opción del menú de administradores, la cual es es ver las carreras que se encuentran ingresadas en el sistema. Para cada carrera, se detalla la institución a la que pertenece, los cursos que posee, y una opción que permite editar información acerca del programa (nombre, institución, cursos, etc).

Por último, la figura 3.6 muestra la tercera opción del menú de administradores, la cual le permite a un administrador ver los profesores ingresados en el sistema, con sus cursos correspondientes, y las invitaciones que se han mandado en el periodo actual. Un administrador puede enviar o cancelar una invitación, y puede añadir o eliminar un profesor.

## Análisis de limitaciones

Al momento de hacer la exploración, uno de los problemas que se encontraron fue el uso de métodos GET para obtener información y cambiar las interfaces según ésta. Esto ocurría al momento de hacer click en los detalles de un periodo, los detalles de una encuesta en específico, o las estadísticas de un periodo. Si bien esto no es una limitación en si, es una mala práctica que puede ser usada para obtener información que no debería ser pública para ciertos usuarios. Además, al momento de actualizar una interfaz que fue accedida a través de una llamada GET, la aplicación lanzaba un error. Esto se debe a que, al actualizar la página, la llamada GET no posee las variables que usó en la primera llamada para acceder a los datos correspondientes.

Analizando las interfaces de la encuesta, se encontraron algunas inconveniencias menores. Por ejemplo, la figura 2.6 posee problemas de formato, lo que la hace mucho más difícil de leer y entender. Esto puede desalentar a los profesores a responder la encuesta, dado que los términos y condiciones son fundamentales para conseguir el consentimiento del usuario [12].

Dentro de las interfaces de la encuesta (figuras 2.7, 2.8, 2.9), se notó un problema en el uso de colores aleatorios al momento de diseñar la página, lo que puede causar problemas dado que el color juega un papel fundamental en la percepción visual de un usuario en cualquier sistema [13].

### 3.3. Requisitos de la solución

Habiendo descrito los problemas y limitaciones mencionadas en la sección 3.1 y 3.2, se puede deducir que la mayoría de los problemas asociados al sistema legado caen en la categoría de “baja utilidad de las visualizaciones” entregadas a los profesores, y la falta de datos en el reporte final, que permita a quienes lo lean sacar conclusiones acerca de la enseñanza de la ingeniería de software a nivel nacional, institucional, entre otros ámbitos. Dicho esto, la reingeniería del sistema legado deberá apuntar al logro de los requisitos que se indican a continuación:

#### 3.3.1. Requisitos funcionales

- R1** - Un usuario cualquiera debe poder descargar un reporte nacional con datos agregados para un cierto periodo de la encuesta.
- R2** - Un usuario administrador debe tener la posibilidad de descargar reportes con información con cualquiera de los tres niveles de agregación indicados: de nivel nacional, por institución y por programa.
- R3** - Un usuario docente debe recibir un reporte y poder consultar la información acerca de las carreras donde imparte cursos de ingeniería de software, una vez que haya respondido la encuesta correspondiente.
- R4** - Un usuario docente debe tener la posibilidad de ver estadísticas generadas con los datos de su carrera.
- R5** - Un usuario docente representante de una institución, debe tener la posibilidad de ver estadísticas interactivas generadas con los datos de las carreras e institución que representa.

Por un lado, el origen de los requisitos **R1**, **R2** y **R3** viene de las reuniones realizadas a través del semestre con los profesores guías, donde se estableció la importancia del reporte generado por el sistema. Por el otro, el origen de los requisitos **R4** y **R5**, vienen del memorista, y surgieron una vez terminada exploración detallada del estado actual de la solución. Para estos requisitos, la prioridad es alta, y los usuarios que se ven afectados por estos son los profesores que participan en un periodo de encuestas y los administradores del sistema.

### 3.3.2. Requisitos no funcionales

**RN1** - El sistema no debe ser usar imágenes para mostrar las visualizaciones generadas por éste.

**RN2** - Las visualizaciones deben permitir interacción con el usuario.

El objetivo de los requisitos no funcionales es mejorar el sistema para que sea lo suficientemente seguro, confiable y robusto como para ponerse en producción.

Al igual que los requisitos funcionales, las prioridades de los no funcionales son altas, y los usuarios afectados por estos requisitos son los profesores que responden la encuesta y los administradores del sistema. Estos requisitos permiten que el sistema pueda entregar información útil a los profesores que respondan la encuesta, sin tener que almacenar información que pueda crecer a volúmenes muy grandes. Solucionando ese problema, se hace más fácil agregar gráficos que puedan ser útiles, y que motiven al docente a responder la encuesta, ya que no sería necesario guardar imágenes de visualizaciones para una carrera específica en un periodo específico, lo que significa que los profesores reciben una motivación más grande para responder la encuesta.

## 3.4. Rediseño realizado

En las siguientes subsecciones se explican las modificaciones hechas al modelo de datos y a la arquitectura lógica del sistema legado, para poder soportar los requerimientos mencionados en la sección anterior. Cabe destacar que, debido a que la arquitectura física no presentaba limitaciones relacionadas a su funcionamiento, no se menciona ese aspecto en esta sección, pues no se hicieron cambios a este aspecto en el sistema legado.

### 3.4.1. Arquitectura lógica

En la subsubsección 3.2.2. se mencionó que el módulo MD5 (Generación de Estadísticas y Reportes) presentaba la mayor cantidad de problemas en términos de escalabilidad, utilidad del reporte generado, y sobrecarga de funcionalidades. Para solucionar esto, se separó el módulo **MD5** en dos para asegurar que la generación de estadísticas y generación del reporte entregado a los profesores se hagan por módulos distintos, y así especializar cada solución. La figura 3.7 ilustra este cambio.

En la siguiente lista se explican los nuevos módulos agregados a la arquitectura lógica y sus funcionalidades:

**MD5** - *Generación de Estadísticas*: Este módulo se encarga de generar las visualizaciones interactivas para las carreras que participen en un periodo de encuestas específico, asociando una carrera y un periodo, con un vínculo que le permite a los profesores de esa carrera ingresar al sistema para ver las visualizaciones correspondientes.

**MD10** - *Generación de Reportes*: Este componente engloba a todas las funcionalidades asociadas a la generación de reportes entregados a los profesores luego de un periodo de encuestas. También cuenta con un submódulo que genera reportes de información a

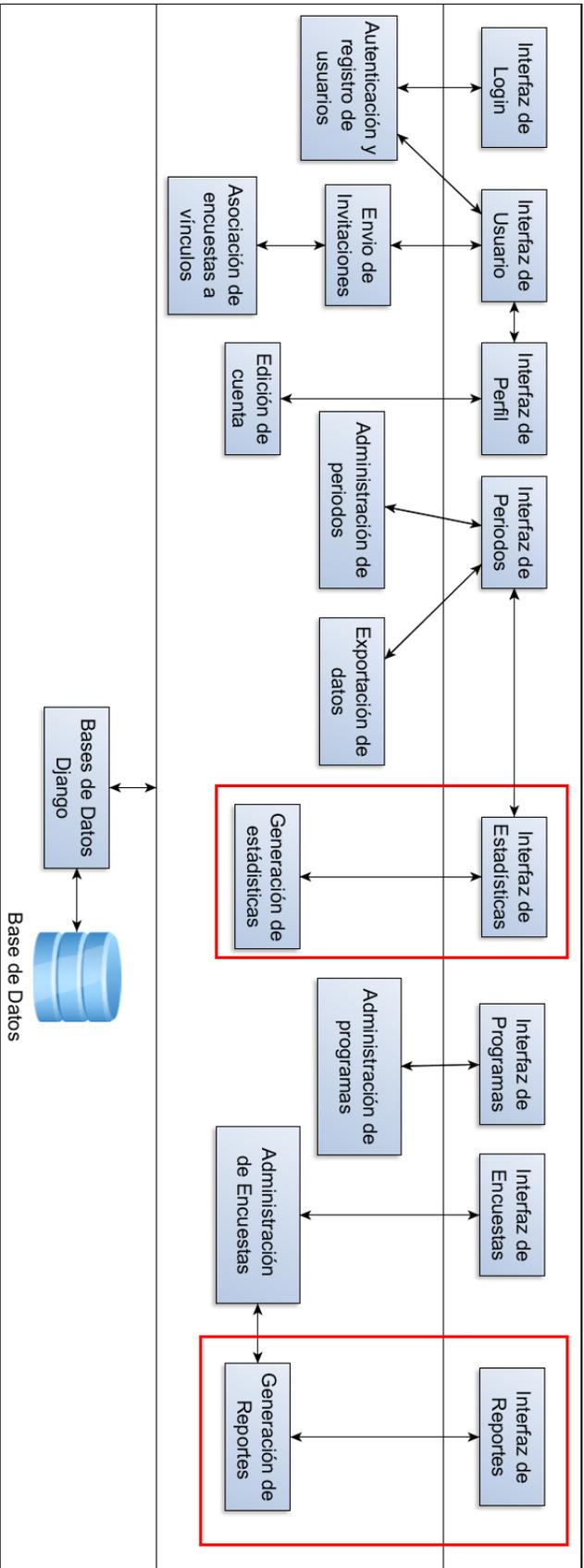


Figura 3.7: Arquitectura lógica del nuevo sistema. Los rectángulos en rojo muestran los módulos agregados en la solución.

nivel nacional e institucional. Por último, cuenta con un componente que se encarga de generar dinámicamente plantillas de tablas en LaTeX.

Separar el módulo de generación de estadísticas y reportes **MD5** en dos, nos asegura que la generación de visualizaciones no estará sujeta a la generación del reporte, lo que hace más fácil la mantención del sistema y la detección de posibles errores al momento de llevar a cabo alguna de estas funcionalidades. Además, esta separación permite agregar nuevas operaciones al módulo **MD10** como, por ejemplo, generar otros tipos de reportes usando la información de un periodo en específico.

Al igual que en el capítulo 2, las dependencias entre módulos fueron determinadas a través de un análisis de las declaraciones *import* dentro de los archivos de código del sistema.

### 3.4.2. Modelo de datos

Si bien ya existía un modelo de datos creado por las iteraciones anteriores, se determinó que para soportar los requisitos mencionados anteriormente, se debían agregar nuevas tablas o atributos a las tablas existentes. La figura 3.8 muestra el nuevo modelo de datos, donde las tablas y atributos con un borde rojo representan las cosas agregadas. Es importante destacar que, como parte de esta memoria, toda la funcionalidad implementada por el sistema legado debía mantenerse, pero ahora considerando el nuevo modelo de datos.

En la siguiente lista se detallan las tablas que fueron sujetas a cambios y que se añadieron al modelo anterior:

- *Encuesta*: Como se mencionó anteriormente, esta tabla guarda parte de la información ingresada en una encuesta. A esta tabla se le agregaron 3 atributos: uno que guarda los créditos transferibles de un curso (SCT), otro que detalla si el curso es obligatorio en la malla de la carrera respectiva, y uno que guarda el semestre cuando se debe tomar el curso.
- *Ciudad*: Para poder hacer estudios a nivel de región, se le agregó a esta tabla uno atributo que almacena la región en la cual está situada la ciudad donde se imparte la carrera.
- *Resultados de Carrera*: Esta nueva tabla asocia una carrera con un periodo en el cual uno de sus cursos fue parte de la encuesta. Dentro de esta tabla, además, se guarda una serie de caracteres que funcionan como identificadores (id) de esta asociación. Usando este id, se puede acceder a las visualizaciones de la carrera en específico para un cierto periodo. Este id siempre es válido, por lo que se puede acceder sin importar el periodo de encuesta actual del sistema.
- *Resultados de Institución*: Esta nueva tabla asocia una institución con un periodo en el cual una de sus carreras participó en la encuesta. Similar a la tabla anterior, ésta genera un id que sirve para poder acceder a visualizaciones de la institución en específico para un cierto periodo.

Agregando la tablas de resultados por carrera e institución, nos aseguramos de que se pueda tener un vínculo para acceder a las visualizaciones específicas a una carrera o institución, en un periodo en específico. Esto además permite que exista una línea temporal donde los

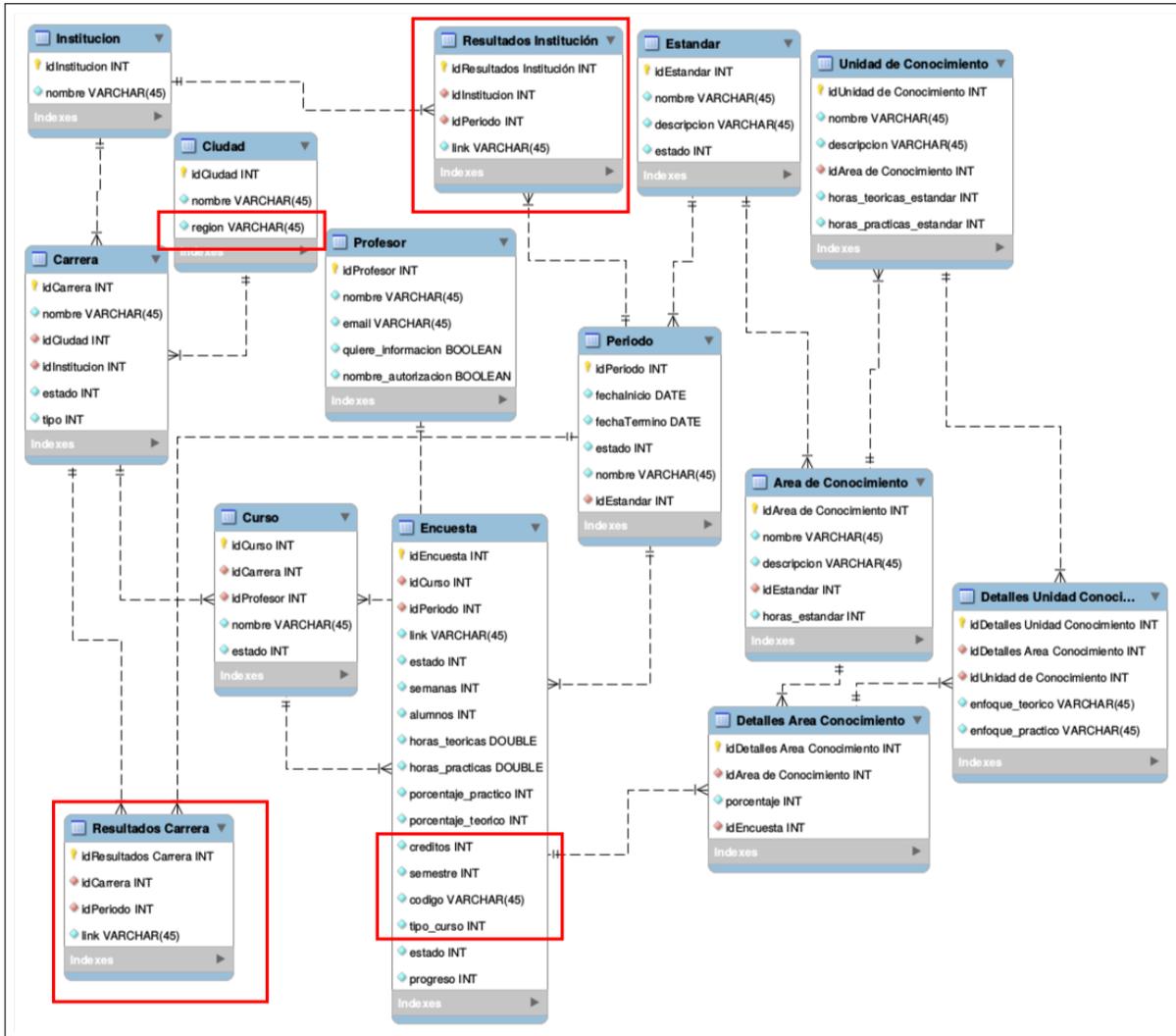


Figura 3.8: Nuevo modelo de datos de la solución.

profesores pueden ver la evolución de sus carreras a través del tiempo.

Los atributos añadidos a las tablas de Ciudad y Encuesta permiten guardar más información acerca de la enseñanza de la ingeniería de software en Chile, lo que implica que se pueden llevar a cabo otro tipo de estudios, tales como la cantidad de horas dedicadas a la enseñanza de una cierta área en una región, o la cantidad de créditos transferibles promedio a nivel nacional.

### 3.5. Resumen

En esta sección se mostraron los resultados del focus group realizado con los profesores del Departamento de Ciencias de la Computación de la Universidad de Chile, de la lluvia de ideas para identificar los análisis y visualizaciones generados a partir los datos recolectados por la encuesta, y del análisis detallado del sistema, detallando además las limitaciones de cada componente de éste. Con esos resultados, se definieron los problemas a abordar en esta reingeniería, y se presentó la solución a estos a través de los requisitos funcionales y no funcionales. Además, se presentaron los ajustes realizados al modelo de datos y a la

arquitectura lógica para que soporten los nuevos requisitos.

En los siguientes capítulos, se explican y detallan las decisiones tomadas al momento de llevar a cabo la implementación de los requisitos establecidos en esta sección. Por último, se muestran los resultados de la validación de las nuevas funcionalidades añadidas a la solución actual.

# Capítulo 4

## Solución desarrollada

Tomando como base lo señalado en el capítulo 3, se comenzó el desarrollo de la nueva solución. Este capítulo presenta las distintas soluciones implementadas para mejorar el funcionamiento del sistema legado. En cada sección se explica y justifica el uso de las tecnologías involucradas en la implementación de la nueva solución. Particularmente, la sección 4.1 detalla qué análisis se escogieron de la lista de ideas, el por qué, su implementación y la información que se quiere transmitir. La sección 4.2 habla acerca de los cambios al reporte generado por el sistema, y cómo se implementaron éstos. En la sección 4.3 se especifican los cambios realizados en las interfaces de la encuesta y en el panel de control de los administradores. Por último, en la sección 4.4 se presentan cambios que no están directamente relacionados con lo realizado en las secciones anteriores.

### 4.1. Análisis implementadas

Lo primero que se implementó dentro del sistema fueron los análisis, y por tanto, las visualizaciones que mostraran éstas. A continuación, se explica el análisis y las decisiones tomadas al momento de abordar esta problema.

#### 4.1.1. Análisis de la lluvia de ideas

Lo primero que se hizo antes de implementar los análisis de datos fue revisar los resultados de la lluvia de ideas realizada sobre los análisis a implementar (ver sección 3.1.2). Luego, se escogió cuáles de éstas eran factibles de implementar con los datos obtenidos desde las encuestas, y se dieron ideas de visualizaciones que pudieran ser útiles para representar dichos análisis. Esta revisión se llevo a cabo para priorizar las necesidades, pues no era posible implementar todas los análisis de datos sin perjudicar el desarrollo de los otros requisitos. Es por eso que los análisis que se decidieron implementar fueron las siguientes:

**M1** - Evolución de los alumnos egresados de una carrera específica, comparada con la cantidad de alumnos egresados en promedio a nivel país. Esto permite ver qué tantos alumnos están saliendo a la industria del software por carrera.

**M2** - Progresión de horas semanales teóricas y prácticas a través de un programa. Con esto

se puede observar cómo las horas dedicadas a la enseñanza teórica y práctica de la disciplina, evolucionan a medida que se sigue una línea temporal dentro del programa. Esta visualización corresponde al análisis de datos **P3** de la lluvia de ideas.

**M3** - Comparación de las horas totales dedicadas a la enseñanza de las áreas de conocimiento establecidas por el Currículum Guidelines 2014, entre una carrera específica, el promedio a nivel de carrera, y las horas recomendadas por el Currículum Guidelines 2014. Esto nos permite ver cómo una carrera enseña las distintas áreas de conocimiento propuestas por la recomendación de la ACM/IEEE, en comparación al currículum y al promedio de horas totales a nivel de las carreras que participaron en la encuesta.

**M4** - Comparación del nivel de dedicación a la enseñanza de las áreas de conocimiento establecidas por el Curriculum Guidelines, entre los cursos de una carrera en específico. Este análisis nos permite ver en qué áreas de conocimiento se enfocan los cursos de un programa en específico, y también si allí se hace foco en aspectos teóricos o prácticos del aprendizaje de los alumnos.

#### 4.1.2. Revisión de librerías

Finalizada la definición de los análisis de datos, se revisó la herramienta usada para crear las visualizaciones que se encontraban en el sistema legado (Matplotlib [11]), eso con el fin de determinar la posibilidad de su uso en las nuevas visualizaciones. A través de un análisis detallado, se llegó a la conclusión de que la herramienta Matplotlib no permitiría cumplir con el primer requerimiento no funcional **RN1** mencionado en la sección 3.3. Esto se debe a que, para crear las visualizaciones y mostrarlas a un usuario, esta librería debía exportar las imágenes y dejarlas guardadas en una carpeta, lo que a largo plazo implica un problema de escalabilidad muy grande. Además, si es que alguna de esas imágenes se eliminara del sistema, no habría forma de recuperarla dado a que el sistema genera las imágenes para un periodo solo una vez.

Por el problema mencionado anteriormente, se decidió eliminar el uso de la herramienta Matplotlib dentro del sistema. Después de eso, se comenzó a ver qué librerías se podían usar para visualizar la información de las encuestas y resolver el problema de escalabilidad. El criterio principal para escoger una librería fue que no fuese necesario exportar imágenes para poder mostrarlas en el sistema. Según esto, se escogió la librería **Chart.js** [14], escrita en el lenguaje Javascript.

Esta librería permite generar visualizaciones interactivas en el front-end de la aplicación, lo que implica que el back-end solamente debe entregar los datos necesarios para poder crear la visualización, y no exportar imágenes para que puedan ser vistas por alguno de los usuarios. En la figura 4.1 se puede ver un ejemplo de un gráfico hecho con Chart.js.

#### 4.1.3. Visualizaciones a usar

Luego de escoger las librerías, se eligió un tipo de visualización para cada análisis de datos en particular. A continuación, se explica la razón de la elección de cada tipo de visualización para cada análisis de datos:

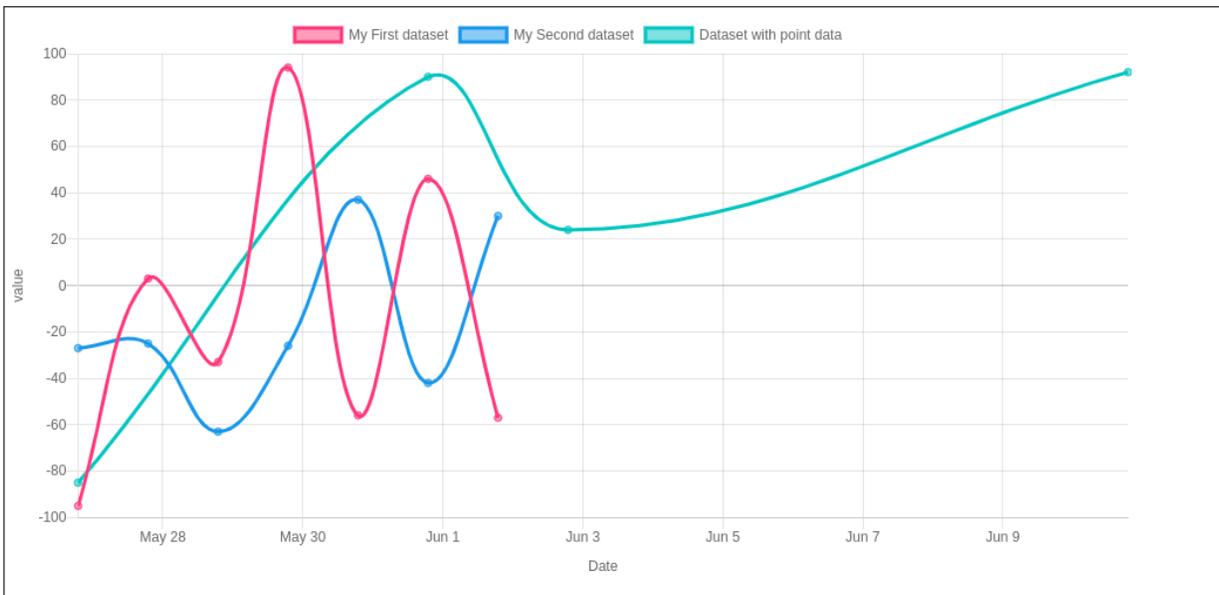


Figura 4.1: Ejemplo de visualización usando Chart.js.

- Para el análisis **M1** se decidió usar un gráfico de barras adyacentes. Esto se debe a que se quiere mostrar una progresión de dos variables discretas (horas de docencia teóricas y prácticas) a través del tiempo (avance de los cursos de una carrera).

Si bien se podría haber usado un gráfico de líneas múltiples para mostrar este análisis, se consideró más sencillo usar un gráfico de barras debido a su simplicidad al momento de comparar dos valores, a diferencia de un gráfico de líneas donde éstas se pueden cruzar. El eje X corresponde a los cursos de una carrera en específico, mientras que el eje Y corresponde a la cantidad de horas directas. La barra azul corresponde a las horas de docencia teóricas, y la barra naranja indica las horas de docencia prácticas.

La figura 4.2 muestra la implementación de esta visualización.

- El análisis **M2** se muestra a través de un gráfico que combina barras y líneas. Similar al análisis de datos **M1**, se están comparando dos variables discretas: las cantidad de alumnos que egresan de la carrera, versus promedio de alumnos que egresan a nivel de país. Esto se muestra a través del tiempo; es decir considerando los periodos (años) en los que se realizaron las encuestas.

La diferencia entre esta visualización y la usada en el análisis **M1** es que, en ésta, se está trabajando con los periodos de encuesta. Si es que se usara un gráfico de barras adyacentes, las barras se verían muy pequeñas a medida que se creen más periodos, dado el espacio limitado donde se encuentra el gráfico. Como se usa un gráfico de líneas y barras, la línea no toma espacio adicional, lo que la hace mejor en términos del espacio ocupado. El eje X corresponde a los distintos periodos de encuestas, mientras que el eje Y indica la cantidad de alumnos egresados. El color azul representa los alumnos egresados de la carrera en particular, y el color naranja corresponde a los alumnos egresados en promedio a nivel de país.

La figura 4.3 muestra la implementación de esta visualización.

- Se decidió utilizar un gráfico radar para visualizar este análisis de datos **M3**. Esto se debe a que se requiere comparar múltiples variables cuantitativas (horas dedicadas a la

enseñanza de las áreas de conocimiento) a nivel de dos entidades (carrera y promedio del país), mostrando qué variables poseen un valor alto o bajo. Dado a que el Curriculum Guidelines 2014 de la ACM/IEEE recomienda horas distintas para cada área de conocimiento, se decidió normalizar los datos y comparar el porcentaje de horas dedicadas a la enseñanza de un área de conocimiento en comparación a las horas establecidas por el Currículum Guidelines 2014. Esto asegura tener valores entre un rango de 0 y 100 %.

La alta cantidad de variables hace difícil comparar este tipo de valores de otra forma. Además, ya que se quiere ver si se enseña más o menos horas que las recomendadas por el Currículum Guideline 2014 de la ACM/IEEE, es necesario tener algún tipo de referencia al 100 %, que en este caso son los límites de cada eje.

Cada eje de la visualización corresponde a un área de conocimiento del estándar utilizado, para cada encuesta y periodo. Los puntos sobre los ejes indican el porcentaje de horas que una carrera le dedica a la enseñanza de esa área, comparada con las horas recomendadas por el Currículum de Referencia. El color azul representa la carrera en particular, y el color naranja representa el promedio a nivel de país.

La figura 4.4 muestra la implementación de esta visualización.

- Para el análisis **M4**, se utilizó una visualización de cuatro cuadrantes. En este caso, lo que se quiere lograr es ver la dedicación que los cursos de una carrera le da a la enseñanza de una o más áreas de conocimiento. Se quiere, además, clasificar estas dedicaciones en cuatro categorías: teórico-práctico, teórico-no práctico, no teórico-práctico, y no teórico-no práctico. El eje X representa la dedicación a la enseñanza teórica de un área de conocimiento, mientras que el eje Y indica la dedicación a la enseñanza práctica de un área de conocimiento.

La figura 4.5 muestra la implementación de esta visualización. Los colores usados en el gráfico representan las distintas áreas de conocimiento, y cada punto en el plano simboliza un curso de la carrera en particular.

El último paso antes de implementar las visualizaciones fue agregar código al módulo de encuestas para que, cuando un profesor terminase su encuesta, se asociara su carrera con el periodo en el cuál estaba participando. De esta manera, se crea el vínculo para poder ingresar a las visualizaciones. La figura 4.6 muestra un extracto del código que hace esto posible. Este extracto busca si existe algún objeto *CareerResults* y *InstitutionResults* donde esté asociada la carrera con el periodo de encuestas actual. Si es que no lo encuentra, crea un identificador único que toma el nombre del periodo, el nombre de la carrera (o institución), y la fecha en la que se hizo la operación. Por último, la aplicación toma este identificador, la carrera (o institución) y periodo correspondiente, y los guarda dentro de las tablas correspondientes.

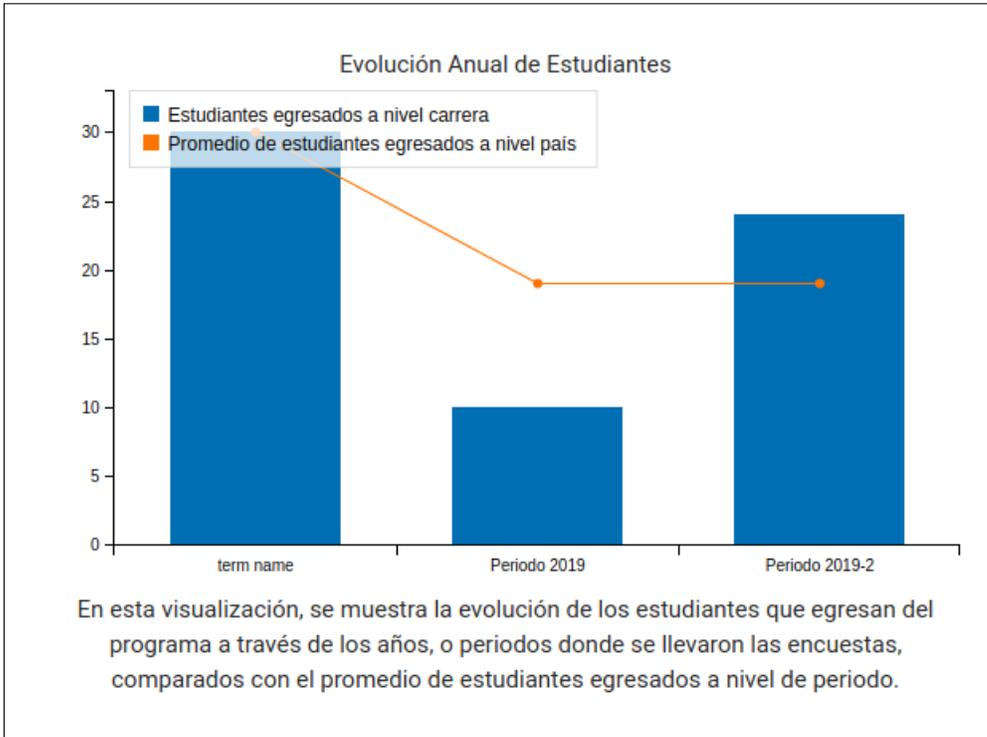


Figura 4.2: Visualización del análisis **M1** implementada en el sistema.

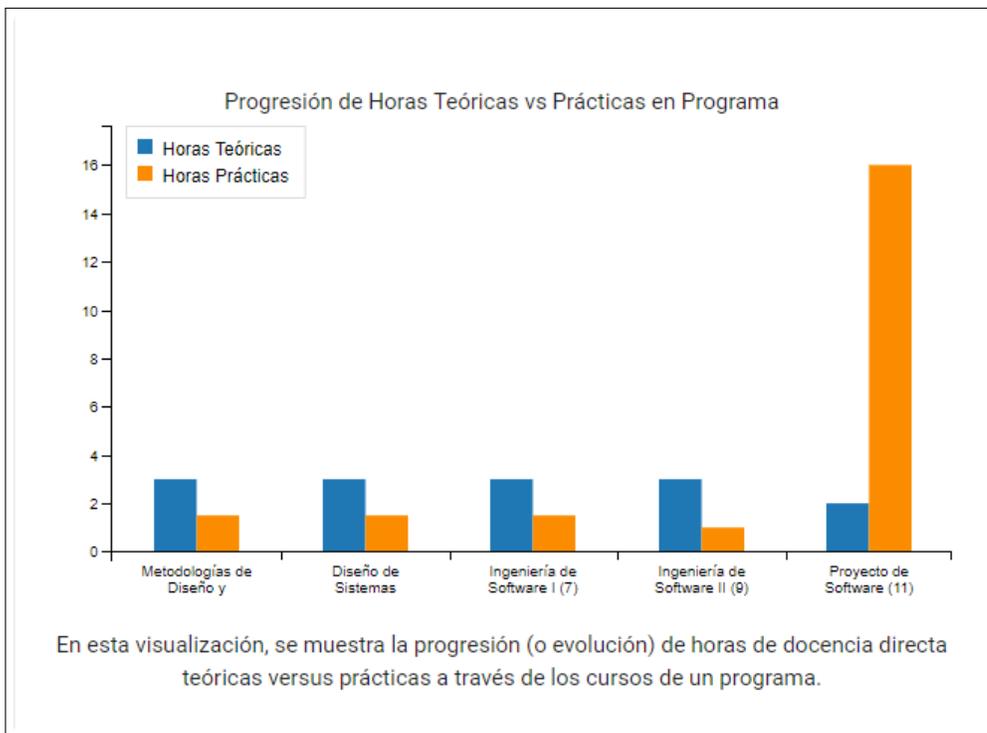


Figura 4.3: Visualización del análisis **M2**.

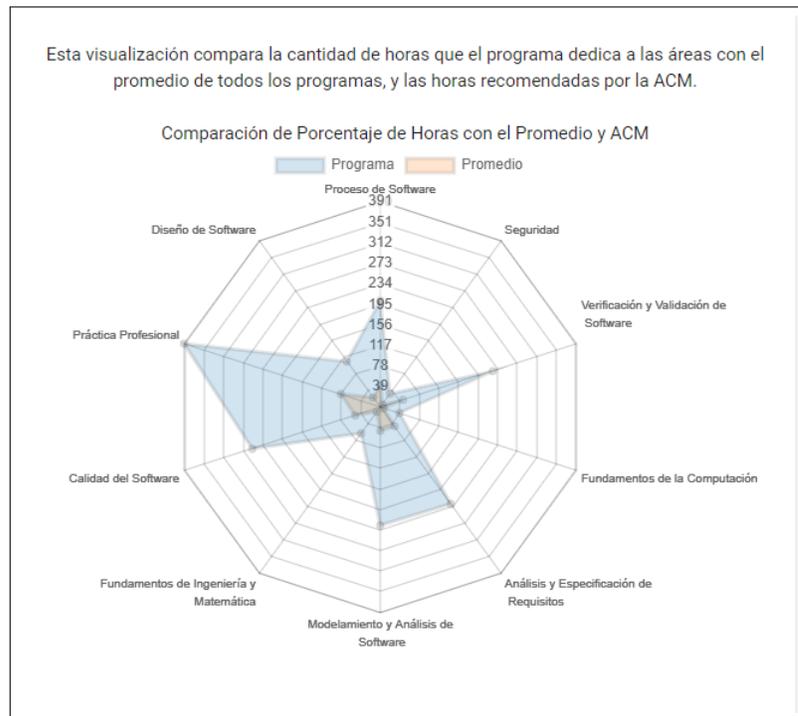


Figura 4.4: Visualización del análisis M3.

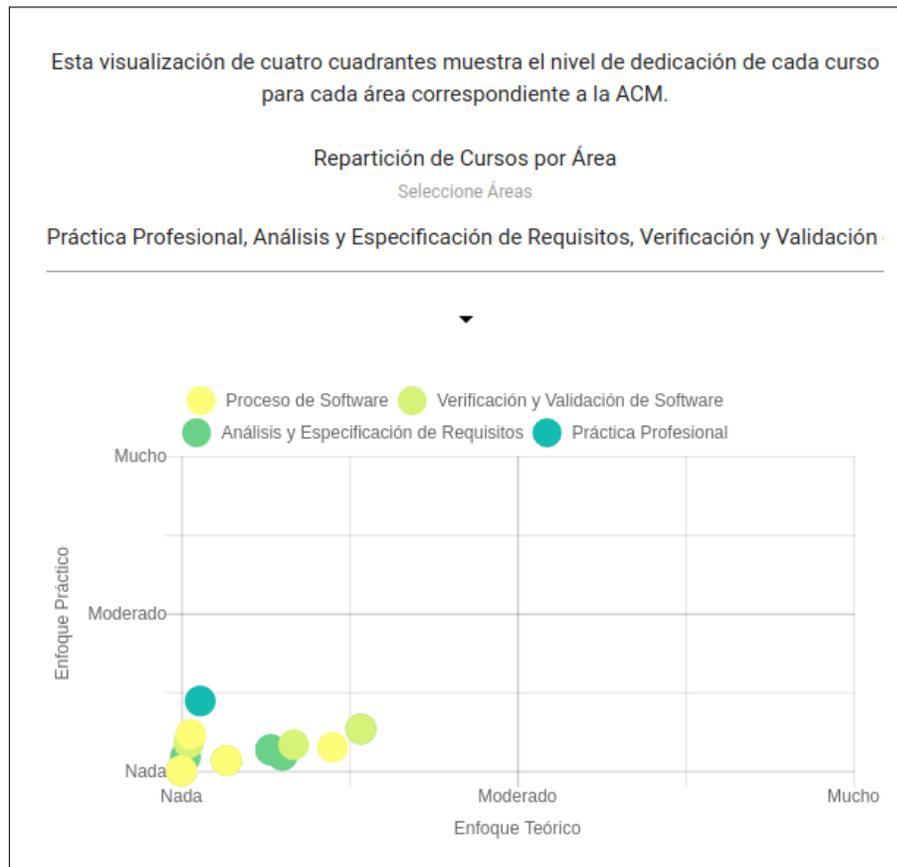


Figura 4.5: Visualización del análisis M4.

```

276 filter_results_program = models.CareerResults.objects.filter(period=survey.period,
277                                                                career=survey.course.career)
278
279 filter_results_institution = models.InstitutionResults.objects.filter(period=survey.period,
280                                                                        institution=survey.course.
281                                                                        career.institution)
282
283 if not filter_results_institution:
284     plain_career = survey.period.name + survey.course.career.institution.name + datetime.now().isoformat()
285     p = plain_career.encode('utf-8')
286     token_inst = sha1(p).hexdigest()
287     institution_program = InstitutionResults.objects.create(period=survey.period,
288                                                            institution=survey.course.career.institution,
289                                                            link=token_inst)
290     institution_program.save()
291 if not filter_results_program:
292     plain_career = survey.period.name + survey.course.career.name + datetime.now().isoformat()
293     p = plain_career.encode('utf-8')
294     token_career = sha1(p).hexdigest()
295     results_program = CareerResults.objects.create(period=survey.period,
296                                                    career=survey.course.career,
297                                                    link=token_career)
298     results_program.save()

```

Figura 4.6: Extracto de código que asocia una carrera con un periodo, creando así un vínculo específico para los resultados de esa carrera.

#### 4.1.4. Implementación de la visualizaciones

Para la implementación de las visualizaciones, se creó una interfaz que usa el vínculo generado por el sistema al momento de asociar una carrera con un periodo. Este vínculo es enviado a los profesores al momento de mandar los reportes generados por el sistema, tarea que sólo los administradores pueden realizar.

Luego de eso, se generaron los endpoints necesarios para obtener la información adecuada para las visualizaciones. Cada uno de estos endpoints posee una ruta que es llamada a través de AJAX, una técnica de desarrollo web para crear aplicaciones dinámicas. Se usó esta técnica para realizar peticiones a la aplicación en segundo plano, sin tener que recargar la página web completa. A continuación se detallan estos endpoints:

- E1** - *get\_alumni\_ajax\_program*: Esta función recibe como identificador un objeto *CareerResults* y entrega un objeto JSON, un formato de texto para el intercambio de datos, con la cantidad de alumnos egresados por cada periodo de la carrera específica. Además, entrega el promedio de la cantidad de alumnos egresados a nivel país. Este endpoint es usado por el análisis de datos **M1**.
- E2** - *get\_progression\_ajax\_program*: Similar a la función anterior, ésta recibe el identificador de un objeto *CareerResults* y le entrega al front-end un objeto JSON con los cursos de la carrera que participaron en el periodo de la encuesta, la cantidad de horas de docencia teóricas y prácticas, y el semestre en el cuál se imparte el curso para ordenar los cursos según el orden en el que se imparten. Este endpoint es usado por el análisis de datos **M2**.

**E3** - *get\_areas\_radar\_ajax\_program*: Esta función calcula el porcentaje de horas dedicadas a la enseñanza de las áreas de conocimiento de una carrera específica, con respecto al curriculum guidelines. Además, calcula el mismo porcentaje a nivel de país para poder comparar ambas entidades. Este endpoint es usado por el análisis de datos **M3**.

**E4** - *get\_focus\_per\_area\_program*: Aparte de recibir el identificador de un objeto *CareerResults*, esta función recibe una lista con identificadores de objetos *KnowledgeArea* para calcular cuántas horas de su curso un profesor le dedica a la enseñanza de cada una de esas áreas de conocimiento. El front-end, al final, recibe un objeto JSON con esta información. Este endpoint es usado por el análisis de datos **M4**.

La interfaz fue dividida en cuatro cuadrantes, donde cada uno de ellos contiene una visualización. El primer cuadrante muestra el gráfico que corresponde análisis de datos **M1**. El segundo corresponde al gráfico que representa el análisis de datos **M2**. El tercero muestra la visualización usada para el análisis de datos **M3**, y el cuarto muestra la visualización que representa el análisis de datos **M4**.

## 4.2. Reestructuración del reporte

Finalizando la implementación de las visualizaciones, se comenzó con el análisis e implementación del reporte que genera el sistema, en base a los resultados de un periodo de encuestas. Como se mencionó en la subsección 3.2.2, el reporte generado no muestra información agregada que le pueda resultar útil a un docente. En las siguientes subsecciones se explica cómo se abordó este problema.

Nombre Docente	Institución	Programa	Curso	Progreso encuesta	Estado encuesta	Ver encuesta	Descargar informe
Javier Espinoza	Universidad de Chile	Ingeniería Civil en Computación v3	Curso de Prueba	100%	Completada		

Figura 4.7: Interfaz que permite descargar o enviar reportes a los profesores. El cuadrado rojo corresponde a la opción de mandar los reportes a los profesores. El cuadrado verde indica la opción de descargar el informe de la encuesta en específico.

### 4.2.1. Aspectos generales del reporte

Existen dos formas de acceder a un reporte de encuesta, la primera consiste en enviarlo a través de un correo electrónico a cada persona que completó una encuesta. La segunda forma es descargar el informe de una encuesta en específico a través de una interfaz de administrador. La figura 4.7 muestra la interfaz que permite llevar a cabo estas acciones.

Para crear el reporte en si, el sistema recolecta información acerca de una encuesta y la guarda en un diccionario. Luego de eso, el sistema abre un archivo .tex que corresponde a la plantilla del informe. Esta plantilla posee strings entre llaves que deben ser reemplazados

por los valores que están en el diccionario. Por ejemplo, si se encuentra en la plantilla un string `{{alumnos}}`, el sistema irá al diccionario y buscará una llave `alumnos`. Si es que la encuentra, reemplazará el string `{{alumnos}}` en el archivo `.tex` por el valor encontrado en el diccionario, usando la llave `alumnos`.

Luego de reemplazar los datos, el sistema genera un archivo nuevo, que es temporal, con los datos reemplazados. Para finalizar el proceso, el sistema usa el comando `pdflatex` [15] para compilar el archivo `.tex` a PDF y, así, generar el reporte de la encuesta de manera automática. Los siguientes comandos ilustran los últimos pasos antes de poder descargar el reporte.

```
pdflatex cliente/static/tex/temp.tex
mv temp.pdf cliente/static/tex
```

Como se mencionó anteriormente, las únicas forma para generar reportes son a través de enviarlos por correos electrónicos a cada uno de los docentes que participó en el periodo de encuesta uno por uno, o descargando un reporte específicamente (ver figura 4.7). Ya que los correos se mandan a través de una cola y no paralelamente, usar un nombre estático no genera problemas de *data-race*, pues nunca hay dos reportes generándose al mismo tiempo.

Este flujo para generar los reportes no fue modificado, debido a que permite crear reportes para cada profesor de manera dinámica y rápida. Sin embargo, como será explicado en la subsección 4.2.3, fue necesario añadir más compilaciones de archivos `.tex` debido a la cantidad de secciones y tablas que la nueva estructura del reporte posee.

## 4.2.2. Estructura del reporte nuevo

Debido a que el reporte anterior ofrecía poca información útil, se decidió reestructurar el documento en su completitud. La intención detrás de esta reestructuración fue crear un documento oficial que mostrara información que le pudiese servir al país, a las instituciones de educación superior, y a las carreras que éstas imparten. Por esto, se decidió separar el reporte en las siguientes tres versiones.

- *Reporte nacional*: Esta versión del reporte presenta solamente información acumulada acerca de la enseñanza de la ingeniería de software en el país, dejando afuera cualquier tipo de información personal que pueda ser usada para distinguir las respuestas de un profesor en específico. Este reporte es público y puede ser accedido por cualquiera que lo desee.
- *Reporte institucional*: Esta versión del informe incluye la anterior, y además presenta información acumulada acerca de la enseñanza de la ingeniería de software en una institución en particular. Dicho reporte se le entrega a los profesores de la institución, y puede ser solicitado por autoridades (representantes) de dicha institución.
- *Reporte por carrera*: Por último, esta versión incluye el reporte anterior, y además presenta información detallada de los cursos de ingeniería de software de una carrera que pertenezca a la institución previamente señalada. Solamente los profesores participantes del periodo de encuesta reciben el informe por carrera.

Todas estas versiones del reporte vienen con un anexo que explica las fórmulas utilizadas para calcular los datos, y los nombres de los profesores que dieron la autorización para aparecer como participantes de dicha encuesta.

La reestructuración del reporte de la encuesta permite entregar información más útil a los profesores a nivel de país, por institución, y por carrera. Un reporte a nivel de curso no aportaría mucho dado que la información que pueda servirle al profesor de un curso, él/ella ya la conoce (es la indicada por él/ella en la encuesta) y además se encuentra contenida en el reporte por carrera.

Finalizando esta reestructuración, se estableció qué tipo de información debía ir en cada versión del reporte. En esta etapa de la implementación, se tuvo que abordar un problema que no se había considerado anteriormente: mostrar datos parametrizados en LaTeX.

### 4.2.3. Generación de plantillas parametrizadas

Dado que se está trabajando con un sistema de recolección de datos, puede darse que una carrera o institución posean mayores o menores cantidades de datos con respecto a las otras. Por ejemplo, una carrera puede poseer 3 cursos que respondieron una encuesta, mientras que otra carrera puede tener 5 cursos. También puede ocurrir que se ingrese una nueva ciudad a la base de datos, o que se incorpore un nuevo estándar (o Currículum Guideline) con más o menos áreas de conocimiento que el anterior.

Esto representa un problema para el sistema, debido a que actualmente sólo se trabaja con plantillas fijas para crear el reporte de cada encuesta. Esto implica que no se pueden generar dinámicamente tablas, listas, o reportes que se ajusten a los datos ingresados para cada carrera o institución. Es por esto que se creó un submódulo (dentro del módulo **MD10: Generación de Reportes**) llamado **MD: Generación de Plantillas** que genera tablas, listas o secciones que se ajustan a los datos ingresados dentro del sistema.

Para este submódulo se crearon siete servicios que funcionan de la misma manera: el módulo crea un archivo *.tex* vacío y escribe una lista o tabla en formato LaTeX donde el largo de la lista, o la cantidad de celdas de la tabla, depende de los datos entregados a la función correspondiente. Cada ítem de una tabla o lista generada posee un string entre llaves que corresponde a un valor a buscar en un diccionario, similar al método por el cual se genera un reporte, según lo explicado en la sección 4.2.1. Por último, estas tablas o listas son incluidas en el documento base del reporte usando el comando `\input{}` de LaTeX.

Con esta solución no existen problemas de escalabilidad, dado que cada archivo *.tex* posee un nombre específico que nunca cambia. Por ejemplo, un archivo que muestre cuántas horas le dedica cada región del país a la enseñanza de un área de conocimiento siempre tendrá el nombre *table-cities.tex*. Esto significa que si el sistema genera un reporte de una carrera, después de haber generado uno de otra, el sistema no creará otro archivo *table-cities.tex*. Lo que hará será abrir el archivo *table-cities.tex* previamente generado, borrar el contenido de éste y escribir los nuevos contenidos en dicho archivo, evitando así tener un archivo *table-cities.tex* distinto para cada reporte generado. Si bien se podría haber usado otra librería como WeasyPrint [16] (herramienta que renderiza HTMLs para generar PDFs), resultaba mucho más fácil generar plantillas LaTeX debido a que ya se estaba trabajando con LaTeX

---

```

1 def write_table_cities(number_cities, number_areas):
2     rows = '|c|'
3     areas = r"Area de Conocimiento"
4     file2 = open('cliente/static/tex/tables/table-cities.tex', 'w+').close()
5     file = open('cliente/static/tex/tables/table-cities.tex', 'w+')
6     file.write(r"\begin{sidewaystable}" + "\n")
7     file.write(r"\centering" + "\n")
8     file.write(r"\caption{Distribución de Horas Por Área en cada Región\label{tableregion}}" + "\n")
9     columns = " & \multicolumn{" + str(number_cities) + "}{c|}{Regiones} " + "\\\\" + r"\hline" + "\n"
10    for i in range(number_cities):
11        rows += 'c|'
12        areas += r"& \rotatebox{90}{ {{city_name_" + str(i + 1) + "}} }"
13    file.write(r"\begin{tabular}" + rows + ")" + "\n")
14    file.write(r"\hline" + "\n")
15    file.write(columns)
16    file.write(areas + " \\\\" + "\n")
17    file.write("\hline" + "\n")
18    for i in range(number_areas):
19        initial_city = r"{{A" + str(i + 1) + "}}"
20        for j in range(number_cities):
21            initial_city += " & {{city_" + str(j + 1) + "_area_" + str(i + 1) + "}}"
22            file.write(initial_city + "\\\\" + "\n")
23            file.write("\hline" + "\n")
24        file.write(r"\end{tabular}" + "\n")
25        file.write(r"\label{city-hours}" + "\n")
26        file.write(r"\end{sidewaystable}" + "\n")
27    file.close()

```

---

Listado 1: Extracto de código de una de las funciones que genera un archivo .tex dependiendo de los valores que se le entrega.

para generar la versión antigua del reporte.

Dado a que estos archivos se generan al momento de crear un reporte, no existen problemas de *data-race* por las mismas razones explicadas en la subsección 4.2.1.

En el listado 1 se muestra un extracto de código del submódulo creado. En este caso, la función genera una tabla dependiendo del número de ciudades y áreas de conocimiento. Lo que hace es definir los distintos nombres de los cabezales de la tabla a generar, y luego abre un archivo llamado *table-cities.txt* donde escribe una tabla en LaTeX a partir del número de ciudades, y número de áreas que se le entrega a la función. La función después añade una leyenda a la tabla, y luego cierra el archivo.

En el listado 2 se muestra un extracto del archivo generado, y en la figura 4.8 se muestra el resultado de compilar este archivo.

#### 4.2.4. Implementación del reporte

Habiendo resuelto el problema anterior, se prosiguió con la implementación de las tres versiones del reporte mencionadas anteriormente. A continuación se explica la información que se encuentra en cada una de éstas.

##### Reporte nacional

El reporte Nacional muestra los siguientes cuatro indicadores:

- *Información general acerca del país:* En esta sección se muestran la cantidad de instituciones que imparten una carrera que incluye cursos de ingeniería de software (típica-

---

```

1 \begin{sidewaystable}
2 \centering
3 \begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|}
4 \hline
5 & \multicolumn{12}{c}{Regiones} & \hline
6 Area de Conocimiento& \rotatebox{90}{ {{city_name_1}} }& \rotatebox{90}{ {{city_name_2}} }&
  ↳ \rotatebox{90}{ {{city_name_3}} }& \rotatebox{90}{ {{city_name_4}} }& \rotatebox{90}{
  ↳ {{city_name_5}} }& \rotatebox{90}{ {{city_name_6}} }& \rotatebox{90}{ {{city_name_7}} }&
  ↳ \rotatebox{90}{ {{city_name_8}} }& \rotatebox{90}{ {{city_name_9}} }& \rotatebox{90}{
  ↳ {{city_name_10}} }& \rotatebox{90}{ {{city_name_11}} }& \rotatebox{90}{ {{city_name_12}} } & \hline
7 \hline
8 {{A1}} & {{city_1_area_1}} & {{city_2_area_1}} & {{city_3_area_1}} & {{city_4_area_1}} & {{city_5_area_1}}
  ↳ & {{city_6_area_1}} & {{city_7_area_1}} & {{city_8_area_1}} & {{city_9_area_1}} & {{city_10_area_1}}
  ↳ & {{city_11_area_1}} & {{city_12_area_1}}\hline
9 \hline
10 {{A2}} & {{city_1_area_2}} & {{city_2_area_2}} & {{city_3_area_2}} & {{city_4_area_2}} & {{city_5_area_2}}
  ↳ & {{city_6_area_2}} & {{city_7_area_2}} & {{city_8_area_2}} & {{city_9_area_2}} & {{city_10_area_2}}
  ↳ & {{city_11_area_2}} & {{city_12_area_2}}\hline
11 \hline
12 {{A3}} & {{city_1_area_3}} & {{city_2_area_3}} & {{city_3_area_3}} & {{city_4_area_3}} & {{city_5_area_3}}
  ↳ & {{city_6_area_3}} & {{city_7_area_3}} & {{city_8_area_3}} & {{city_9_area_3}} & {{city_10_area_3}}
  ↳ & {{city_11_area_3}} & {{city_12_area_3}}\hline
13 \hline

```

---

Listado 2: Extracto del archivo generado por el código del listado 1.

mente son carreras de Computación o Informática), la cantidad de carreras, la cantidad de cursos de ingeniería de software, y la cantidad de alumnos totales que egresaron de dichas carreras. Esta sección sirve para comunicar la situación general de la enseñanza de la ingeniería de software a la persona que lee el reporte.

- *Horas dedicadas a la enseñanza de áreas de conocimiento a nivel nacional comparadas con el Currículum de Referencia de la ACM/IEEE:* Luego de la sección anterior, se muestra una tabla donde se comunica el promedio de horas dedicadas a la enseñanza de las diversas áreas de conocimiento indicadas por la recomendación de referencia, las horas recomendadas por el Currículum de Referencia, y el estado del país para cada área. Este estado indica si la cantidad de horas dedicadas a un área de conocimiento está bajo el 33 % de las horas recomendadas, bajo del 66 %, o arriba de este valor. Esta sección nos permite tener una idea rápida y visual de cómo se imparte la ingeniería de software a nivel nacional según el Currículum de Referencia.
- *Información general acerca de las regiones del país:* En esta sección, se muestra un promedio de los datos generales de la encuesta a nivel de región. Por ejemplo, se muestra la cantidad de horas de docencia prácticas que la Región Metropolitana imparte en promedio.
- *Horas totales dedicadas a la enseñanza de áreas de conocimiento a nivel regional:* Similar a la segunda sección, ésta se encarga de mostrar las horas totales dedicadas a la enseñanza de las áreas de conocimiento a nivel de región.

El objetivo de estas dos secciones es ver qué tan distinta es la enseñanza de la ingeniería de software, a través de las regiones del país. Dado que la cantidad de datos que se tienen por región es menor que por país, se decidió usar las horas totales en vez de las horas promedio por región.

Área de Conocimiento	Regiones											
	Antofagasta	Araucanía	Arica y Parinacota	Atacama	Biobío	Coquimbo	Los Ríos	Maule	Metropolitana	Tarapacá	Valparaíso	Ñuble
Análisis y Especificación de Requisitos	22.4	0.0	-	0.0	14.8	0.0	-	-	45.8	-	-	-
Calidad del Software	0.0	22.5	-	16.2	40.8	0.0	-	-	0.0	-	-	-
Diseño de Software	0.0	0.0	-	0.0	15.3	0.0	-	-	36.0	-	-	-
Fundamentos de Ingeniería y Matemática	0.0	0.0	-	0.0	0.0	0.0	-	-	0.0	-	-	-
Fundamentos de la Computación	21.0	9.0	-	0.0	0.0	0.0	-	-	0.0	-	-	-
Modelamiento y Análisis de Software	9.6	0.0	-	0.0	14.8	0.0	-	-	30.2	-	-	-
Proceso de Software	0.0	0.0	-	10.8	62.1	26.0	-	-	0.0	-	-	-
Práctica Profesional	84.0	0.0	-	0.0	22.5	0.0	-	-	0.0	-	-	-
Seguridad	0.0	13.5	-	0.0	0.0	0.0	-	-	0.0	-	-	-
Verificación y Validación de Software	0.0	0.0	-	27.0	39.6	26.0	-	-	0.0	-	-	-

Tabla 4. Distribución de Horas Por Área en cada Región

Figura 4.8: Tabla generada al compilar el archivo del listado 2.

## Reporte por institución

El reporte por Institución muestra los siguientes indicadores:

- *Información acumulada por institución:* Esta sección permite comparar los datos generales ingresados por los docentes entre las carreras pertenecientes a una institución. Por ejemplo, se puede ver la cantidad de alumnos que egresaron de ambas carreras, o el promedio de horas de docencia prácticas por carrera.
- *Horas totales dedicadas a la enseñanza de áreas de conocimiento a nivel institucional, comparadas con el currículum de la ACM/IEEE:* Esta sección compara las horas totales que cada carrera perteneciente a la institución le dedica a la enseñanza de las diversas áreas de conocimiento involucradas en la ingeniería de software. Además, se ve si la cantidad de horas está debajo del 33% de las horas recomendadas por el currículum, del 66%, o arriba de estos valores.

## Reporte por carrera

El reporte por carrera muestra los siguientes cuatro indicadores:

- *Profesores y cursos que participaron en el periodo de encuesta:* En esta sección se muestran los profesores (con sus cursos correspondientes) que participaron en la encuesta y dieron su autorización para mostrar su nombre. La idea de esta sección es dar a conocer a los profesores de una carrera que participaron de la encuesta (previo consentimiento de ellos), permitiendo así claridad dentro del programa.
- *Información general de cada curso de la carrera:* En esta sección, se muestra en detalle los datos generales de un curso, que cada profesor llenó en su encuesta. Por ejemplo, se detalla la cantidad de horas de docencia teórica que un profesor imparte en su curso.

¿Cuál es el código que identifica al curso dentro del programa?	Código de Identificación del Curso:	
¿Cuántos créditos transferibles (SCT) involucra el curso?	Créditos del Curso (ej: 3):	
¿Cuántas horas semanales de docencia directa posee este curso?	Horas Teóricas (ej: 2.XX):	Horas Prácticas (ej: 2.XX):
Anualmente, ¿cuántos alumnos toman este curso en promedio?	Número de Alumnos Promedio:	
Según la malla del programa, ¿es el curso obligatorio?	Tipo de Curso: <input checked="" type="radio"/> Sí <input type="radio"/> No	
Según la malla del programa, ¿en qué semestre se imparte el curso?	Semestre del Curso (ej: 5):	

Figura 4.9: Extracto de la primera parte de la encuesta.

Esta información es importante para ver la evolución de estos datos a través de los cursos de un programa y también a través del tiempo.

- *Horas totales dedicadas a la enseñanza de áreas de conocimiento por cada curso de la carrera:* Esta sección se encarga de mostrar la cantidad de horas que los cursos de un programa le dedican a la enseñanza de las áreas de conocimiento. Similar a la sección anterior, el objetivo es ver la evolución de las horas dedicadas a la ingeniería de software.
- *Comparación de información general de la carrera con el promedio a nivel país:* Esta sección toma los datos generales de una carrera, los promedia, y luego los compara con los datos generales en promedio a nivel de país. Esto le sirve a las carreras ver qué tan alejados o cercanos están de la enseñanza promedio de la ingeniería de software a nivel nacional.

Para obtener la información asociada a cada ítem mencionado, se crearon funciones que son llamadas al momento de generar el reporte; es decir, luego de concluido el período de encuesta. En la sección A.1 del anexo se muestra la tercera versión de dicho reporte. Éste incluye la información a nivel nacional, a nivel institucional, y la información de la carrera a la que pertenecen los cursos.

### 4.3. Cambios a las interfaces del sistema

Ciertas interfaces fueron cambiadas para calzar mejor con las nuevas funcionalidades incluidas en el sistema, ajustarse al nuevo modelo de datos definido, o para mostrar mejor la información de la encuesta a través de las interfaces de usuario. Cada apartado de esta sección explica los cambios realizados a las interfaces relacionadas, y también las interfaces nuevas que se agregaron.

## Encuesta de curso: Proyecto de Software II

### Acuerdo de Confidencialidad

Al contestar esta encuesta, usted se atiene al siguiente Acuerdo de Confidencialidad y No Divulgación:

Este acuerdo entre EDUCA-IS, quién en lo sucesivo se denominará "**EL RECEPTOR**", y por la otra el usuario que ingresa al sistema, quién se le denominará "**EL DIVULGANTE**", establece las siguientes cláusulas:

**PRIMERA.-**

**Objeto.** El presente Acuerdo se refiere a la información que EL DIVULGANTE proporcione al RECEPTOR, ya sea de forma oral, gráfica o escrita que sirva para llevar a cabo el estudio sobre la enseñanza de la Ingeniería de Software en Chile.

**SEGUNDA.-**

1. EL RECEPTOR únicamente utilizará la información facilitada por EL DIVULGANTE para el fin antes mencionado, comprometiéndose EL RECEPTOR a mantener la más estricta confidencialidad al respecto de dicha información, advirtiendo de dicho deber de confidencialidad y secreto las personas que participen en el procesamiento de la información y la generación de resultados.

2. EL RECEPTOR o las personas mencionadas en el párrafo anterior no podrán reproducir, modificar, hacer pública o divulgar a terceros de manera directa la información detallada objeto del presente estudio, sin previa autorización escrita y expresa del DIVULGANTE.

3. De igual forma, EL RECEPTOR adoptará respecto de la información objeto de este Acuerdo las mismas medidas de seguridad que adoptaría normalmente respecto a la información confidencial de su propia organización, evitando en la medida de lo posible su pérdida, robo o sustracción.

**TERCERA.-**

Sin perjuicio de lo estipulado en el presente Acuerdo, ambas partes aceptan que la obligación de confidencialidad no se aplicará en los siguientes casos:

Figura 4.10: Extracto mejorado de los términos y condiciones de participación en la encuesta

### 4.3.1. Encuesta

Debido a que se añadieron cuatro campos al modelo *Encuesta* (código del curso, cantidad de créditos, si es obligatorio o no el curso, y el semestre en el que se dicta), fue necesario cambiar la primera interfaz de la encuesta para que los profesores puedan llenar estos datos. Para el código del curso, semestre de dictado y cantidad de créditos del mismo, se utilizó un campo de texto, mientras que para la obligatoriedad del curso se utilizó un input de tipo radio. La figura 4.9 muestra, entre cuadrados rojos, los nuevos campos añadidos a la interfaz.

En la subsección 3.2.4 se detallaron algunos problemas con los términos y condiciones de la encuesta, así como con la paleta de colores usada para las tablas de la encuesta. Para el primer problema, se ordenó y mejoró la calidad de texto de los términos y condiciones, dejando más en claro lo que los docentes deben aceptar antes de responder la encuesta. En la figura 4.10 se muestra este nuevo formato.

El segundo problema implicó un cambio en la paleta de colores usada en las interfaces del sistema en general. En la nueva versión se usó una paleta de azules generada a través de Material Design<sup>1</sup>, una página que crea paletas de colores para interfaces de usuario. A continuación, en la figura 4.11 se muestra una mejora de la interfaz.

### 4.3.2. Profesores participantes

Anteriormente, la interfaz mostrada en la figura 4.10 solo indicaba los nombres de los profesores que daban su autorización para aparecer como participantes. Ahora, esta interfaz fue cambiada para que se mostrasen las instituciones a las que los docentes pertenecen. Este cambio ayuda a dar visibilidad acerca de las instituciones que han participado de la encuesta, a diferencia de antes donde sólo se mostraban nombres de personas.

<sup>1</sup><https://material.io/tools/color/#/?view.left=0&view.right=0>

EDUCA-IS: Encuesta sobre la Enseñanza de Ingeniería de Software

## Encuesta de curso: Curso de Prueba

*Ir a:*

Áreas de Enseñanza ▼

IR

### Áreas de Enseñanza

**Barra de Progreso:**

30%

Indique las áreas de conocimiento de la Ingeniería de Software abordadas por este curso. Además, indique el porcentaje del curso que se dedica a cada área de conocimiento. Es necesario que los porcentajes sumen 100% para proseguir con la encuesta.

Nombre del Área	Descripción del Área	Porcentaje del curso
Fundamentos de la Computación	Esta área incluye los fundamentos de ciencias de la computación que apoyan el diseño e implementación de productos de software. Esto incluye el conocimiento necesario para convertir un diseño en una implementación, así como las técnicas y herramientas utilizadas durante este proceso.	<p style="font-size: x-small;">% del Curso: _____</p> <input type="checkbox"/> No enseñó esta área

Figura 4.11: Extracto de interfaz mejorada de la segunda parte de la encuesta.

### 4.3.3. Reportes por institución

Para que un administrador pueda descargar reportes a nivel de institución y entregarlos a los miembros de misma, se creó una interfaz que es accedida a través de la interfaz donde se muestra la información de cada periodo de encuestas. La figura 4.13 muestra, en un cuadro rojo, el botón que permite al administrador acceder a dicha interfaz. Al clickear ese botón, se muestra la interfaz de la figura 4.14. En ésta, las instituciones que participaron en el periodo de encuestas en particular. Al lado de cada una de éstas, se encuentra un vínculo que permite descargar el reporte de la institución específica.

Se creó esta interfaz para que los administradores le puedan dar información a los miembros de una institución, sin tener que dar datos acerca de los cursos que pertenecen a ésta. Así, no se rompen los términos y condiciones establecidos en la encuesta.

## 4.4. Otros cambios realizados

En las limitaciones de las interfaces (ver sección 3.2.4) se mencionó que varias de éstas utilizaban llamadas GET para acceder a la información correspondiente. Esto fue cambiado para que el vínculo de la página correspondiente siempre tenga el identificador de un periodo o encuesta. Con esto, se evitan errores tales como errores al actualizar una página que fue accedida a través de una llamada GET, ya que la URL siempre le provee al sistema las variables necesarias para obtener los datos correspondientes. Dado esto, las URLs de las siguientes páginas fueron cambiadas:

- *Acceder a los detalles de un periodo de encuestas:* Para esta URL, se agregó a ésta un identificador que le permite al sistema reconocer qué periodo mostrar al administrador.

Un ejemplo de esta ruta es:

`https://gems.dcc.uchile.cl/admin/term/1`

- *Acceder a los detalles de una encuesta en un periodo en específico:* En esta ruta se hizo

EDUCA-IS: Encuesta sobre la Enseñanza de Ingeniería de Software	
Profesores contribuyentes a EDUCA-IS	
Nombre de Institución	Profesores Asociados
Universidad de Chile	Alexandre Bergel
	Javier Espinoza
	Sergio Ochoa
	Jocelyn Simmonds
	Francisco Gutierrez

Figura 4.12: Extracto de interfaz mejorada de los profesores que han dado su autorización para mostrar su nombre como participantes de la encuesta.

INICIAR NUEVO PERIODO ⊕		AÑADIR ESTÁNDAR ⊕				
Nombre	Inicio	Término	Detalles	Estadísticas	Reportes	Exportar
Periodo de Validacion	14/05/2019	14/05/2019				
Validación con Profesores	14/05/2019	<b>TERMINAR</b>				

Figura 4.13: Botón que redirige al usuario a la interfaz para descargar reportes de institución.

el mismo proceso que en la anterior, agregando dos identificadores en ésta, para que el sistema reconozca qué periodo y encuesta mostrar respectivamente. Un ejemplo de esta nueva ruta es:

<https://gems.dcc.uchile.cl/admin/survey/1/2>

En esta ruta, el primer número corresponde al identificador del periodo donde se llevó a cabo la encuesta, mientras que el segundo número corresponde a la encuesta misma.

- *Editar un programa ingresado en el sistema:* Se hizo el mismo proceso que las rutas anteriores. Un ejemplo de esta nueva ruta es:

<https://gems.dcc.uchile.cl/admin/edit/program/3>

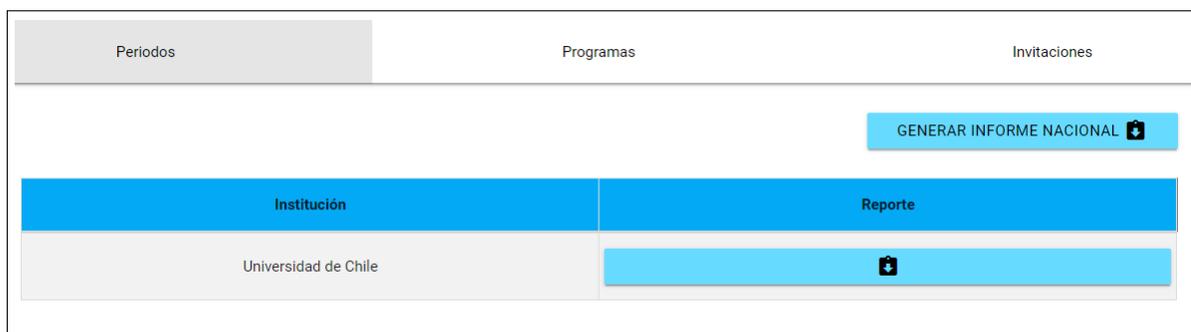


Figura 4.14: Interfaz que permite descargar reportes por institución.

## 4.5. Resumen

En esta sección se explicaron las decisiones tomadas al momento de abordar los problemas mencionados en el capítulo 3, así como las tecnologías usadas para la implementación de los nuevos módulos del sistema. Los requisitos **R1**, **R2** y **R3** se ven plasmados en las secciones 4.2 y 4.3.3, mientras que la implementación de los requisitos **R4**, **R5**, **RN1** y **RN2** se muestran en la sección 4.1.

Los puntos más importantes de este capítulo son los tipos de visualizaciones utilizados para los análisis de datos definidos en la sección 4.1, y la generación de plantillas en LaTeX dinámicas para que los reportes de cada carrera se ajusten a los datos ingresados en el sistema. En los siguientes capítulos se mostrará la validación de estas nuevas funcionalidades, y por último, las conclusiones que surgen como producto del trabajo realizado.

# Capítulo 5

## Validación

Para validar que la implementación del trabajo realizado es una mejora del estado del sistema legado, se llevaron a cabo ciertas instancias de validación de todos los componentes del sistema que fueron afectados por los nuevos módulos. Entre estas actividades, se encuentra la realización de un piloto de la encuesta que se llevó a cabo con los profesores de ingeniería de software de la Universidad de Chile, y una serie de entrevistas que se desarrollaron para validar la usabilidad y utilidad de las visualizaciones creadas. En la sección 5.1 se habla acerca de cómo se llevó a cabo el proceso de validación, qué fue lo que se validó, y cuáles fueron los resultados de esas validaciones.

### 5.1. Validación de usabilidad y utilidad del sistema

Antes de llevar a cabo el proceso de validación, fue necesario desarrollar un protocolo para desarrollar dicha actividad. La siguiente subsección explica ese proceso.

#### 5.1.1. Protocolo de la validación

El objetivo de esta validación fue determinar la usabilidad y utilidad de las visualizaciones implementadas en el capítulo anterior. Los pasos que se tomaron para diseñar este proceso fueron los siguientes:

1. Para determinar la cantidad de usuarios para la validación del sistema, se usó la metodología propuesta por Turner et al. [17], que postula que tres a cinco usuarios son necesarios para determinar la mayoría de los problemas de usabilidad. Estos usuarios, además, deberían ser profesores que imparten algún curso de la rama de ingeniería de software.
2. Luego de eso, se determinó que la forma de obtener datos para generar las visualizaciones era a través de un piloto, o focus group. En éste, cada uno de los profesores participantes recibirían un vínculo para una encuesta de algún curso que impartieran. Ahí tendrían que completar la encuesta con información fidedigna de sus cursos.
3. Terminado ese periodo de encuestas, se agendaría una reunión con tres de los profesores que participaron en el piloto. Esto se debe a las mismas razones explicadas por la

metodología de Turner et al. [17]. Si bien se podría haber contactado a profesores que no pertenecen a la Universidad de Chile, éstos no sabían de la encuesta, sus contenidos, y los resultados, lo que habría hecho incierto el proceso de evaluación.

4. Luego de definir quiénes serían las personas entrevistadas, se formularon preguntas para las visualizaciones generadas por el sistema para el período del piloto. Para las visualizaciones, se hicieron 2 preguntas. La primera apunta a la usabilidad de los gráficos, mientras que la segunda indaga acerca de si el gráfico es útil en un cierto contexto.
  - Al mirar el gráfico, ¿se entiende qué es lo que se quiere mostrar?
  - En un contexto donde se quiera hacer algún cambio a la malla curricular de una carrera, ¿es esta información útil?

### 5.1.2. Piloto de la encuesta

Con el protocolo diseñado, se comenzó el proceso de validación. Lo primero que se hizo fue llevar a cabo el piloto para recolectar información. Se implantó el sistema en el servidor GEMS del DCC luego de implementar todos los cambios mencionados en el capítulo anterior. Después de que esto se contactó a través de correo electrónico, a cinco profesores de la carrera Ingeniería Civil en Computación, que imparten cursos relacionados con la ingeniería de software:

**Alexandre Bergel:** Dicta el curso obligatorio de Metodologías de Diseño y Programación.

**Jocelyn Simmonds:** Dicta el curso obligatorio de Ingeniería de Software I.

**Sergio Ochoa:** Dicta el curso obligatorio de Ingeniería de Software II.

**Cecilia Bastarrica:** Dicta el curso obligatorio de Proyecto de Software.

**Francisco Gutiérrez:** Dicta el curso electivo de Diseño de Sistemas Interactivos.

Si bien se podría haber llamado a profesores de otras universidades, se decidió contactar a los profesores de esta carrera para corroborar que lo que muestra el reporte generado por el sistema es útil y tiene sentido.

El periodo para responder las encuestas fue desde el 14 de mayo al 3 de junio del 2019. A cada profesor se le pidió que llenara una encuesta con datos que correspondían a los cursos mencionados anteriormente. Terminado el periodo, se generaron los reportes y visualizaciones correspondientes. En el anexo A.1 se muestra un reporte completo de uno de los profesores, mientras que las imágenes 5.1, 5.2, 5.3 y 5.4 corresponden a las visualizaciones generadas por el sistema.

### 5.1.3. Primera ciclo de entrevistas

Cada profesor fue contactado a través de correo electrónico, y se agendaron reuniones en distintos periodos de tiempo. en las reuniones se les dio un contexto acerca del por qué se recolectaban los datos, y se les recordó la estructura de la encuesta que respondieron. Luego de eso, se les mostraron las visualizaciones desarrolladas para la carrera Ingeniería Civil en Computación, y se les hicieron las preguntas mencionadas en la sección 5.1.1. Cada visuali-

zación tenía un significado que se esperaba que los profesores entendieran. A continuación, se presentan esos significados:

- Para la primera visualización (ver figura 5.1), se esperaba que los docentes entendieran que la cantidad de alumnos que egresaban de la carrera aumentaron en el segundo periodo. Además, que ese valor era igual a la cantidad de alumnos que egresan en

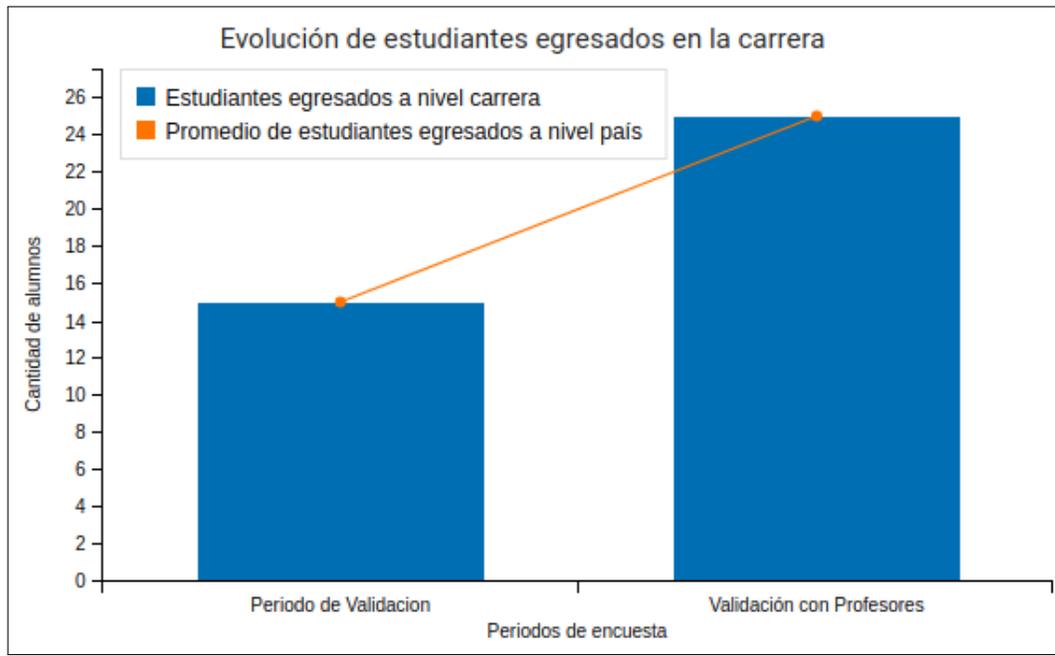


Figura 5.1: Primera visualización generada para el periodo del piloto, que corresponde al análisis M1.

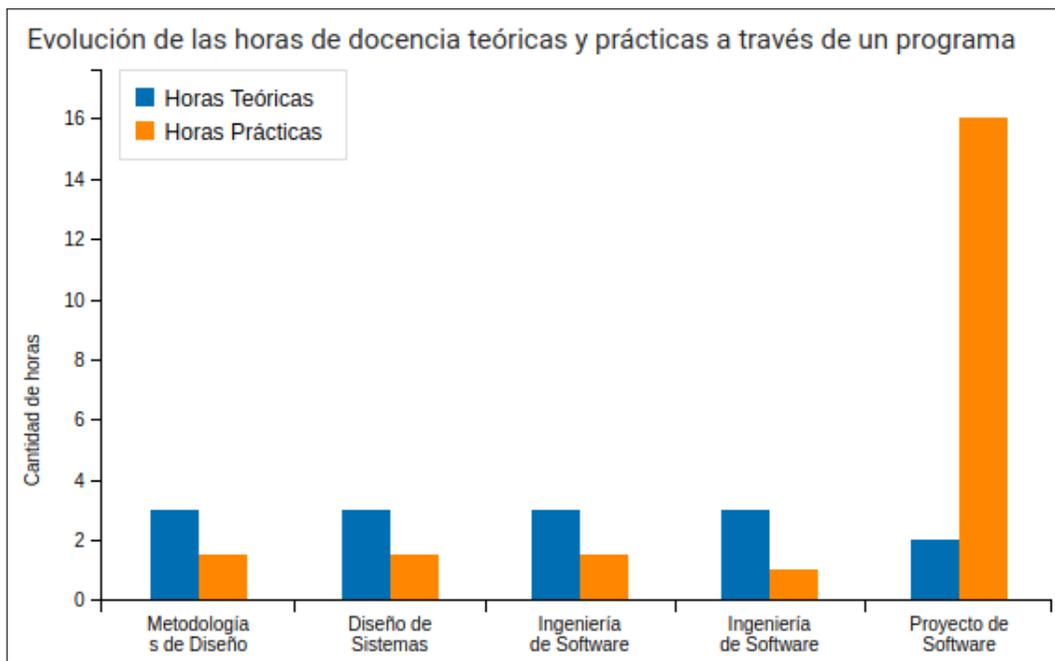


Figura 5.2: Segunda visualización generada para el periodo del piloto, que corresponde al análisis M2.

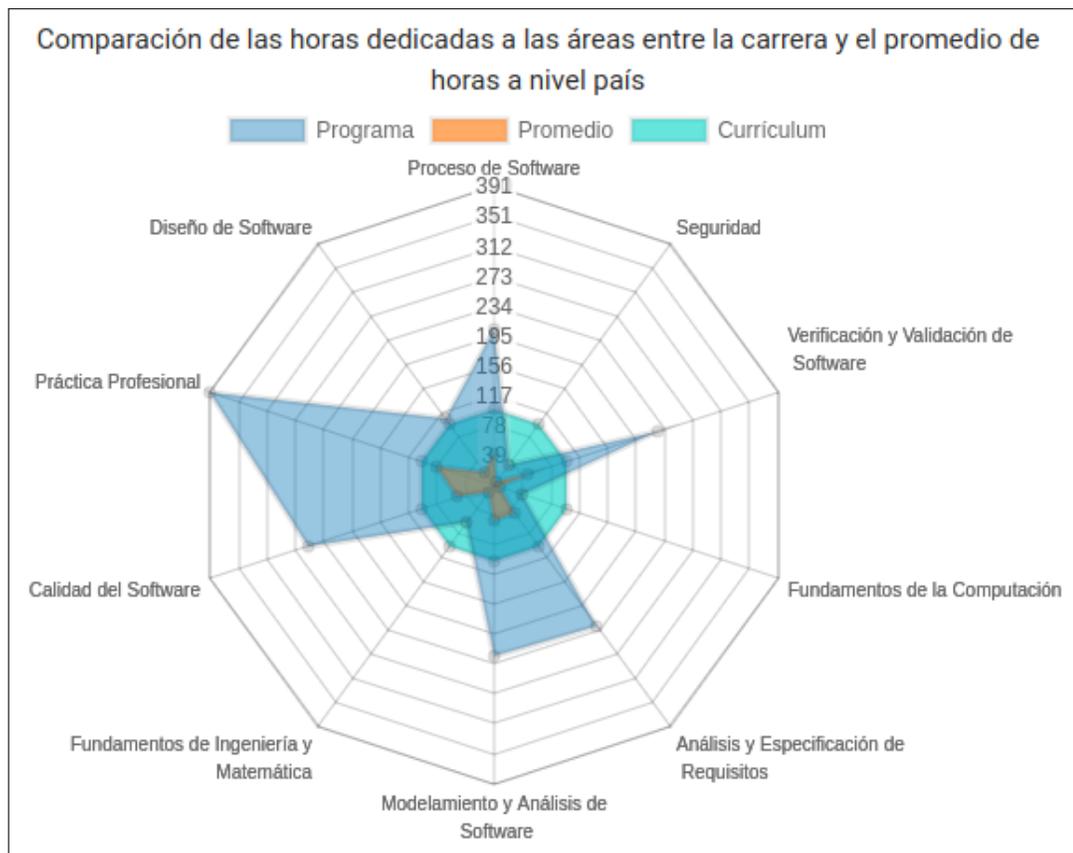


Figura 5.3: Tercera visualización generada para el periodo del piloto, que corresponde al análisis **M3**.

promedio, en las carreras que participaron en la encuesta.

- Se esperaba que para la segunda visualización (ver figura 5.2), se entendiera que la cantidad de horas de docencia teóricas y prácticas semanales se mantenían constante hasta el curso de Proyecto de Software (último de la malla de la carrera), donde la cantidad de horas de docencia práctica aumentaba considerablemente.
- Para la tercera visualización (ver figura 5.3), se esperaba que los profesores entendieran que la carrera de Ingeniería Civil en Computación de la Universidad de Chile dedicaba, en ciertas áreas, más horas de lo recomendado por el Currículum de Referencia de la ACM/IEEE, y más horas que el promedio a nivel de las carreras que participaron en la encuesta.
- Lo esperado para la cuarta visualización (ver figura 5.4) era que los profesores entendieran que la mayoría de los cursos le prestaban poca atención a algunas de las áreas de conocimiento establecidas por el Currículum de Referencia. Estas áreas eran: Fundamentos de la Computación, Análisis y Especificación de Requisitos, Práctica Profesional, Modelamiento y Análisis de Software, y Fundamentos de la Ingeniería y Matemáticas.

Los primeros profesores con los que se tuvo una reunión fueron el profesor Bergel y la profesora Bastarrica. Se les contactó a través de correo electrónico y se agendó una entrevista con ellos. A continuación se discute lo que ellos dijeron acerca de la usabilidad de los análisis de datos implementados implementadas.



Figura 5.4: Cuarta visualización generada para el periodo del piloto, que corresponde al análisis M4.

### Usabilidad de los análisis de datos

De los comentarios que hicieron los profesores acerca de la usabilidad de los análisis de datos y sus visualizaciones, los principales fueron los siguientes:

- La primera visualización no fue entendida por ambos profesores. Mencionaron, además, que no entendían el concepto de “periodos de encuesta”, ni de con quién se estaba comparando la cantidad de alumnos egresados en el programa. El profesor Bergel dio como sugerencia que, en vez de hacer una comparación por periodos de encuesta, se hiciesen por años.
- Ambos profesores entendieron la segunda visualización. La profesora Bastarrica criticó la poca visibilidad de las etiquetas de los cursos, y el hecho que el título de la visualización no especificase si las horas que se comparaban eran semanales o totales.
- Si bien el profesor Bergel entendió que era lo que quería comunicar el análisis de la tercera visualización, la profesora Bastarrica no lo entendió, indicando que el gráfico era, según sus palabras, “raro y confuso”. En esta visualización, ambos profesores expresaron la necesidad de algún recordatorio de la encuesta, ya que, según ellos, no era fácil acordarse qué fue lo que se preguntó en la encuesta.
- La cuarta visualización fue la más difícil de entender para los dos profesores. Ambos indicaron que las áreas de conocimiento y los cursos no estaban claros, y que el hecho

de que varios puntos estaban encima de otros hacía muy difícil la comprensión de la visualización.

## Utilidad de los análisis de datos

De los comentarios que hicieron los docentes acerca de la utilidad de los análisis de datos y sus visualizaciones, los más importantes fueron los siguientes:

- Los profesores no le encontraron utilidad a la primera visualización. El profesor Bergel, sin embargo, mencionó que sería útil comparar la cantidad de alumnos que salen de la Universidad de Chile, contra los que salen de otras instituciones. Sin embargo, esto está afuera del alcance del trabajo de memoria.
- En términos de utilidad de la segunda visualización, la profesora Bastarrica la consideró útil con tal que hubiese más información de afuera (o sea, que no fuese parte del sistema) que complementase esa visualización. Sin embargo, el profesor Bergel no le encontró utilidad a la visualización debido a que era información que ya sabía de antes.
- El profesor Bergel declaró que le encontraba utilidad a la visualización, indicando que podría servir para un contexto donde se requieren hacer cambios a la malla curricular, pero la profesora Bastarrica no le pudo encontrar utilidad debido al diseño de la visualización.
- Debido a que los profesores no entendieron la cuarta visualización, no le encontraron utilidad al gráfico.

### 5.1.4. Discusión de los resultados

Terminadas estas entrevistas, se empezó a ver que los análisis de datos, y sus visualizaciones, no eran del todo claras. La visualización del análisis **M1** no mostraba una evolución a través del tiempo, la visualización del análisis **M2** no decía si las horas que se mostraban eran semanales o totales, el gráfico del análisis **M3** era difícil de entender debido a la normalización de los datos, y que el análisis **M4** era muy difícil de entender debido a que los puntos de la visualización se traslapaban los unos con los otros.

En términos de utilidad, las visualizaciones de los análisis **M2** y **M3** fueron las únicas que los profesores consideraron útiles en un contexto donde se quisiera mejorar la malla curricular del programa analizado. Las visualizaciones de los análisis **M1** y **M4**, en cambio, no eran útiles según los profesores. De todo esto, se pudo rescatar que la profesora Bastarrica mencionó, en sus palabras, que las visualizaciones que se desarrollaron en este trabajo son de mejor calidad y proveen mejor información que las del sistema legado. Para hacer una comparación, dos de las visualizaciones anteriores generadas por el sistema legado se encuentran en la sección A.2 del anexo.

Debido a que los resultados de ambas entrevistas no fueron positivos, y que validar con el profesor Gutiérrez habría dado los mismos resultados (o similares) que las validaciones anteriores, se decidió parar el proceso de validación y ver de qué manera se podían mejorar la usabilidad y utilidad de los gráficos de los análisis de datos previamente implementados. Basado en los comentarios que dieron los profesores Bergel y Bastarrica, se tomó la decisión de eliminar el gráfico del análisis **M4** debido a su utilidad nula y la manera desordenada en

la cual presentaba los datos. Además, se separó el análisis **M3** en dos análisis distintos:

**M3a.** Una donde se compara la cantidad de horas totales que una carrera le dedica a la enseñanza de un área de conocimiento, con la cantidad de horas totales recomendadas por el Currículum Guidelines 2014.

**M3b.** Otra donde se compara lo mismo que el análisis anterior, sólo que en vez de comparar con la cantidad de horas recomendadas por el Currículum Guidelines 2014, se compara con el promedio de horas totales que las carreras que participaron en la encuesta le dedican a la enseñanza de las áreas de conocimiento.

Estos dos análisis tendrían la opción de considerar sólo las horas totales de los cursos obligatorios, o las horas totales de todos los cursos de un programa, incluyendo los electivos. Esta decisión fue hecha pues no todos los alumnos que cursan un programa inscribirán los cursos electivos de éste, por lo que es necesario saber cuál es el estado del programa considerando sólo los cursos obligatorios, y también su estado considerando todos los cursos del programa (obligatorios y electivos).

Ambos análisis utilizan la misma técnica de visualización usada para el análisis **M3**, un gráfico de radar, porque se requiere comparar múltiples variables cuantitativas (horas dedicadas a la enseñanza de las áreas de conocimiento) a nivel de dos entidades: En el análisis **M3a**, se está comparando la carrera con el documento de la ACM. En el análisis **M3b**, se está comparando la carrera con el promedio de todas las otras carreras que participaron en la encuesta.

También se decidió cambiar el análisis de datos **M2**, optando por ver una evolución de las horas de docencia directa semanales (cantidad de horas teóricas y prácticas) a través de los semestres de un programa, en vez de indicarlo para cada curso. Esto se hizo para reducir la cantidad de barras en la visualización del análisis. Además, esto se cambió para mostrar una evolución a través del programa de una manera más clara, ya que en la visualización del análisis **M2** anterior no estaba claro si los cursos estaban ordenados por semestres o no. Por último, el tipo de visualización usado para este nuevo análisis sigue siendo barras, dado que se están comparando dos variables discretas a través de una línea temporal.

Otra medida que se tomó para mejorar el reporte generado por el sistema, fue añadir una página de contexto acerca de la encuesta, la cual aparece al inicio del reporte. Esto se hizo para que los profesores se acordasen de las preguntas que se le hicieron en la encuesta, y para recordarles acerca que la comparación se hace contra el Currículum de Referencia de la ACM/IEEE. En las figuras 5.5 y 5.6 se muestran los extractos de esta página.

La última medida tomada fue eliminar dos áreas de conocimiento de la base de datos: Fundamentos de la Computación, y Fundamentos de Ingeniería y Matemáticas. La razón por la cual se tomó esta decisión es que, si bien estas áreas están establecidas en el Currículum de Referencia, dicha recomendación esta hecha particularmente para carreras especializadas en ingeniería de software, mientras que varias de las carreras relacionadas con Ingeniería en Computación o Informática en Chile abordan la enseñanza de la ingeniería de software de manera más general [6], y particularmente la enseñanza de estos dos áreas de conocimiento se realiza en diversos cursos que no necesariamente son de ingeniería de software.

## Carrera: Ingeniería Civil en Computación v3

### Institución: Universidad de Chile

CONTEXTO GENERAL DE LA ENCUESTA
VISUALIZACIONES DE LA CARRERA

---

Estimados profesores:

Agradecemos de nuevo su participación en la encuesta 2019 de la Iniciativa EDUCA-IS llevada a cabo entre 14/05/2019 y 03/06/2019, cuyo objetivo es recabar información sobre la enseñanza de la ingeniería de software en Chile. Los resultados de la encuesta se utilizarán para crear un mapa nacional del área, y generar un seguimiento anual de este tema dado el impacto que éste tiene en el desarrollo de nuestro país. Esta iniciativa es impulsada por el Departamento de Ciencias de la Computación de la Universidad de Chile.

Esta encuesta que respondieron los profesores de la carrera consistió en 3 fases:

Primera fase:	Segunda fase:	Tercera fase:
<p>En esta fase, se preguntaron datos generales acerca de sus cursos respectivos (número de alumnos al año, cantidad de horas de docencia</p>	<p>Luego de eso, se preguntó cuánto tiempo se dedicaba a la enseñanza de las áreas de conocimiento en forma de porcentajes. Como</p>	<p>Por último, se preguntó cuál era el enfoque teórico y práctico que se le daba a las diferentes áreas de conocimiento.</p>

Figura 5.5: Primer extracto de la página de contexto entregada a los profesores.

Primera fase:	Segunda fase:	Tercera fase:
<p>En esta fase, se preguntaron datos generales acerca de sus cursos respectivos (número de alumnos al año, cantidad de horas de docencia directa tanto teóricas como prácticas, entre otros).</p>	<p>Luego de eso, se preguntó cuánto tiempo se dedicaba a la enseñanza de las áreas de conocimiento en forma de porcentajes. Como pequeño recordatorio, éstas son:</p> <ul style="list-style-type: none"> <li>▪ Fundamentos de la Computación</li> <li>▪ Fundamentos de Ingeniería y Matemática</li> <li>▪ Práctica Profesional</li> <li>▪ Modelamiento y Análisis de Software</li> <li>▪ Análisis y Especificación de Requisitos</li> <li>▪ Diseño de Software</li> <li>▪ Verificación y Validación de Software</li> <li>▪ Proceso de Software</li> <li>▪ Calidad del Software</li> <li>▪ Seguridad</li> </ul>	<p>Por último, se preguntó cuál era el enfoque teórico y práctico que se le daba a las diferentes áreas de conocimiento.</p>

Figura 5.6: Segundo extracto de la página de contexto entregada a los profesores.

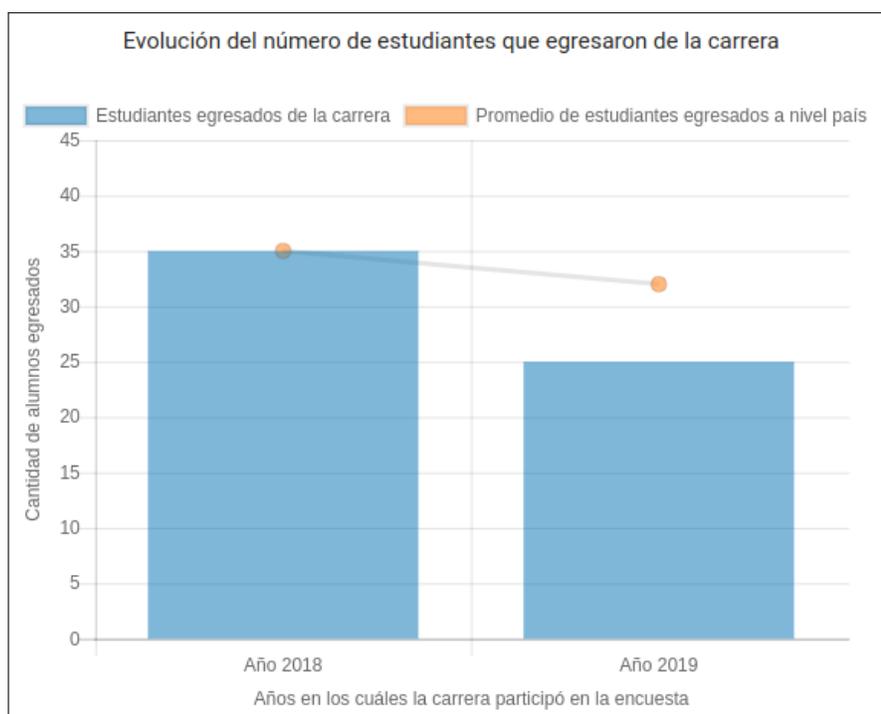


Figura 5.7: En este gráfico, se compara la cantidad de alumnos egresados de una carrera en específico, con la cantidad de alumnos egresados en promedio a nivel de las carreras que participaron en la encuesta.

Como consecuencia de esto, se redujo la cantidad de áreas que un profesor puede responder, lo que a su vez, disminuye el tiempo necesario para responder la encuesta e incrementa la usabilidad del sistema.

### 5.1.5. Segundo ciclo de entrevistas

Habiendo hecho estos cambios, se resumió el proceso de validación y se contactó al profesor Francisco Gutiérrez para llevar a cabo una reunión. Además, se creó un nuevo periodo de encuestas donde se tomaron las encuestas respondidas en el piloto (subsección 5.1.2) y se añadieron 8 encuestas más, correspondientes a cursos pertenecientes a programas de las siguientes universidades: Universidad Técnica Federico Santa María, Universidad Católica Santísima Concepción, y Universidad Austral de Chile.

Las visualizaciones generadas en este periodo corresponden a las figuras 5.7, 5.8, 5.9, y 5.10. La figura 5.7 corresponde al gráfico del análisis de datos **M1**, mientras que la figura 5.8 representa el gráfico del análisis **M2**. La la figura 5.9 representa el gráfico del análisis de datos **M3a**, y, por último, la figura 5.10 corresponde al gráfico del análisis **M3b**.

El protocolo de la entrevista fue el mismo que fue usado (ver subsección 5.1.1) para las entrevistas anteriores. De los comentarios que se hicieron, los más importantes fueron:

#### Usabilidad de los análisis de datos

El profesor entendió la información que los cuatro análisis querían comunicar, y apreció el hecho que hubiese una página de contexto para recordarle a la gente acerca del proceso en

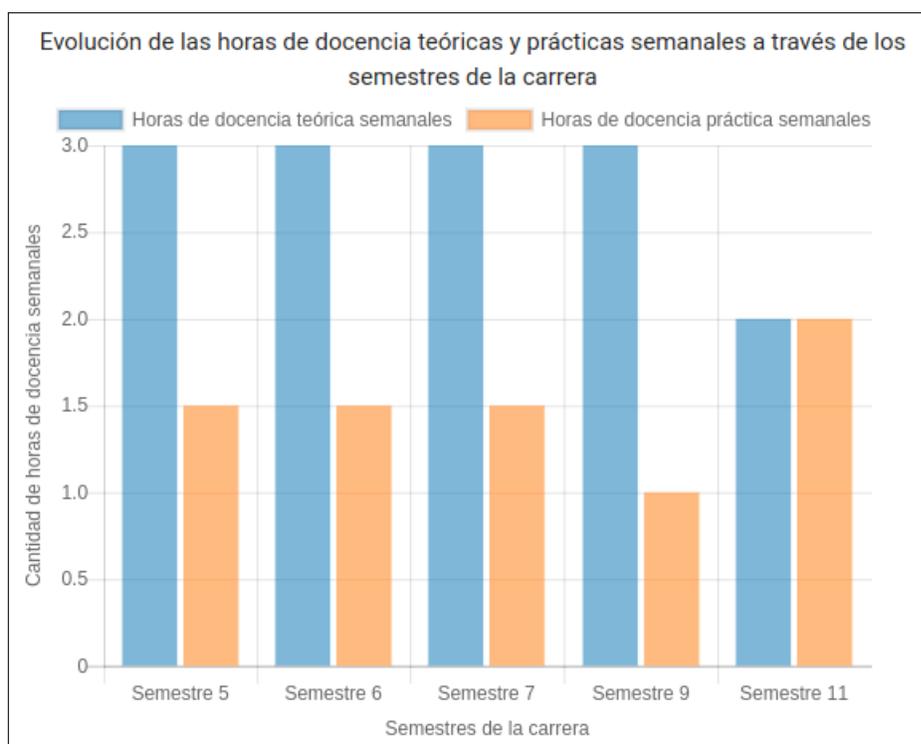


Figura 5.8: En este gráfico, se muestra la evolución de las horas de docencia semanales teóricas y prácticas a través de los semestres de una carrera.

el que habían participado. Uno de los comentarios que él hizo acerca de las visualizaciones de los análisis **M3a** y **M3b** fue que no era muy claro que se podía hacer click en los botones de *Considerar cursos obligatorios* y *Considerar cursos obligatorios y electivos*.

Otro comentario que dio fue el uso del color azul y naranja en la visualización del análisis **M2**. Su comentario se presenta en forma textual a continuación:

*“[...] Cuidado con el uso del color naranja y azul. Primero porque en los gráficos de los análisis **M1**, **M3a** y **M3b** el color azul se usa para hablar de datos acerca de la carrera, mientras que el color naranja se usa para hablar de externos como la ACM u otras carreras. En el gráfico del análisis **M2**, se está usando el azul para hablar de las horas semanales de docencia teóricas, y el naranja para las horas semanales de docencia prácticas. Hay que tener consistencia.”*

Por último, el profesor indicó que el diseño de los cuatro cuadrantes de la página de visualizaciones no era el mejor debido a que, en sus palabras, *“No contaba una historia lógica”*.

### Utilidad de los análisis de datos

En términos de utilidad, el profesor apreció las visualizaciones de los análisis **M3a** y **M3b**, declarando que la utilidad de éstas es muy grande para los programas de Chile. Mencionó además que la utilidad de estos gráficos podría depender del programa. Su comentario específico se presenta a continuación:

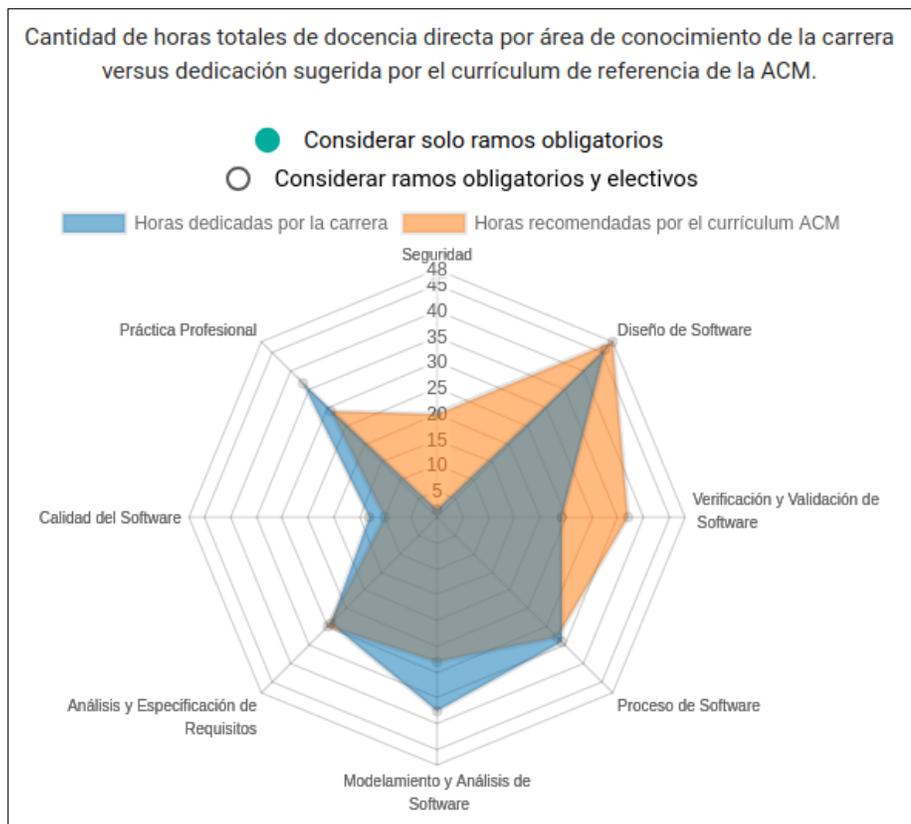


Figura 5.9: En este gráfico, se comparan las horas totales dedicadas a la enseñanza de las áreas de conocimiento de una carrera, con las horas recomendadas por el documento de la ACM.

*“[...] A algunos programas, como el de la Universidad de Chile y la Católica, les puede ser muy útil ver cómo estamos comparados con lo que dice el documento de la ACM. Pero para programas que no estén familiarizados con el tema, como por ejemplo, la Universidad de Magallanes, les puede importar más cómo están en comparación con el promedio en Chile.”*

El profesor agregó, además, que la utilidad de los gráficos de los análisis **M1** y **M2** sufrían debido al diseño de la página, ya que no complementaba lo que la página quería mostrar. Destacó que, si este problema se arreglara, y se tuviese un mejor diseño de la página, le encontraría utilidad a los gráficos mencionados.

Como sugerencias, indicó mejorar el diseño de la página, organizándola por información nacional e información del programa. También sugirió darle a los profesores que imparten cursos en carreras con menos estudiantes, la posibilidad de ver la cantidad de horas que los programas con más estudiantes le dedican a la enseñanza de las diversas áreas de conocimiento.

### 5.1.6. Discusión de los resultados

Si bien solo se hizo una entrevista para este nuevo conjunto final de datos, el hecho que todas las visualizaciones pudiesen ser entendidas, y que se encontrara utilidad en éstas,

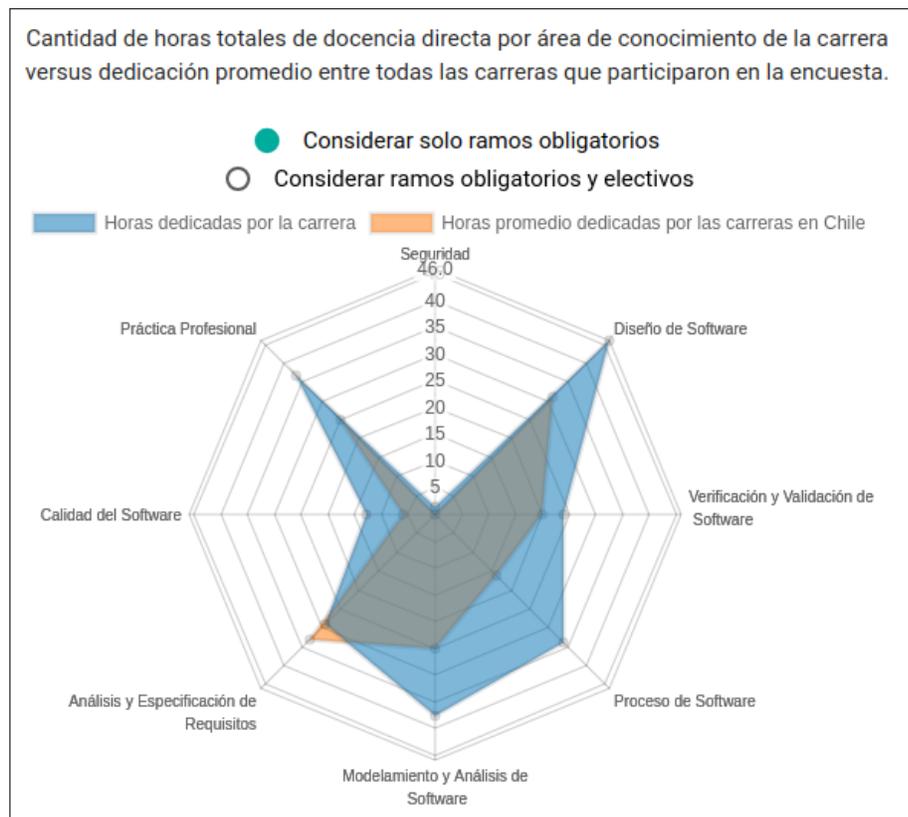


Figura 5.10: En este gráfico, se comparan las horas totales dedicadas a la enseñanza de las áreas de conocimiento de una carrera, con el promedio de horas totales a nivel de las carreras que participaron en la encuesta.

es mucho mejor si lo comparamos con el conjunto de datos anterior. Además, el Profesor Gutiérrez es especialista en visualizaciones y diseño de interfaces de usuario, lo cual hace muy valiosas sus opiniones.

Sin embargo, es bastante claro que el diseño de la página de resultados debe ser hecho de nuevo, para poder darle sentido a la información que se le muestra a los profesores. En vez de solamente mostrarles las visualizaciones, debe haber un hilo conductor que guíe a los profesores y ayude a entender mejor la información entregada por el sistema.

Está claro que aún se requieren más validaciones para determinar la usabilidad y utilidad exacta de las visualizaciones, pero este resultado obtenida ya es una mejora sustancial con respecto a la funcionalidad, usabilidad, utilidad, y los resultados entregados por el sistema legado.

## 5.2. Resumen

En este capítulo se presentaron los resultados de las validaciones hechas con tres académicos del Departamento de Ciencias de la Computación de la Universidad de Chile. Lo que se destaca de éstas es que las primeras dos entrevistas no dieron resultados positivos en términos de usabilidad y utilidad, por lo que fue necesario hacer cambios fundamentales al sistema y realizar una tercera iteración. Con esos cambios hechos, los resultados de la tercera entrevista

fueron bastante positivos en los aspectos antes mencionados. A continuación, se presentarán las conclusiones del trabajo realizado, y el trabajo que puede realizarse a futuro.

# Capítulo 6

## Conclusiones

El objetivo general de esta memoria era mejorar la usabilidad, utilidad, y robustez del Gestor de Recolección de Datos. Esto era importante pues se planea poner este sistema en producción para recolectar datos acerca de la enseñanza de la ingeniería de software en Chile, por lo que mejorar estos aspectos del sistema es imperativo para obtener datos correctos.

El trabajo realizado en esta memoria permitió automatizar la generación de información útil utilizando los datos que los profesores de ingeniería de software ingresan a través de las encuestas. Se rediseñó la manera en la cual se generaban las visualizaciones del sistema, presentando un modelo de datos modificado y usando una librería de gráficos interactiva que elimina el problema de escalabilidad del sistema a largo plazo, debido al limitado manejo de las imágenes que tenía el sistema legado. Además, se reestructuró el reporte generado por el sistema legado, para ofrecer información más precisa y útil acerca de la enseñanza de la ingeniería de software a nivel nacional, institucional, y por carrera, dando así más valor a este recurso.

En base a lo expuesto en esta memoria, se cumplieron los objetivos específicos en su totalidad. A partir del sistema legado, se diseñaron e implementaron distintos análisis y visualizaciones que podrían motivar a los docentes a responder la encuesta. Además, se implementaron mecanismos que automatizan el procesamiento de datos en la forma de reportes de distintos niveles, mejorando así la utilidad del sistema y de los datos que son ingresados actualmente a éste. Por último, se validaron estas funcionalidades con usuarios finales, cumpliendo así ese objetivo.

Es por esto que, basado en lo dicho anteriormente, y lo desarrollado en este trabajo, se cumplió el objetivo general de esta memoria. El sistema está ahora en un estado en el que puede ser lanzado a producción, y que provee información procesada que, según estudios locales, puede motivar a los docentes a responder las encuestas de los cursos que imparten. Se mejoró bastante la usabilidad y utilidad del sistema y de la encuesta en sí, en comparación con el sistema legado. Por último, la robustez del nuevo sistema fue mejorada a través de la automatización del proceso de datos.

Como se mencionó antes, el objetivo principal de este sistema es poder determinar y

cerrar la brecha entre lo que la industria del software quiere de los egresados de carreras de ingeniería de software, y las universidades que imparten estas carreras. Si bien el trabajo realizado provee información acumulada a los docentes que responden la encuesta, es necesario diseñar y agregar más información (visualizaciones, reportes, entre otros) para poder cerrar esta brecha.

La metodología de trabajo usada en este trabajo dio resultados positivos, y el trabajo propuesto se pudo completar en el tiempo fijado. Algo que hubiese sido mejor para complementar esta metodología sería una sesión de validación con usuarios docentes luego de cada iteración. Eso daría más tiempo para darse cuenta de las posibles mejoras de usabilidad y utilidad, y permitiría cerrar las iteraciones de manera más concisa.

El hecho de usar una librería que genera visualizaciones dinámicas en el front-end de la aplicación demostró ser exitoso pues permite generar análisis de datos a largo plazo sin tener que comprometer la integridad del software, ni generar archivos para cada periodo en el cuál se pueda responder la encuesta.

Hacer una validación con expertos en el tema, ya sea docentes del área de ingeniería de software, o expertos en diseño de software, vale la pena pues dio la posibilidad de mejorar las visualizaciones y análisis implementados, ofreciendo un mejor feedback para los docentes que respondiesen la encuesta en un futuro próximo.

Por último, desarrollar este tipo de software para llevar a cabo estas encuestas es válido, ya que este proceso se quiere llevar a cabo periódicamente y siguiendo los mismos pasos (invitar a docentes, esperar a que respondan la encuesta, y luego mandar los reportes correspondientes). Además, tener un software permite guardar la información de periodos anteriores sin tener que crear otras carpetas para guardar datos antiguos. Esto es una funcionalidad que los sistemas como Google Forms no tienen.

## **Trabajo a Futuro**

Existen bastantes trabajos que uno puede abordar para mejorar aún más el sistema. Uno de estos es utilizar los otros Currículums de Referencia elaborados por la ACM/IEEE, para hacer más comparaciones acerca de la enseñanza de ciertas disciplinas, como la Ciencia de la Computación, la Ingeniería en Computación, y Tecnologías de Información de la Computación, en Chile. Usando el mismo proceso de trabajo que se hizo en este trabajo de memoria, se podrían hacer comparaciones entre la enseñanza de las distintas ramas mencionadas anteriormente, abarcando así un espectro de la computación más grande.

Otro hilo de trabajo corresponde a mejorar el diseño de la página de visualizaciones de los análisis de datos para cada carrera, y/o añadir más información que puede ser útil para los profesores. Esto se debe a que las visualizaciones actuales, si bien son útiles en sí, no cuentan una historia o guían al profesor que las ve, de manera que él/ella entienda la información presentada. Para llevar a cabo este trabajo, es necesario diseñar una página donde se puedan ver las visualizaciones, ver qué tipo de información se quiere mostrar (nacional, institucional, entre otros), diseñar nuevos análisis de datos o estadísticas que se quieran mostrar a los profesores, e implementarlas.

Si bien es más simple que los anteriores, un tercer hilo de trabajo puede ser darle la posibilidad a los profesores que han participado en encuestas anteriores, de reutilizar sus respuestas previas en las siguientes ediciones de la encuesta; si es que él/ella participa. Esto ayudaría a mejorar la usabilidad del sistema, y reduciría el tiempo necesario para responder la encuesta, lo que motivaría a la gente a responder la encuesta.

Por último, se puede expandir esta encuesta para distintos países de América Latina, y así obtener información acerca de la enseñanza de la ingeniería de software de manera más amplia. Este hilo de trabajo implica cambiar varios aspectos del sistema, tales como motivar el interés de los profesores de otros países para que participen en la encuesta, cambiar el lenguaje utilizado en las preguntas de la encuesta para que todos los profesores entiendan lo mismo, y diseñar un modelo de datos que soporte universidades de otros países, a diferencia del modelo de datos actual que solo permite universidades de Chile.

# Bibliografía

- [1] Carlo Ghezzi and Dino Mandrioli. The challenges of software engineering education. In *International Conference on Software Engineering*, pages 115–127. Springer, 2005.
- [2] Mary Shaw. Software engineering education: a roadmap. In *ICSE-Future of SE Track*, pages 371–380, 2000.
- [3] Timothy C Lethbridge. What knowledge is important to a software professional? *Computer*, 33(5):44–50, 2000.
- [4] Kathy Beckman, Neal Coulter, Soheil Khajenoori, and Nancy R Mead. Collaborations: closing the industry-academia gap. *IEEE software*, 14(6):49–57, 1997.
- [5] Alex Radermacher and Gursimran Walia. Gaps between industry expectations and the abilities of graduates. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13*, pages 525–530, New York, NY, USA, 2013. ACM.
- [6] Maíra Marques, Sergio F. Ochoa, and María Cecilia Bastarrica. Software Engineering Education in Chile - Status Report. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '16*, pages 180–185, New York, NY, USA, 2016. ACM.
- [7] Mark A Ardis, David Budgen, Gregory W Hislop, Jeff Offutt, Mark J Sebern, and Willem Visser. SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *IEEE Computer*, 48(11):106–109, 2015.
- [8] Emilio Aburto, Gabriel Chandía, Javier Espinoza, Constanza Fierro, Patricio Taiba, and Cristóbal Velásquez. Sistema de recolección de datos sobre la ingeniería de software, Proyecto realizado en el marco del curso CC5401: Ingeniería de Software II. Diciembre 2017. [https://github.com/jaevespinoza/memoria/blob/master/Documento\\_Historicojav.pdf](https://github.com/jaevespinoza/memoria/blob/master/Documento_Historicojav.pdf).
- [9] Juan José Álvarez, Paolo Curotto, Alexis Espinoza, Cristóbal Fuentes, Alejandro Quijada, and Elías Zelada. Estudio de la enseñanza de la ingeniería de software., Proyecto realizado en el marco del curso CC5401: Ingeniería de Software II. Agosto 2018. [https://github.com/jaevespinoza/memoria/blob/master/Documento\\_Historico1.pdf](https://github.com/jaevespinoza/memoria/blob/master/Documento_Historico1.pdf).
- [10] Adrian Holovaty and Jacob Kaplan-Moss. *The definitive guide to Django: Web develop-*

*ment done right.* Apress, 2009.

- [11] John D Hunter. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3):90, 2007.
- [12] Ewa Luger, Stuart Moran, and Tom Rodden. Consent for all: revealing the hidden complexity of terms and conditions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2687–2696. ACM, 2013.
- [13] Iztok Humar, Mirko Gradis, et al. The impact of color combinations on the legibility of a web page text presented on crt displays. *International journal of industrial ergonomics*, 38(11-12):885–899, 2008.
- [14] Nick Downie. Chart.js| Open source HTML5 Charts for your website. <https://www.chartjs.org/>, 2015.
- [15] Nicola Talbot. Creating a PDF document using PDFLaTeX. <http://theoval.cmp.uea.ac.uk/~nlct/latex/pdfdoc/pdfdoc-a4.pdf>, 2004.
- [16] Antonio Melé. *Django By Example*. Packt Publishing Ltd, 2015.
- [17] Carl W Turner, James R Lewis, and Jakob Nielsen. Determining usability test sample size. *International encyclopedia of ergonomics and human factors*, 3(2):3084–3088, 2006.

# Apéndice A

## Apéndice

### A.1. Reporte generado para el periodo de piloto

A continuación, desde la página 71 hasta la 85, se presenta el reporte creado por la última versión del sistema para cada curso que termine su encuesta asociada. En este caso, se muestra el reporte del curso CC4401 - Ingeniería de Software I de la carrera Ingeniería Civil en Computación de la Universidad de Chile.

### A.2. Ejemplos de visualizaciones generadas por la segunda versión del sistema

En esta sección, se muestran dos ejemplos de gráficos generados por la segunda versión del sistema. Se incluyeron estos para poder hacer una comparación de los gráficos de la segunda versión del sistema, con los de la nueva versión. Las figuras A.1 y A.2 corresponden a las visualizaciones de la segunda versión del sistema.

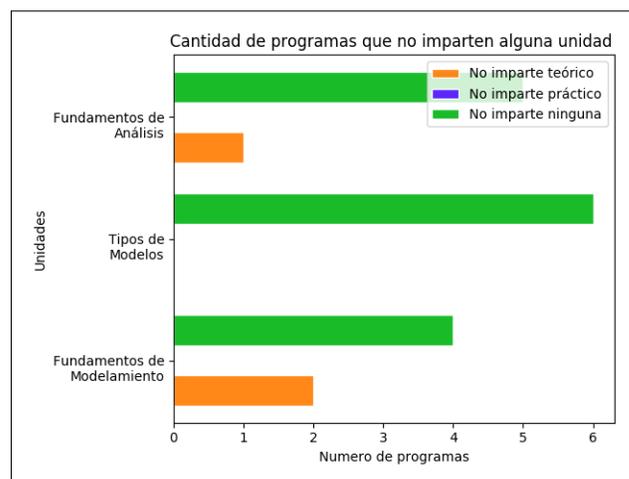


Figura A.1: Gráfico de barras que cuenta la cantidad de programas que no enseñan una unidad de conocimiento.

# **Resultados de la encuesta 2019 de caracterización de la enseñanza de ingeniería de software en Chile**

*Universidad de Chile  
Ingeniería Civil en Computación v3 (Santiago)  
CC4401 - Ingeniería de Software I*

Reporte generado por el grupo EDUCA-IS, del Departamento de Ciencias de la Computación de la Universidad de Chile

<http://gems.dcc.uchile.cl>

Estimado/a Jocelyn Simmonds, agradecemos de nuevo su participación en la encuesta 2019 de la Iniciativa EDUCA-IS llevada a cabo entre 14/05/19 y 03/06/19, cuyo objetivo es recabar información sobre la enseñanza de la ingeniería de software en Chile. Los resultados de la encuesta se utilizarán para crear un mapa nacional del área, y generar un seguimiento anual de este tema dado el impacto que éste tiene en el desarrollo de nuestro país. Esta iniciativa es impulsada por el Departamento de Ciencias de la Computación de la Universidad de Chile.

Este estudio toma como referencia la recomendación: ACM/IEEE Software Engineering Curriculum Guidelines 2014 (SECG-2014)<sup>1</sup>. Esta recomendación establece 10 áreas de la ingeniería de software que deberían enseñarse en carreras de computación e informática. Además, detalla la cantidad de horas de docencia que se le debería dedicar a cada una de dichas áreas.

Este informe de resultados de la encuesta se entrega en tres formatos:

- *Resultados nacionales y regionales.* Esta versión del informe es público y presenta sólo información agregada.
- *Resultados por institución.* Esta versión del informe incluye el anterior, y además presenta información acumulada de una institución particular. Dicho informe se le entrega a los profesores de la institución que participaron en la encuesta, y está disponible para autoridades de dicha institución que lo soliciten.
- *Resultados por curso.* Esta versión del informe incluye el anterior, y además presenta información detallada de los cursos de ingeniería de software que imparte un profesor en particular. Únicamente los profesores participantes en la encuesta reciben el informe relativo a sus cursos.

---

<sup>1</sup><https://www.acm.org/binaries/content/assets/education/se2014.pdf>

## Índice

<b>1. Acerca de la Encuesta</b>	<b>3</b>
1.1. Preguntas de la Encuesta . . . . .	3
<b>2. Datos agregados</b>	<b>4</b>
2.1. Información general acerca de la situación del país . . . . .	4
2.2. Resultados Nacionales por Área de Conocimiento . . . . .	4
2.3. Información general acerca de las regiones del país . . . . .	5
2.4. Resultados regionales por área de conocimiento . . . . .	5
<b>3. Información Acumulada Por Carrera</b>	<b>8</b>
3.1. Resultados Generales de la Carrera . . . . .	8
3.2. Resultados de la Carrera por Área de Conocimiento . . . . .	10
3.3. Información de la Carrera en Comparación al País . . . . .	10
<b>A. Anexo</b>	<b>12</b>
A.1. Fórmulas . . . . .	12
A.2. Detalle del ACM/IEEE Software Engineering Curriculum Guidelines 2014 . . . . .	13
A.3. Profesores Participantes . . . . .	14

## 1. Acerca de la Encuesta

La encuesta se llevó a cabo entre el 17/04/2019 y el 20/04/2019. Se invitaron a profesores de 9 cursos de ingeniería de software, abarcando un total de 1 carreras a lo largo del país. De las 9 invitaciones cursadas, 5 fueron aceptadas, y de éstas se obtuvieron 5 respuestas completas, las cuales se utilizaron para calcular los resultados presentados en este informe. Las respuestas incompletas no fueron consideradas.

### 1.1. Preguntas de la Encuesta

El proceso de encuesta involucró dos pasos, cada uno con preguntas específicas. En el primer paso se buscó caracterizar, de manera general, el curso sobre el cual el profesor estaba entregando información. Para ello se le solicitó al profesor indicar, por ejemplo:

- La duración del curso (en semanas).
- La cantidad de alumnos promedio que participan en el curso.
- La cantidad de horas de docencia directa semanales del curso. Este ítem abarca tanto las horas de cátedra, como las de prácticas y laboratorios.
- La cantidad de créditos (SCT) del curso.
- Si el curso es obligatorio o no.
- El número de semestre en el que se dicta el curso según la malla de la carrera.
- El peso que tiene la nota de las actividades teóricas y prácticas, con respecto a la nota final del curso.

En el segundo paso el profesor entregó información detallada de su curso, particularmente el porcentaje de dedicación que se le asignaba a cada una de las áreas establecidas por la recomendación (SECG-2014) de la ACM/IEEE. Además se les solicitó indicar si el conocimiento en esas áreas se entregaba de forma teórica, práctica, o ambas.

En resumen, las preguntas de la encuesta de un curso se dividieron en 3 categorías:

- Información general del curso: 9 ítems.
- Información del porcentaje de docencia directa en las áreas de conocimiento indicadas en la SECG-2014: 10 ítems.
- Información acerca del enfoque (teórico y/o práctico) usado para enseñar cada área de conocimiento: 8-10 ítems en promedio (depende de la cantidad de áreas que se enseñen en un curso).

## 2. Datos agregados

### 2.1. Información general acerca de la situación del país

En la tabla 1 se muestra información general acerca de la enseñanza de la ingeniería de software (I.S.) a nivel nacional, considerando el conjunto que participó en la encuesta y el período de la misma.

**Tabla 1.** Datos Acumulados a nivel País

Ítem	Cantidad
Cantidad de instituciones que imparten carreras de ingeniería de software	1
Cantidad de programas que enseñan ingeniería de software	1
Cantidad de cursos de ingeniería de software	5
Número de estudiantes que completaron los cursos de I.S. de una carrera	25

### 2.2. Resultados Nacionales por Área de Conocimiento

A continuación, en la tabla 2 se muestra información acumulada acerca de la dedicación a la enseñanza de las áreas de conocimiento (en la Ingeniería de Software) establecidas por la ACM/IEEE en su Currículum Guidelines 2014.

Área de Conocimiento	Promedio de Horas por Carrera	Horas sugeridas por SEGC	Estado
Práctica Profesional	113.3	29	●
Verificación y Validación de Software	83.1	37	●
Análisis y Especificación de Requisitos	68.7	30	●
Proceso de Software	66.9	33	●
Modelamiento y Análisis de Software	62.8	28	●
Fundamentos de la Computación	59.4	152	●
Diseño de Software	51.9	48	●
Fundamentos de Ingeniería y Matemática	48.6	80	●
Calidad del Software	25.6	10	●
Seguridad	6.5	20	●

**Tabla 2.** Dedicación de Horas por Área en el País

La primera columna representa las áreas de conocimiento de la Ingeniería de Software establecidas por la ACM/IEEE. La segunda columna corresponde al promedio de horas a nivel país dedicadas a cada área de conocimiento. La tercera columna corresponde a la cantidad de cursos que imparten esa cierta área.

La columna **Estado** debe interpretarse de la siguiente forma:

- Un círculo rojo significa que la cantidad de horas dedicadas a la enseñanza de un área en particular son menores al 33 % de las horas estándar recomendadas por el Currículum de referencia de la ACM/IEEE.
- Un círculo amarillo significa que las horas dedicadas a la enseñanza de un área en particular son mayores al 33 %, pero menores al 66 % de las horas estándar recomendadas por el Currículum de referencia de la ACM/IEEE.
- Un círculo rojo significa que las horas dedicadas a la enseñanza de un área en particular son mayores al 66 % de las horas estándar recomendadas por el Currículum de referencia de la ACM/IEEE.
- Un círculo negro significa que no hay respuestas asociadas a una carrera en específico.

Para obtener el promedio de horas de una cierta área de conocimiento, se calcula la cantidad de horas totales para cada encuesta, sumando la cantidad de horas de docencia teóricas y prácticas, y luego multiplicándolas por la cantidad de semanas de clase declaradas en los cursos. La fórmula 1 ilustra cómo obtener este resultado.

Luego de calcular esto, se tomaron las horas totales de cada encuesta y se les multiplicó el porcentaje de dedicación para esa área de conocimiento que fue declarado en la encuesta. Sumando todos esos valores, se divide la cantidad de horas por la cantidad de encuestas. La fórmula 2 del anexo explica cómo obtener este dato.

### **2.3. Información general acerca de las regiones del país**

Habiendo mostrado los datos agregados nacionales, a continuación se expone la distribución de éstos a través de las regiones. La tabla 3 muestra los datos de la tabla 1, haciendo un desglose por cada región que está ingresada en la encuesta.

Nota: La suma de las instituciones puede ser mayor a la de la tabla 1 debido a que ciertas instituciones (ej: la Universidad Católica del Norte) están presentes en más de una región.

### **2.4. Resultados regionales por área de conocimiento**

En la tabla 4 se muestra la distribución de horas totales dedicadas a la enseñanza de las áreas de conocimiento por región del país. El objetivo de esta tabla es dar a conocer qué tan distinta la enseñanza de las áreas de conocimiento es a través de las regiones del país:

Cada columna representa un región del país y la cantidad de horas dedicadas a la enseñanza de un área de conocimiento en particular.

Para calcular las horas dedicadas a un área de conocimiento por región, se tomaron las carreras pertenecientes a cada región. Por cada carrera, se escogen los cursos que enseñan cierta área y se

Variable	Regiones											
	Antofagasta	Araucanía	Arica y Parinacota	Atacama	Bíobío	Coquimbo	Los Ríos	Maule	Metropolitana	Tarapacá	Valparaíso	Nuble
Cantidad de Instituciones que imparten carreras de I.S.	0	0	0	0	0	0	0	0	1	0	0	0
Cantidad de programas que enseñan ingeniería de software	0	0	0	0	0	0	0	0	1	0	0	0
Cantidad de cursos de ingeniería de software	0	0	0	0	0	0	0	0	5	0	0	0
Número de estudiantes que completaron los cursos de I.S. de una carrera	0	0	0	0	0	0	0	0	25	0	0	0

**Tabla 3.** Distribución de Datos Generales por Regiones

	Regiones											
	Antofagasta	Araucanía	Arica y Parinacota	Atacama	Bío Bio	Cochilco	Los Ríos	Maule	Metropolitana	Tarapacá	Valparaíso	Nuble
Area de Conocimiento	-	-	-	-	-	-	-	-	-	-	-	-
Análisis y Especificación de Requisitos	-	-	-	-	-	-	-	-	68.7	-	-	-
Calidad del Software	-	-	-	-	-	-	-	-	25.6	-	-	-
Diseño de Software	-	-	-	-	-	-	-	-	51.8	-	-	-
Fundamentos de Ingeniería y Matemática	-	-	-	-	-	-	-	-	48.6	-	-	-
Fundamentos de la Computación	-	-	-	-	-	-	-	-	59.4	-	-	-
Modelamiento y Análisis de Software	-	-	-	-	-	-	-	-	62.8	-	-	-
Proceso de Software	-	-	-	-	-	-	-	-	66.9	-	-	-
Práctica Profesional	-	-	-	-	-	-	-	-	113.2	-	-	-
Seguridad	-	-	-	-	-	-	-	-	6.5	-	-	-
Verificación y Validación de Software	-	-	-	-	-	-	-	-	83.0	-	-	-

**Tabla 4.** Distribución de Horas Por Área en cada Región

multiplica el porcentaje de dedicación a la enseñanza de esa área por las horas totales del curso. Luego de calcular todos estos valores, se suman esas cantidades. La fórmula 4 del anexo explica esta operación.

Nota:

- El símbolo  $\in$  significa *pertenece a cierto conjunto*. En este caso, encuestas que pertenezcan a una cierta región.
- “-” significa que una región, carrera o institución no tiene encuestas asociadas para este periodo.

### 3. Información Acumulada Por Carrera

#### 3.1. Resultados Generales de la Carrera

Los cursos pertenecientes a la carrera *Ingeniería Civil en Computación v3* que participaron en la encuesta llevada a cabo en el periodo 14/05/19 - 03/06/19 se muestran en la tabla 5.

Nombre del Curso	Nombre del Profesor
Ingeniería de Software II	Sergio Ochoa
Diseño de Sistemas Interactivos	Francisco Gutierrez
Proyecto de Software	Cecilia Bastarrica
Metodologías de Diseño y Programación	Alexandre Bergel
Ingeniería de Software I	Jocelyn Simmonds

**Tabla 5.** Nombre de los cursos y profesores de la carrera que respondieron la encuesta en su completitud.

Tomando los cursos mencionados en la tabla 5, se muestra en la tabla 6 la información general de todos los cursos de la carrera Ingeniería Civil en Computación v3 perteneciente a la institución Universidad de Chile.

Las columnas de la tabla 6 indican los distintos cursos que pertenecen a la carrera Ingeniería Civil en Computación v3. Éstos son:

**CS1** Diseño de Sistemas Interactivos

**CS2** Ingeniería de Software I

**CS3** Ingeniería de Software II

**CS4** Metodologías de Diseño y Programación

**CS5** Proyecto de Software

Variable	Cursos				
	CS1	CS2	CS3	CS4	CS5
Semanas de Clase al Semestre	15	15	15	15	18
Número Promedio de Estudiantes del Curso	10	90	20	100	25
Horas Totales de Docencia Teórica	3.00	3.00	3.00	3.00	2.00
Horas Totales de Docencia Práctica	1.50	1.50	1.00	1.50	16.00
Cantidad de Créditos del Curso (SCT)	6	6	6	6	20
Peso de las de Actividades Teóricas	0	50	30	70	0
Peso de las de Actividades Prácticas	100	50	70	30	100
Tipo de Curso	Electivo	Obligatorio	Obligatorio	Obligatorio	Obligatorio

**Tabla 6.** Información general acerca de los cursos pertenecientes a la carrera Ingeniería Civil en Computación v3.

### 3.2. Resultados de la Carrera por Área de Conocimiento

Habiendo obtenido información de los cursos de la carrera, se calculó la cantidad de horas que cada curso le dedicaba a la enseñanza de las áreas de conocimiento establecidas por el Curriculum Guidelines. La tabla 7 muestra los resultados de este cálculo. La columna CS1 representa las horas que el curso CC5510 Diseño de Sistemas Interactivos le dedica a las áreas de conocimiento de la ACM. La columna CS2 representa las horas que el curso CC4401 Ingeniería de Software I le dedica a las áreas de conocimiento de la ACM. La columna CS3 representa las horas que el curso CC5401 Ingeniería de Software II le dedica a las áreas de conocimiento de la ACM. La columna CS4 representa las horas que el curso cc3002 Metodologías de Diseño y Programación le dedica a las áreas de conocimiento de la ACM. La columna CS5 representa las horas que el curso cc5402 Proyecto de Software le dedica a las áreas de conocimiento de la ACM.

### 3.3. Información de la Carrera en Comparación al País

Tomando los datos de la primera parte de la encuesta, se calcularon los promedios de los ítemes preguntados para la carrera correspondiente. Luego de eso, se tomaron los promedios de estos valores a nivel país y se compararon con el promedio de la carrera. La tabla 8 muestra esta comparación.

Variable	Promedio de la Carrera	Promedio del País
Semanas de Clase al Semestre	16	15.6
Horas de Docencia Teórica Semanal	2.8	2.8
Horas de Docencia Práctica Semanal	4.3	4.3
Cantidad de Alumnos por Curso	49	49
Cantidad de Créditos (SCT)	9	9
Peso de Actividades Teóricas	30.0 %	30.0 %
Peso de Actividades Prácticas	70.0 %	70.0 %

**Tabla 8.** Características Generales de la Carrera y Curso

	Cursos				
	CS1	CS2	CS3	CS4	CS5
Áreas de Conocimiento					
Análisis y Especificación de Requisitos	27.0	13.5	12.0	0	16.2
Calidad del Software	0	3.4	6.0	0	16.2
Diseño de Software	0	6.8	12.0	16.9	16.2
Fundamentos de Ingeniería y Matemática	0	0	0	0	48.6
Fundamentos de la Computación	0	10.1	0	16.9	32.4
Modelamiento y Análisis de Software	0	13.5	0	16.9	32.4
Proceso de Software	0	6.8	18.0	0	42.1
Práctica Profesional	13.5	6.8	12.0	0	81.0
Seguridad	0	0	0	0	6.5
Verificación y Validación de Software	27.0	6.8	0	16.9	32.4

**Tabla 7.** Horas totales dedicadas a cada área de conocimiento por curso.

La primera columna representa las distintas variables que se preguntaron en la encuesta. La segunda columna corresponde al promedio entre las variables de cada encuesta perteneciente a la carrera. La tercera columna representa el promedio entre las variables de todas las encuestas en total.

Para calcular los valores promedio del carrera, se tomaron las encuestas asociadas a la carrera y se suman los datos correspondientes, dividiendo el resultado por la cantidad de encuestas asociadas.

Para calcular los valores promedio del país, se tomaron todas las encuestas, sumando los datos y dividiéndolos por la cantidad de encuestas totales.

## A. Anexo

### A.1. Fórmulas

- 1) Fórmula para obtener las horas totales en un semestre de un curso:

$$horas_{totales} = semanas \times (horas_{teóricas} + horas_{prácticas})$$

- 2) Fórmula para obtener el promedio de horas dedicadas a un área de conocimiento en el país:

$$promedio_{area} = \frac{\sum_{encuestas} (horas_{totales} \times \%area_{dedicacion})}{cantidad(encuestas)}$$

- 3) Fórmula para obtener las horas totales dedicadas a un área de conocimiento en el país:

$$horas_{area} = \sum_{encuestas} (horas_{totales} \times \%area_{dedicacion})$$

- 4) Fórmula para obtener las horas totales dedicadas a un área de conocimiento en una región específica:

$$hora_{area_{region}} = \sum_{encuestas \in region} (horas_{totales} \times \%area_{dedicacion})$$

- 5) Fórmula para obtener las horas totales dedicadas a un área de conocimiento en una carrera específica:

$$hora_{carrera_{area}} = \sum_{encuestas \in carrera} (horas_{totales} \times \%area_{dedicacion})$$

## A.2. Detalle del ACM/IEEE Software Engineering Curriculum Guidelines 2014

KA/KU	Title	Hours	KA/KU	Title	Hours
<b>CMP</b>	<b>Computing essentials</b>	<b>152</b>	<b>DES</b>	<b>Software design</b>	<b>48</b>
CMP.cf	Computer science foundations	120	DES.con	Design concepts	3
CMP.ct	Construction technologies	20	DES.str	Design strategies	6
CMP.tl	Construction tools	12	DES.ar	Architectural design	12
			DES.hci	Human-computer interaction design	10
			DES.dd	Detailed design	14
			DES.ev	Design evaluation	3
<b>FND</b>	<b>Mathematical and engineering fundamentals</b>	<b>80</b>	<b>VAV</b>	<b>Software verification and validation</b>	<b>37</b>
FND.mf	Mathematical foundations	50	VAV.fnd	V&V terminology and foundations	5
FND.ef	Engineering foundations for software	22	VAV.rev	Reviews and static analysis	9
FND.ec	Engineering economics for software	8	VAV.tst	Testing	18
			VAV.par	Problem analysis and reporting	5
<b>PRF</b>	<b>Professional practice</b>	<b>29</b>	<b>PRO</b>	<b>Software process</b>	<b>33</b>
PRF.psy	Group dynamics and psychology	8	PRO.con	Process concepts	3
PRF.com	Communications skills (specific to SE)	15	PRO.imp	Process implementation	8
PRF.pr	Professionalism	6	PRO.pp	Project planning and tracking	8
			PRO.cm	Software configuration management	6
			PRO.evo	Evolution processes and activities	8
<b>MAA</b>	<b>Software modeling and analysis</b>	<b>28</b>	<b>QUA</b>	<b>Software quality</b>	<b>10</b>
MAA.md	Modeling foundations	8	QUA.cc	Software quality concepts and culture	2
MAA.tm	Types of models	12	QUA.pca	Process assurance	4
MAA.af	Analysis fundamentals	8	QUA.pda	Product assurance	4
<b>REQ</b>	<b>Requirements analysis and specification</b>	<b>30</b>	<b>SEC</b>	<b>Security</b>	<b>20</b>
REQ.rfd	Requirements fundamentals	6	SEC.sfd	Security fundamentals	4
REQ.er	Eliciting requirements	10	SEC.net	Computer and network security	8
REQ.rsd	Requirements specification and documentation	10	SEC.dev	Developing secure software	8
REQ.rv	Requirements validation	4			

Para mayor detalle del estándar usado puede descargar el documento completo<sup>2</sup> desde la página de la ACM. También puede seguir este link para ver otros estándares curriculares de la misma organización.

<sup>2</sup><https://www.acm.org/binaries/content/assets/education/se2014.pdf>

### A.3. Profesores Participantes

En total participaron 5 profesores de 1 instituciones. A continuación, se indica la lista de profesores por institución que aceptaron dar visibilidad de su participación en esta encuesta.

Institución	Profesores
Universidad de Chile	Jocelyn Simmonds
	Cecilia Bastarrica
	Sergio Ochoa
	Alexandre Bergel
	Francisco Gutierrez

**Tabla 9.** Nombres de profesores que aceptaron dar visibilidad de su nombre.

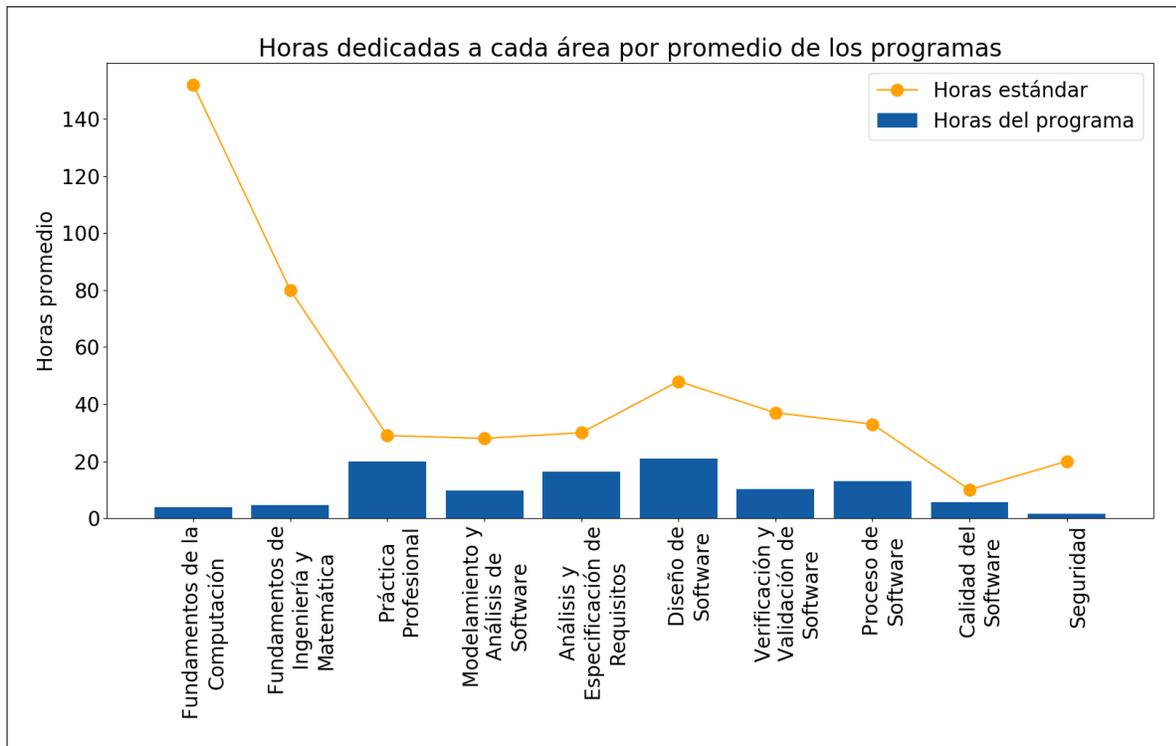


Figura A.2: Gráfico de barras que compara el promedio de cantidad de horas totales a nivel de país con las horas recomendadas por el documento de la ACM.

### A.3. Interactividad de las visualizaciones

En esta sección, se detalla la información mostrada a los usuarios al momento de interactuar con las visualizaciones implementadas en el capítulo 4. La figura A.3 muestra la información que el primer gráfico detalla, mientras que la figura A.4 muestra lo mismo, pero para el segundo gráfico. Las figuras A.5 y A.6 muestran la información que detallan al momento de interactuar con la tercera y cuarta visualización respectivamente.

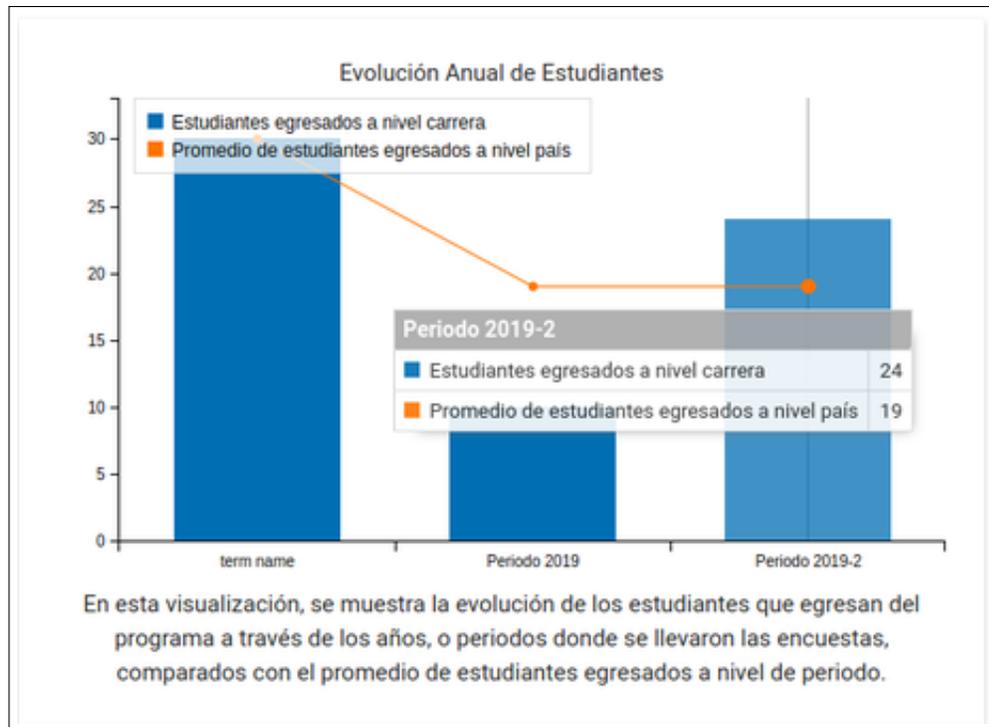


Figura A.3: Información mostrada al usuario al momento de mover el mouse cerca de las barras de la visualización.

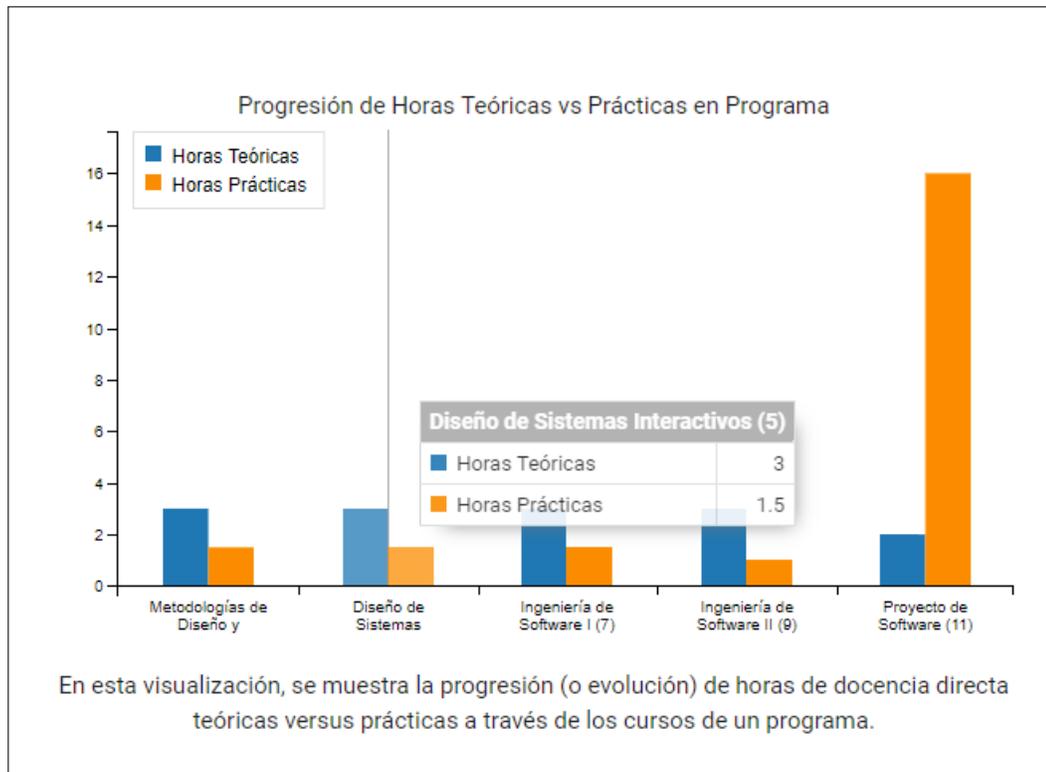


Figura A.4: Información mostrada al usuario al momento de mover el mouse cerca de las barras de la visualización.

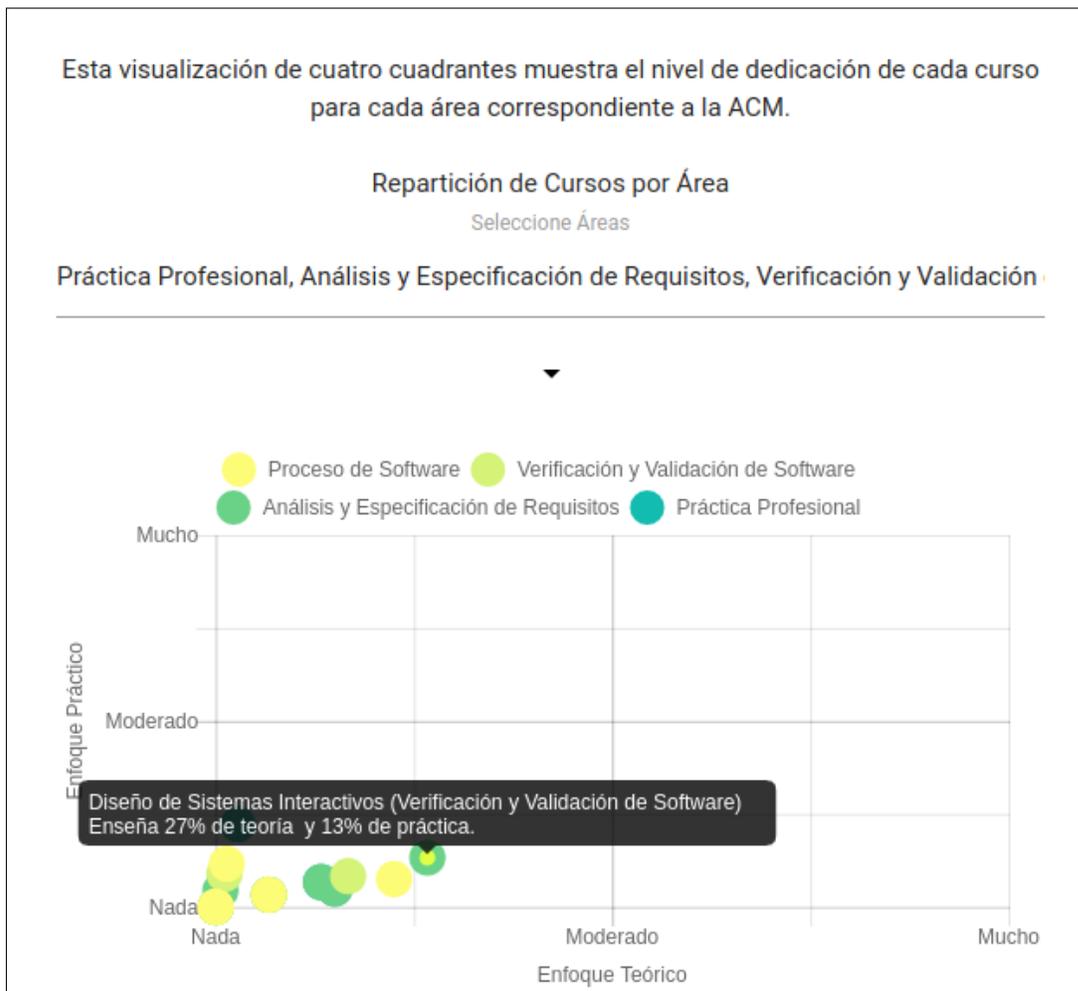


Figura A.5: Información mostrada al usuario al momento de mover el mouse cerca de los puntos de la visualización.

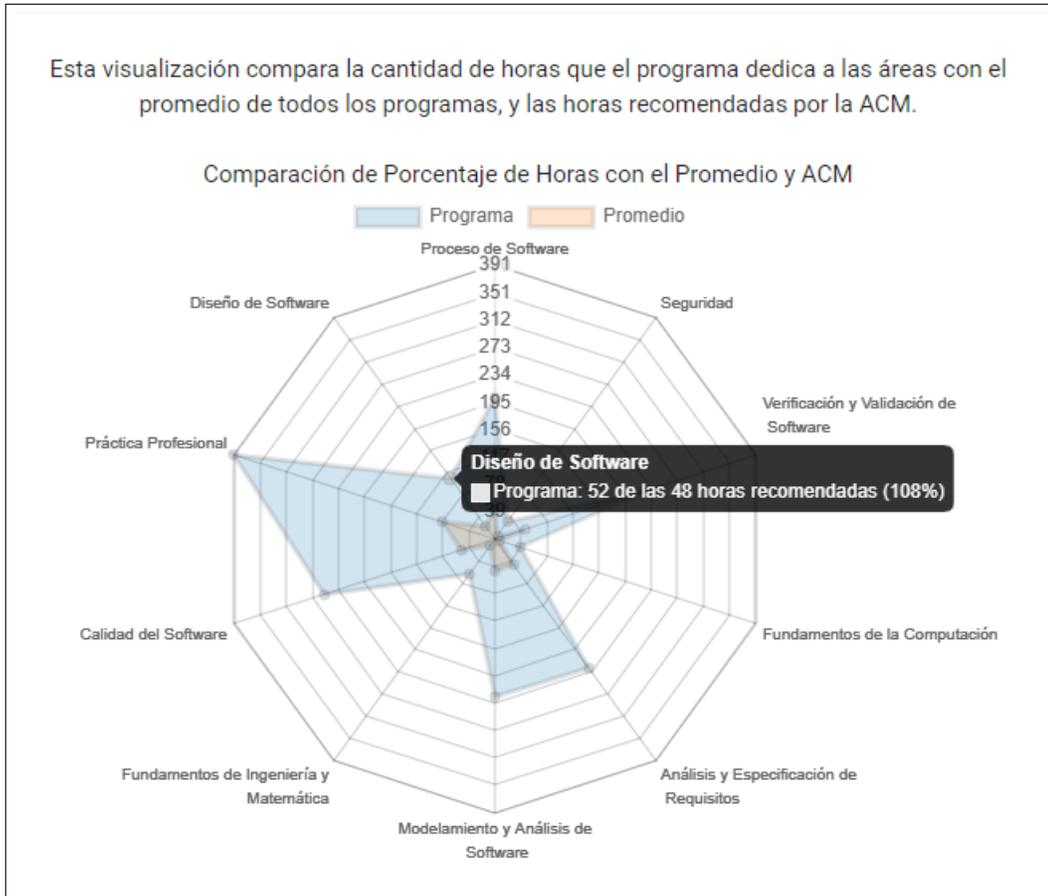


Figura A.6: Información mostrada al usuario al momento de mover el mouse cerca de los puntos de la visualización.