



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

DISEÑO E IMPLEMENTACIÓN DE SISTEMA WEB PARA
RENDICIÓN DE ENSAYOS SIMCE Y PSU

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN
COMPUTACIÓN

ALEXIS GABRIEL HENRÍQUEZ DEL CANTO

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

JAIME SÁNCHEZ ILABACA

JUAN ÁLVAREZ RUBIO

SANTIAGO DE CHILE

Septiembre 2013

Resumen

Ediciones SM es una institución dedicada a la elaboración de material educativo y divulgación de literatura infantil y juvenil. Con el propósito de colaborar con la labor docente y lograr la fidelización de sus principales clientes -los establecimientos educacionales- ha generado un conjunto de iniciativas de apoyo a la gestión académica, entre las que se incluye un programa de evaluaciones SIMCE y PSU externas en sus colegios asociados. Si bien este servicio complementario ha sido muy bien recibido por las instituciones educacionales, su penetración es limitada debido a la naturaleza manual del proceso, consistente en generar los ensayos impresos, enviar personal docente calificado a los distintos establecimientos para realizar la aplicación de la evaluación, procesar los resultados con Excel y otras herramientas y finalmente elaborar los informes para cada establecimiento.

Es en este contexto que se desarrolló el proyecto “Diseño e Implementación de Sistema Web para Rendición de Ensayos SIMCE y PSU”, con el objetivo de ampliar la actual cobertura del servicio de evaluaciones externas de la editorial, y reducir tiempos y costos ligados al proceso.

El sistema construido se trata de una aplicación web desarrollada con el Framework MVC CakePHP, desplegada en servidores Linux sobre la nube de Amazon. Ésta permite a los usuarios de la editorial, crear colegios y definir sus permisos para determinadas evaluaciones, generar bancos de preguntas, construir pruebas y definir periodos de evaluación; a los alumnos, rendir ensayos SIMCE y PSU de forma rápida y fácil, con la posibilidad de visualizar los resultados apenas finaliza la evaluación; y a los profesores, obtener reportes de rendimiento considerando parámetros específicos de cada evaluación, gráficos comparativos e informes con distintos niveles de agregación, lo que le permite tener una visión de todos los cursos, un curso en particular o un alumno específico.

Para la editorial, la inclusión de la aplicación desarrollada en su oferta tecnológica ha significado un importante paso, más considerando que se trata de una herramienta que apunta a reemplazar completamente al servicio tradicional de evaluaciones externas. Si bien los resultados han sido satisfactorios, no son definitivos, pues se espera que la aplicación crezca según los nuevos desafíos que se vayan planteando

Al día de hoy, la aplicación ha recibido más de 5.000 alumnos, con un total cercano a las 17.000 evaluaciones rendidas, y ha logrado la eliminación del proceso manual de elaboración de informes, la inmediatez de los resultados para profesores y alumnos, el análisis automático de los instrumentos de evaluación y una tangible reducción de los costos ligada a la eliminación de gran parte de la logística del servicio tradicional.

Además, ya se cuentan con planes de expansión de la plataforma para adicionar nuevas funcionalidades para el año 2014, siendo la más importante el *Constructor automático de pruebas para profesores*, herramienta que permitirá a los profesores generar rápidamente una prueba impresa según sus necesidades.

Agradezco a mi familia: Malva, Gabriel y Karol; por su amor y apoyo incondicionales. Simplemente los amo.

Tabla de contenido

Resumen	i
Agradecimientos.....	ii
1 Introducción.....	1
1.1 Motivación.....	1
1.2 Principales desafíos	2
1.3 Objetivos de la memoria	3
2 Marco teórico.....	4
2.1 Estructura de las evaluaciones	4
2.2 Análisis de preguntas.....	6
2.3 Proceso tradicional de toma de evaluaciones.....	7
2.4 Arquitectura de una aplicación web.....	8
2.4.1 Modelo.....	9
2.4.2 Vista.....	9
2.4.3 Controlador	9
3 Diseño general de la aplicación.....	11
3.1 Requisitos de la solución	11
3.2 Arquitectura física de la solución	13
3.3 Arquitectura lógica de la solución	14
3.4 Herramientas de desarrollo.....	17
4 Diseño detallado de la aplicación.....	18
4.1 Componente control de acceso	18
4.2 Componente mensajería.....	18
4.3 Componente generación gráficos e informes.....	19
4.4 Módulo administración.....	20
4.4.1 Administración de colegios y profesores	20
4.4.2 Mantenedores de disciplinas, habilidades y ejes temáticos.....	21
4.4.3 Definición de períodos de evaluación.....	21
4.5 Módulo edición.....	22
4.6 Módulo profesores	23
4.6.1 Creación de cursos	24
4.6.2 Calendarización de evaluaciones.....	24
4.6.3 Revisión y descarga de resultados	25

4.7	Módulo alumnos.....	26
5	Implementación de la solución.....	28
5.1	Componente control de acceso	28
5.2	Componente mensajería	30
5.3	Componente generación gráficos e informes.....	31
5.4	Módulo administración.....	32
5.4.1	Administración de colegios y profesores	32
5.4.2	Mantenedores de disciplinas, habilidades y ejes temáticos.....	35
5.4.3	Definición de períodos de evaluación.....	36
5.5	Módulo edición.....	38
5.6	Módulo profesores.....	40
5.6.1	Creación de cursos	40
5.6.2	Calendarización de evaluaciones.....	42
5.6.3	Revisión y descarga de resultados	43
5.7	Módulo alumnos.....	47
5.8	Consideraciones de rendimiento	48
6	Resultados y conclusiones	50
6.1	Ampliación de la cobertura de evaluación	50
6.2	Eliminación del proceso de elaboración de informes e inmediatez de los resultados	51
6.3	Análisis automático de los instrumentos de evaluación	52
6.4	Reducción de costos	52
6.5	Conclusiones y trabajo a futuro	52
7	Bibliografía	54

Índice de tablas

Tabla 1: Interpretación índice de dificultad.....	7
Tabla 2: Interpretación índice de homogeneidad.....	7
Tabla 3: Mensajes enviados por la aplicación.....	19

Índice de figuras

Figura 1: Ciclo de una petición en CakePHP.....	8
Figura 2: Gráfico "Logro por Eje Temático".....	12
Figura 3: Gráfico "Logro por Curso".....	12
Figura 4: Arquitectura física de la aplicación.....	13
Figura 5: Anatomía de la aplicación.....	14
Figura 6: Usuarios de la aplicación.....	14
Figura 7: Componentes de la aplicación.....	15
Figura 8: Módulos de la aplicación.....	15
Figura 9: Modelo de datos simplificado.....	16
Figura 10: Modelo de datos simplificado, componente manejo de usuarios.....	18
Figura 11: Proceso de control de acceso.....	18
Figura 12: Sección modelo de datos, componente mensajería.....	19
Figura 13: Componente generación gráficos e informes.....	19
Figura 14: Modelo de datos simplificado, módulo administración.....	20
Figura 15: Modelo de datos simplificado, módulo edición.....	22
Figura 16: Modelo de datos simplificado, módulo profesores.....	23
Figura 17: Formato archivo de importación de alumnos en Excel.....	24
Figura 18: Proceso de carga de alumnos.....	24
Figura 19: Estructura de resultados de evaluaciones.....	25
Figura 20: Modelo de datos simplificado, módulo alumnos.....	26
Figura 21: Pantalla de ingreso.....	28
Figura 22: Menú de usuario.....	29
Figura 23: Ciclo extendido de una petición en CakePHP.....	29
Figura 24: Correo enviado por la aplicación.....	31
Figura 25: Formulario agregar colegio.....	32
Figura 26: Listado de colegios.....	33
Figura 27: Asignación de permisos a colegio.....	34
Figura 28: Formulario agregar disciplina.....	35
Figura 29: Listado de disciplinas.....	35
Figura 30: Formulario definición de período de evaluación.....	36
Figura 31: Listado de períodos de evaluación.....	36
Figura 32: Resultados detallados de un período de evaluación en Excel.....	37
Figura 33: Formulario de creación/edición de preguntas.....	38
Figura 34: Vista previa de una prueba.....	39
Figura 35: Detalle pregunta e indicadores de validación.....	40

Figura 36: Formulario de ingreso de alumnos mediante archivo Excel.....	41
Figura 37: Listado de cursos.....	41
Figura 38: Detalle de un curso	42
Figura 39: Formulario de cambio de fecha	42
Figura 40: Resultados generales a nivel de curso.....	43
Figura 41: Gráfico de resultados detallados a nivel de curso.....	44
Figura 42: Resultados detallados de un curso a nivel alumno.....	45
Figura 43: Informe en PDF	46
Figura 44: Pantalla de ingreso alumno.....	47
Figura 45: Pantalla de rendición de prueba.....	47
Figura 46: Resultados de un alumno	48
Figura 47: Gráfico pruebas rendidas por día.....	50
Figura 48: Gráfico tiempo promedio de resolución por período de evaluación.....	51
Figura 49: Gráfico porcentaje de logro promedio por período de evaluación.....	51

Índice de extractos de código

Extracto de código 1: Menú perfil profesor (coordinador_evaluaciones.yml).....	29
Extracto de código 2: Control de acceso (app_controller.php).....	30
Extracto de código 3: Definición de eventos para envío de mensajes (config.php).....	30
Extracto de código 4: Utilización de EmilioComponent para el envío de mensajes (emilio.php) .	31
Extracto de código 5: Helper para renderización de gráficos (charts.php).....	31
Extracto de código 6: Validación a nivel de modelo (school.php)	33
Extracto de código 7: Llamada AJAX para activar/desactivar permisos a un colegio (levels.js).34	
Extracto de código 8: Consulta SQL para descarga de resultados de un período (test_run.php)37	
Extracto de código 9: Lectura archivo Excel (school.php)	41
Extracto de código 10: Conversión HTML a PDF (wk_html_to_pdf.php)	46

1 Introducción

Ediciones SM, empresa editorial con fuerte presencia en Iberoamérica y en particular en Chile, es una institución dedicada a la elaboración de material educativo y divulgación de literatura infantil y juvenil. Con el propósito de colaborar con la labor docente y lograr la fidelización de sus principales clientes en el país -los establecimientos educacionales- ha generado un conjunto de iniciativas de apoyo a la gestión académica entre las que se encuentran talleres a profesores y alumnos, facilitación de herramientas tecnológicas a colegios y generación de material informativo adicional, entre otras. Es en este contexto que se enmarca el proyecto y propuesta de trabajo de título, “Diseño e Implementación de Sistema Web para Rendición de Ensayos SIMCE y PSU”, aplicación web que permite a los alumnos de los establecimientos asociados a la editorial, rendir ensayos SIMCE y PSU de forma rápida y fácil, con resultados inmediatos, y a los profesores, obtener reportes de rendimiento considerando parámetros específicos de cada evaluación, gráficos comparativos e informes con distintos niveles de agregación. La ejecución de este proyecto fue solicitada a Idea Uno Ltda., empresa de desarrollo de software en la que el alumno se desempeña desde el año 2011.

1.1 Motivación

Actualmente Ediciones SM a través de su Instituto IDEA, organismo orientado principalmente a la investigación educativa y al desarrollo profesional docente, ha llevado a cabo con gran éxito un programa de evaluaciones externas independientes en establecimientos asociados a la editorial, con el objetivo de identificar fortalezas y debilidades para generar información relevante en la toma de decisiones académicas, proponer estrategias pedagógicas y asesorar a dichos establecimientos. Si bien este servicio complementario ha sido muy bien recibido por las instituciones educacionales, su penetración es limitada debido a la naturaleza manual del proceso, consistente en generar los ensayos impresos, enviar personal docente calificado a los distintos establecimientos para realizar la aplicación de la evaluación, procesar los resultados con Excel y otras herramientas como SPSS y finalmente elaborar los informes para cada establecimiento. Considerando que Ediciones SM abarca un importante porcentaje de los más de 12.000 colegios existentes en Chile, el proceso antes descrito hace inviable la ampliación de la cobertura de aplicación, por lo que surge la necesidad de encontrar un mecanismo escalable, de fácil llegada a los establecimientos y que permita un rápido procesamiento de los resultados.

Considerando la problemática expuesta, la solución debe ser tal que minimice los costos en logística, es decir, que evite el desplazamiento de personal a los colegios objetivo, que permita una gran flexibilidad a profesores y alumnos para escoger el momento de evaluación, que proporcione de forma automática resultados e informes y que sea de fácil uso para promover su adopción; con lo que el desarrollo de una aplicación web aparece como la mejor alternativa, permitiendo satisfacer todos los requisitos planteados.

Si bien existen herramientas que permiten la realización de evaluaciones online, como Moodle con Quiz Module [1], o ProProfs.com [2], estas no soportan la estructura de negocio que la editorial necesita, por ejemplo, considerando informes automáticos con parámetros específicos derivados de la malla curricular para cada pregunta, la generación de permisos a nivel de evaluación para los

distintos establecimientos o la autonomía de los colegios para cargar sus propios alumnos y de forma sencilla calendarizar evaluaciones.

1.2 Principales desafíos

La construcción de una aplicación que cumpla con lo descrito plantea una serie de desafíos técnicos, que consideran aspectos de comprensión del negocio, generación de informes automáticos y rendimiento acorde a la cantidad de usuarios esperada. A continuación se presenta una enumeración de los principales desafíos detectados:

1. **Comprensión de la estructura de las evaluaciones:** Uno de los primeros puntos a analizar en la etapa de diseño del software es la estructura de una evaluación. Una evaluación es entendida como un conjunto de enunciados que tienen a su vez un conjunto de ítems. Cada ítem está formado por una pregunta, un número arbitrario de alternativas (por lo general entre 4 y 5, donde una es la correcta y el resto son distractores) y una serie de atributos que soportan el análisis que finalmente se presenta en los informes, como la temática que la pregunta evalúa dentro de la disciplina (por ejemplo: temática *Significado de palabras por contexto*, para la disciplina *Lenguaje*) o la habilidad requerida por el alumno para responder correctamente dicha pregunta (por ejemplo las habilidades: *Analizar*, *Inferir* o *Explicar*). La comprensión de la estructura de las evaluaciones es la base para la construcción de las interfaces de creación de pruebas y el diseño del mecanismo de generación de informes automáticos.
2. **Rendimiento satisfactorio en períodos de alta demanda:** Se proyectan 3 grandes períodos de evaluación al año (que tendrán lugar los meses de abril, julio y octubre), con una demanda esperada en el primer año de hasta 20.000 evaluaciones cada mes. Para soportar una demanda como la indicada es necesario adoptar estrategias de optimización del software y escalamiento de la plataforma desde la concepción del proyecto, con lo que aspectos como el caché de la aplicación, la reducción del peso y cantidad de las peticiones a los servidores y políticas de escalamiento de hardware según demanda, son fundamentales.
3. **Generación de informes automáticos:** Un informe corresponde a la presentación y análisis de los resultados obtenidos por un determinado conjunto de alumnos pertenecientes a un establecimiento. Cada informe tendrá 3 niveles de agregación: Alumno (por ejemplo: Juan González del 4to Básico A), Curso (por ejemplo: 4to Básico A) y Nivel educacional (por ejemplo: 4to Básico), y deberá poder ser visualizado tanto online -desde la aplicación-, como offline -mediante un archivo descargable. Estará compuesto principalmente de gráficos que indicarán el logro (porcentaje de preguntas contestadas correctamente) para un conjunto de atributos correspondientes a la evaluación en cuestión.
4. **Procesamiento de resultados para validación de evaluaciones:** Cada vez que finaliza un período de evaluación, junto con obtener resultados sobre el desempeño de los estudiantes, también se obtienen métricas sobre el desempeño de las evaluaciones. Dichas métricas permiten determinar la calidad de las preguntas utilizadas, indicando si fueron muy fáciles o muy difíciles, y si fueron correctamente planteadas. Con esta información se

posibilita la mejora continua de los instrumentos de evaluación, y la generación de conocimiento respecto a la construcción de ensayos.

1.3 Objetivos de la memoria

El objetivo general de este trabajo de memoria es diseñar y construir una aplicación web que permita el ingreso, aplicación y obtención de resultados de ensayos SIMCE y PSU para las disciplinas Lenguaje, Matemática Ciencias e Historia; en establecimientos educacionales asociados a la empresa Ediciones SM. Los objetivos específicos que se derivan del objetivo general son los siguientes:

1. Crear una aplicación web que permita la rendición de ensayos SIMCE y PSU a usuarios asociados a la editorial, logrando ampliar la actual cobertura de rendición.
2. Proporcionar un mecanismo de estadísticas e informes automáticos visibles desde el navegador web y descargables, tanto para profesores como para alumnos, que pueda equipararse al que se genera actualmente de forma manual.
3. Validación de los instrumentos de evaluación utilizando los datos generados por la aplicación.
4. Diseñar la aplicación considerando su uso progresivo y sostenido en el tiempo, posibilitando el buen desempeño y escalamiento de la misma.

2 Marco teórico

Para el desarrollo de la aplicación es necesario comprender cómo funcionan las evaluaciones, la editorial y los establecimientos educacionales; así como estudiar aspectos propios del desarrollo web. Estos temas son abordados en el presente capítulo.

2.1 Estructura de las evaluaciones

Una **evaluación** consiste en un conjunto de enunciados que tienen a su vez un conjunto de ítems. Cada ítem está formado por una pregunta, un número arbitrario de alternativas y una serie de atributos que dicen relación con la clasificación del contenido evaluado en la pregunta dentro de la malla curricular definida por el Ministerio de Educación [3]. Dicha malla curricular clasifica los contenidos según:

1. **Objetivos fundamentales (OF):** Son los aprendizajes que los alumnos y las alumnas deben lograr al finalizar los distintos niveles de la Educación Básica y Media. Se refieren a conocimientos, habilidades y conductas que han sido seleccionadas considerando que favorezcan el desarrollo integral de alumnos y alumnas.
2. **Contenidos mínimos obligatorios (CMO):** Los CMO explicitan los conocimientos, habilidades y conductas implicados en los OF y que el proceso de enseñanza debe convertir en oportunidades de aprendizaje para cada estudiante con el fin de lograr los Objetivos Fundamentales.
3. **Nivel educacional:** Se denomina niveles educacionales a los tramos cronológicos en que, tanto por razones técnicas como administrativas, ha sido dividido el proceso escolar que ocurre dentro de cada ciclo o subciclo de aprendizaje. Cada nivel corresponde a un año de estudio.
4. **Sector de aprendizaje:** El concepto sector de aprendizaje alude a las diversas categorías del saber y de experiencias que deben cultivar los niños, las niñas y jóvenes para desarrollar aquellas dimensiones de su personalidad consideradas relevantes por los fines, objetivos generales y requisitos de egreso de la enseñanza básica y media.

En consecuencia, para el proceso de evaluación, Ediciones SM ha definido el siguiente conjunto de atributos relevantes de clasificación para las preguntas en la aplicación:

1. **Disciplina:** Corresponde al sector de aprendizaje al que pertenece la pregunta (Lenguaje, Matemática, Ciencias o Historia).
2. **Temática:** Corresponde al OF que busca desarrollar la pregunta.
3. **Habilidad:** Corresponde a la habilidad necesaria para abordar satisfactoriamente la pregunta.
4. **Dificultad:** Es un indicador cualitativo que dice relación con la complejidad esperada de la pregunta.

5. **Nivel:** Corresponde al nivel educacional. Será perteneciente al intervalo 4to Básico - 4to Medio.

Los atributos Disciplina y Nivel son compartidos por todas las preguntas pertenecientes a una misma evaluación.

El acceso a las evaluaciones estará regulado por la editorial, y será función de los siguientes atributos:

1. **Colegio:** Establecimiento educacional dentro de la aplicación.
2. **Tipo de prueba:** SIMCE o PSU.
3. **Disciplina**
4. **Nivel**

Así cada vector $\langle \text{Colegio}, \text{Tipo de Prueba}, \text{Disciplina}, \text{Nivel} \rangle$ definirá un permiso dentro de la aplicación, por ejemplo: Liceo Lastarria, tendrá acceso a pruebas tipo PSU de la disciplina Matemática para el nivel educacional 4to Medio.

Un **período de evaluación** será un intervalo de tiempo no menor a un mes en que se habilitarán determinados ensayos en la aplicación para su calendarización y posterior aplicación por parte de los establecimientos educacionales.

Un **informe de resultados** corresponderá al conjunto de tablas y gráficos que informen a los profesores del establecimiento sobre el desempeño de los alumnos en los ensayos rendidos a través de la aplicación. Estos informes tendrán 3 niveles de agregación: Alumno (por ejemplo: Juan González del 4to Básico A), Curso (por ejemplo: 4to Básico A) y Nivel educacional (por ejemplo: 4to Básico).

Actualmente la **editorial** realiza jornadas de ensayos SIMCE y PSU de forma presencial en colegios asociados a la misma. La elaboración de estas evaluaciones es un proceso arduo de análisis, construcción, validación y actualización (debido principalmente al dinamismo de las mallas curriculares), a cargo de su área académica. Para la integración de este proceso con la aplicación, se contará con personal dedicado a la digitación de preguntas y construcción de pruebas.

Los **colegios** están cada vez más abiertos a la adopción de herramientas tecnológicas para el apoyo educativo, sin embargo, aún precisan del control sobre las metodologías a aplicar. En este sentido la aplicación busca ser de utilidad a través del profesor, entregando las funcionalidades necesarias para que él decida a quiénes, qué y cuándo evaluar. Las evaluaciones se llevarán a cabo en las salas de computación de los colegios, las que en algunos casos no tienen la capacidad para albergar simultáneamente a todos los alumnos de un curso, por lo que la aplicación permite la evaluación de un mismo curso en distintos horarios, siendo así posible abarcar a todos los alumnos.

2.2 Análisis de preguntas

Con el fin de mejorar continuamente la calidad de las preguntas, se utilizarán los resultados disponibles en la aplicación para medir su desempeño, calculando los siguientes indicadores [4]:

1. **Índice de dificultad:** Se utiliza para cuantificar el grado de dificultad de una pregunta y se calcula como sigue:

$$D_i = \frac{A_i}{N}, \text{ donde}$$

D_i = Índice de dificultad para la pregunta i
 A_i = Cantidad de aciertos en la pregunta i
 N = Total de estudiantes en la muestra

2. **Índice de homogeneidad:** Corresponde a la correlación de Pearson entre las puntuaciones de una pregunta y los puntajes totales obtenidos por los estudiantes en la evaluación, siendo obtenido según la siguiente fórmula:

$$H_i = \frac{\sum_n P_{in} P_{tn} - N \bar{P}_i \bar{P}_t}{N \sigma_{P_i} \sigma_{P_t}}, \text{ donde}$$

H_i = Índice de homogeneidad para la pregunta i
 P_{in} = Puntaje obtenido por el estudiante n en la pregunta i
 P_{tn} = Puntaje total obtenido por el estudiante n
 N = Total de estudiantes en la muestra
 \bar{P}_i = Puntaje promedio obtenido en la pregunta i
 \bar{P}_t = Puntaje total promedio obtenido en la evaluación
 σ_{P_i} = Desviación estándar en la pregunta i
 σ_{P_t} = Desviación estándar total de la evaluación

3. **Índice de homogeneidad corregido o Índice de discriminación:** Corresponde a la correlación de Pearson entre las puntuaciones de una pregunta y los puntajes totales obtenidos por los estudiantes en la evaluación descontando dicha pregunta, y se calcula según se muestra a continuación:

$$HC_i = \frac{\sum_n P_{in}(P_{tn}-P_{in}) - \bar{P}_i \sum_n (P_{tn}-P_{in})}{N \sigma_{P_i} \sqrt{\frac{1}{N} \sum_n (P_{tn}-P_{in})^2}}, \text{ donde}$$

HC_i = Índice de homogeneidad corregido para la pregunta i y el resto de los parámetros son los mismos de la fórmula anterior.

La interpretación de los índices definidos se muestra a continuación (Tabla 1 y Tabla 2).

Tabla 1: Interpretación índice de dificultad

Valor índice dificultad	Interpretación respecto a la pregunta
$D < 0,25$	Muy difícil
$0,25 < D < 0,45$	Difícil
$0,45 < D < 0,55$	Normal
$0,55 < D < 0,75$	Fácil
$D > 0,75$	Muy fácil

Tabla 2: Interpretación índice de homogeneidad

Valor índice homogeneidad	Interpretación respecto a la pregunta
$H > 0,39$	Conservar
$0,30 < H < 0,39$	Buena mejorar
$0,20 < H < 0,29$	Regular revisar
$0,00 < H < 0,20$	Pobre descartar
$H < -0,01$	Pésima descartar

2.3 Proceso tradicional de toma de evaluaciones

Actualmente la toma de evaluaciones presencial se realiza en 3 etapas:

1. **Construcción de material:** Consiste en la recopilación y creación de preguntas para la generación de las distintas evaluaciones a aplicar en los establecimientos educacionales seleccionados por la editorial.
2. **Aplicación presencial:** Consiste en el desplazamiento de personal académico de la editorial a los diversos colegios, con el material generado, para realizar la aplicación presencial de la evaluación. El evaluador está presente hasta el final de la evaluación, retirando posteriormente el material y llevándolo de vuelta a la editorial. Considerando que para cada periodo de evaluación se disponen de 2 evaluadores dedicados, los que máximo aplican 2 evaluaciones diarias, se tiene que la capacidad de evaluación diaria es de 4 cursos.
3. **Digitalización de resultados y generación de informes:** Una vez se tienen una cantidad importante de resultados, se procede al procesamiento de las hojas de respuesta, su traspaso a Excel, la generación manual de gráficos y finalmente la construcción del informe, el que es enviado en forma impresa a los establecimientos entre 4 a 6 semanas después de rendida la evaluación. También, mediante el software SPSS se realiza el estudio de los ítems, calculando indicadores como Índice de dificultad e Índice de homogeneidad, los que deben ser recalculados cada vez que se tienen más resultados.

Cabe mencionar que uno de los principales valores de la aplicación respecto al proceso tradicional antes descrito es la eliminación del trabajo asociado a las etapas 2 y 3.

2.4 Arquitectura de una aplicación web

Conforme a las buenas prácticas de desarrollo web, el patrón más utilizado para la construcción de aplicaciones es el Modelo Vista Controlador (MVC), que permite estructurar la aplicación en 3 capas:

- El **modelo** que representa los datos de la aplicación
- La **vista** que despliega la presentación de los datos del modelo
- El **controlador** maneja las peticiones realizadas por el cliente

La separación que presenta la arquitectura MVC nos permite distribuir el trabajo entre el diseño gráfico y el desarrollo de lógica de la aplicación, facilitando la reutilización y mantención del código. Esta división en capas sitúa a dos de ellas por el lado del servidor, modelo y el controlador, mientras que la vista se encuentra asociada al cliente, correspondiente al navegador web. De esta forma, para desarrollar del lado del servidor es necesario conocer lenguajes tales como JAVA, Ruby o PHP, mientras que para el desarrollo de las vistas es preciso manejar HTML, CSS y JavaScript.

Considerando la experiencia del alumno con el lenguaje PHP y las ventajas ya mencionadas que presenta la utilización del patrón MVC, para el desarrollo de la aplicación se utilizará el framework MVC PHP de código abierto CakePHP [5]. Un framework es una estructura conceptual y tecnológica sobre la que un proyecto de software puede ser más fácilmente organizado e implementado. Entre sus características destacan:

- Arquitectura MVC
- Patrón ActiveRecord para la capa modelo
- Limpieza y validación de datos ingresados desde formularios
- Componentes para el manejo de correos, cookies, seguridad, sesiones y peticiones.

A continuación se presenta un diagrama del ciclo de una petición en CakePHP (Figura 1).

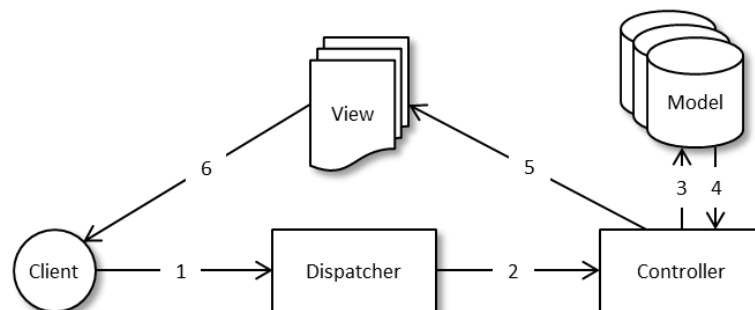


Figura 1: Ciclo de una petición en CakePHP

2.4.1 Modelo

Un modelo representa datos generalmente ligados a una tabla en una base de datos, sin embargo puede ser utilizado para acceder a cualquier estructura que pueda proporcionar datos, como archivos, registros LDAP o Web Services. Un modelo puede estar asociado con otros modelos, por ejemplo, el modelo *Libro* (ligado a la tabla *libros*) puede estar asociado con el modelo *Autor* (ligado a la tabla *autores*), y proveen toda la funcionalidad para realizar consultas de forma rápida utilizando métodos de objetos, en lugar de sentencias SQL.

En CakePHP también existe el concepto de **behavior**, que corresponde a una estructura que encapsula lógica que será transversal a varios modelos. Por ejemplo, si se desea guardar información sobre cada cambio que tengan los datos correspondientes a ciertos modelos, basta crear un behavior *Logging* que encapsule dicha funcionalidad, declarando posteriormente una variable de clase en los modelos para indicarles que “actúen” según el comportamiento *Logging*.

Adicionalmente el framework incluye de forma predeterminada behaviors para manejar listas de control de acceso, restricciones en los resultados por defecto de las consultar a base de datos y traducciones, entre otros.

2.4.2 Vista

Las vistas son responsables de generar las salidas requeridas para una petición HTTP. Generalmente retornan HTML, XML o JSON, aunque también pueden utilizarse para definir archivos Excel o PDF. Las vistas son escritas en PHP, pero con la extensión de archivo *ctp*. Estos archivos contienen la lógica necesaria para manipular los datos recibidos desde el controlador, de tal forma que sean desplegados en el formato requerido.

Esta capa está formada por diferentes partes, que en conjunto constituyen la respuesta que recibirá un navegador web tras una petición:

1. **Layouts:** Archivos de vista que contienen código de presentación que envuelven a muchas vistas en la aplicación. La mayoría de las vistas son renderizadas dentro de un layout.
2. **Views:** Archivos de vista que contienen código de presentación relacionado a una tarea particular.
3. **Elements:** Trozo de vista más pequeño y reutilizable. Son utilizados generalmente dentro de vistas.
4. **Helpers:** Clases que proveen funcionalidades útiles para varias vistas.

2.4.3 Controlador

Un controlador es utilizado para manejar las peticiones del cliente. Es el encargado de –mediante los modelos– obtener los datos necesarios, transformarlos y entregarlos a las vistas que finalmente son despachadas al cliente. Los controladores son clases que extienden de la clase *AppController*, que extiende a su vez de la clase *Controller* proporcionada por CakePHP, éstos pueden incluir cualquier número de funciones, llamadas *actions*, las que corresponden a las diversas rutas

accesibles por el cliente para llevar a cabo diversas tareas. Por ejemplo, la ruta */recetas/agregar* coincidiría con la función (action) agregar de la clase (controlador) RecetaController.

Análogo a los behavior y elements, en esta capa existen los **Components**, que son los encargados de encapsular lógica transversal a varios controladores.

3 Diseño general de la aplicación

En esta sección se describen los requisitos, arquitectura y herramientas de software a utilizar.

3.1 Requisitos de la solución

Los principales requisitos a considerar por la solución son los siguientes:

1. **Creación de pruebas:** La aplicación debe permitir a los evaluadores de la editorial el ingreso de los ensayos. Las preguntas deben almacenarse de forma tal que se permita su reutilización en otras pruebas, dando origen a un banco de preguntas. El número de alternativas por pregunta debe ser variable pero siempre mayor que 2 y menor que 7.
2. **Definición de período de evaluación:** Cada prueba ingresada en la aplicación será un objeto reutilizable, que será instanciado en un período de evaluación. La aplicación debe permitir a un usuario privilegiado la definición de un período de evaluación, el que estará determinado por el vector $\langle Nivel, Prueba, Fecha Inicio, Fecha Fin \rangle$, por ejemplo: Nivel educacional 4to Medio podrá rendir la prueba PSU Matemática entre 2012-08-01 y 2012-08-31.
3. **Importación de alumnos:** La aplicación debe proveer a los profesores un mecanismo sencillo y rápido de importación de alumnos. Dicha importación debe tener como consecuencia la creación de credenciales de usuario para cada alumno, la asociación de los alumnos a cursos y a su vez la asociación de los cursos al colegio correspondiente. Una vez realizado este proceso el profesor estará en condiciones de calendarizar evaluaciones y los alumnos podrán ingresar a la aplicación.
4. **Asignación de permisos de acceso a evaluaciones:** La aplicación pondrá a disposición de los colegios evaluaciones de distinto tipo (SIMCE y PSU) en 4 disciplinas (Lenguaje, Matemática, Ciencias e Historia) y para diversos niveles educacionales (4to Básico a 4to Medio). Dado que no necesariamente los colegios tendrán acceso a todas las evaluaciones, la aplicación debe permitir a un usuario privilegiado la definición de permisos de acceso a dichas evaluaciones. Cada permiso estará definido por el vector $\langle Colegio, Tipo de Prueba, Disciplina, Nivel \rangle$, por ejemplo: Liceo Lastarria, tendrá acceso a pruebas tipo PSU de la disciplina Matemática para el nivel educacional 4to Medio.
5. **Calendarización de pruebas:** Si existe un período de evaluación vigente, se ha realizado la importación de los alumnos y el colegio presenta los permisos de acceso adecuados, la aplicación debe proveer a los profesores un mecanismo para seleccionar el momento de aplicación de las evaluaciones disponibles. Esta calendarización deberá considerar aspectos de rendimiento, evitando la sobre exigencia de la aplicación al distribuir de forma homogénea la carga.
6. **Rendición de prueba:** La aplicación debe proveer a los alumnos (niños y jóvenes entre 9 y 17 años) un mecanismo de resolución de pruebas a través de una interfaz simple y limpia, libre de distracciones, que les permita enfocarse en lo más importante: la resolución de los ensayos. El acceso a dicha interfaz debe ser lo más expedito posible al momento en

que el alumno se identifique en la aplicación, evitando entorpecer la labor de guía de la autoridad docente.

7. **Visualización de resultados e informes:** Los distintos perfiles relacionados a una evaluación deben obtener estadísticas sobre los resultados de la misma, con perspectivas específicas de cada perfil. Así los alumnos tendrán acceso a revisar sus propios resultados, los profesores a resultados de todos los alumnos y estadísticas comparativas entre cursos, y finalmente los evaluadores de la editorial a estadísticas entre colegios. Para los profesores, estos resultados deben ser presentados, además, en forma de un informe descargable, el que debe acercarse lo más posible al actualmente entregado por la editorial cuando realiza el proceso tradicional de evaluación. A continuación se muestran imágenes referenciales (Figura 2 y Figura 3) de dicho informe (la identidad del colegio se mantiene en reserva).

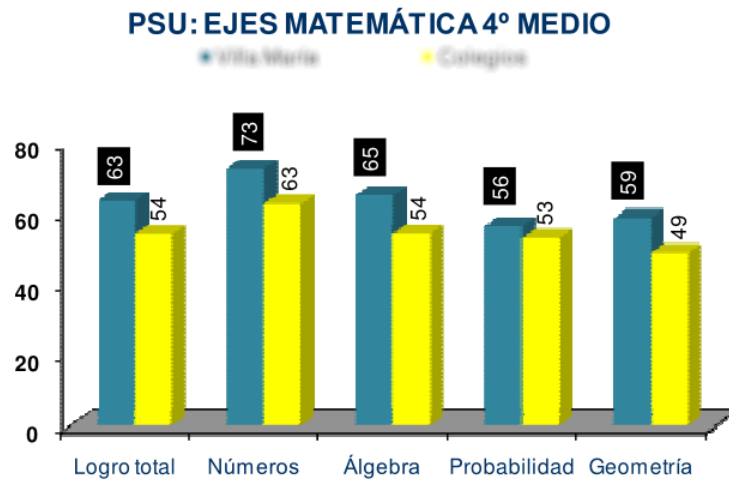


Figura 2: Gráfico “Logro por Eje Temático”

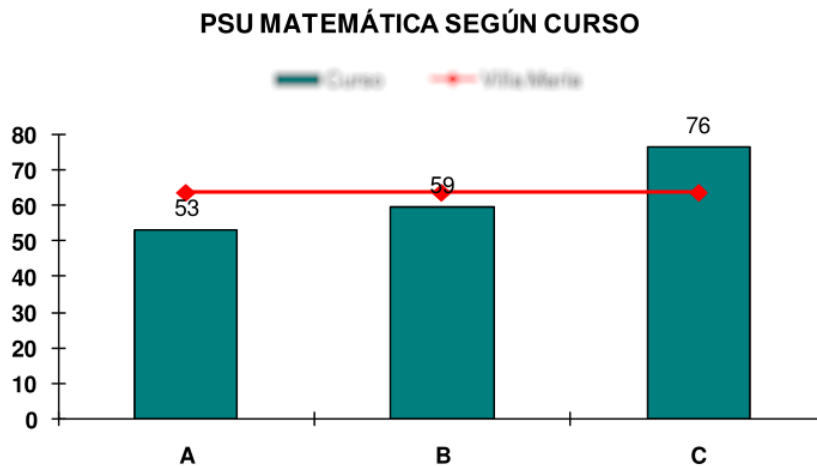


Figura 3: Gráfico "Logro por Curso"

8. **Alertas y notificaciones:** La aplicación deberá generar alertas (entendidas como mensajes a los usuarios mediante un canal externo a la aplicación) y notificaciones (mensajes a los usuarios dentro de la aplicación) ante diversos eventos, como el comienzo de un período de evaluación, la calendarización de una prueba o la proximidad a una fecha de aplicación.

3.2 Arquitectura física de la solución

La plataforma de hardware estará contenida en la nube de Amazon [6], haciendo uso de los siguientes recursos:

- **EC2 (Elastic Cloud Computing) Instance (de 1 – 3 instancias):** Máquina virtual que contendrá al servidor web. Esta instancia podrá ser replicada según demanda.
- **RDS (Relational Database Service) Instance (1 instancia):** Máquina virtual íntegramente dedicada al servicio de base de datos.
- **Load Balancer Instance (1 balanceador):** Instancia especial para balanceo de carga, con capacidad de balanceo de hasta 10 máquinas.
- **Route53:** Servicio DNS.
- **VPC (Virtual Private Cloud) (1 red):** Red privada dentro de la red Amazon.
- **Elastic IP (1 IP):** La IP pública que será asignada al balanceador de carga. Podrá ser direccionada a otra máquina en caso de ser necesario, lo que permitirá cambios “en caliente”, sin necesitar esperar refresco de DNS.

A continuación se presenta el esquema de arquitectura física de la aplicación (Figura 4).

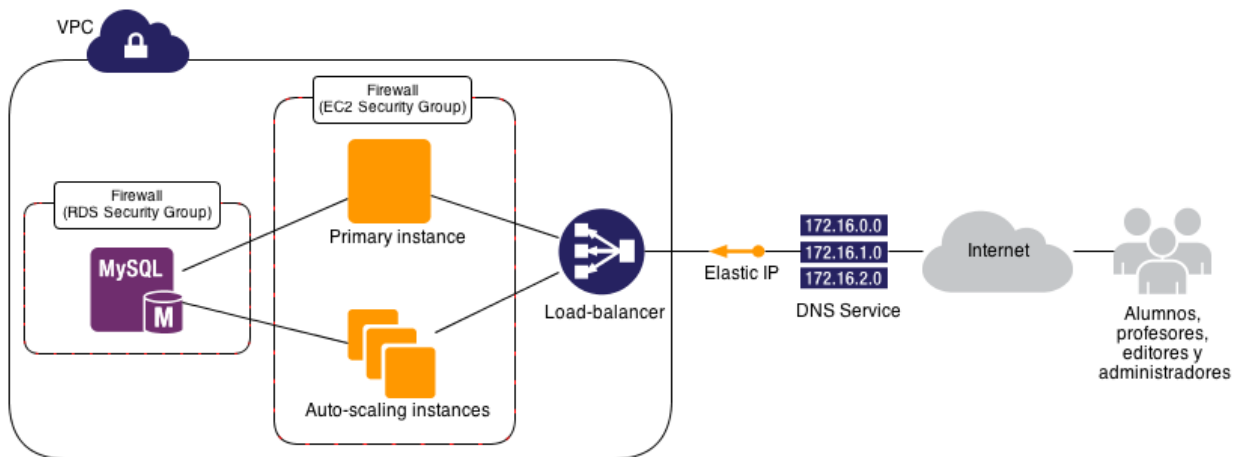


Figura 4: Arquitectura física de la aplicación

3.3 Arquitectura lógica de la solución

La aplicación se presenta organizada en componentes y módulos. Los componentes de aplicación agrupan funcionalidades que darán soporte a los módulos, pudiendo estos últimos ser entendidos como un conjunto de funcionalidades dirigidas a un perfil de usuario específico. Considerando lo anterior, la anatomía de la aplicación responde a la siguiente estructura (Figura 5).

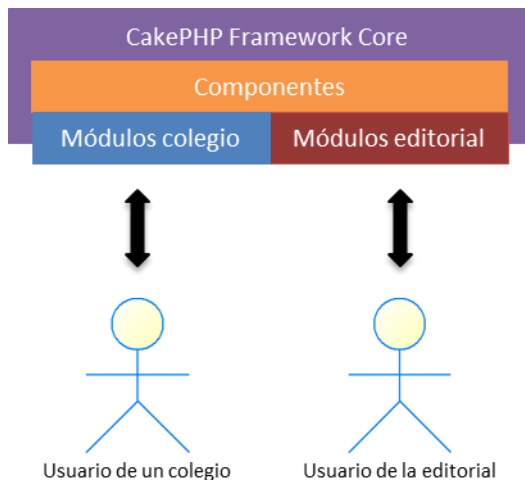


Figura 5: Anatomía de la aplicación

Como se ve en la figura anterior, los usuarios provienen de 2 fuentes: la editorial y los colegios. Los perfiles correspondientes a usuarios de la editorial son *Administrador* y *Editor*, mientras que los relativos a usuarios de los colegios son *Profesor* y *Alumno*, según resume el siguiente diagrama (Figura 6).

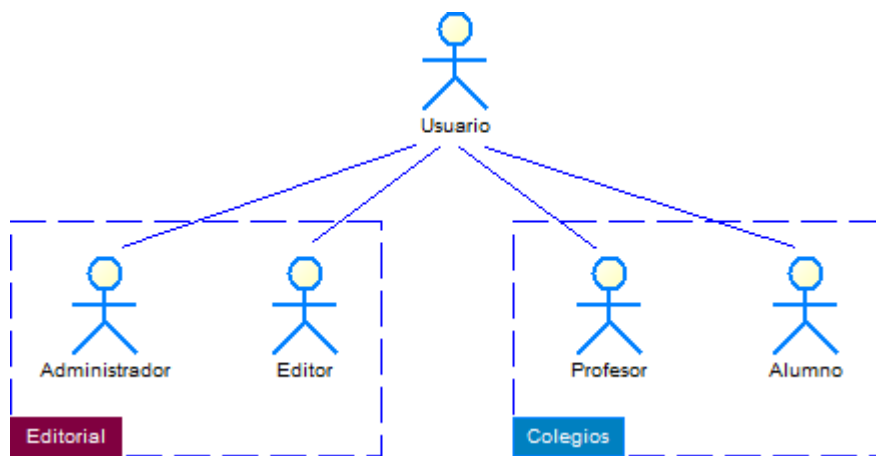


Figura 6: Usuarios de la aplicación

Cabe mencionar que durante etapas tempranas del desarrollo se consideró un perfil de usuario adicional para uso de la editorial, el *Revisor*, quien sería el encargado de dar el visto bueno a cada una de las preguntas presentes en una prueba. Sin embargo, se determinó que dado el flujo de trabajo de la institución, donde *Editor* y *Revisor* en realidad correspondían a la misma persona, no aplicaba tener dos perfiles de usuario distintos.

Los componentes de aplicación y sus principales funcionalidades se indican a continuación (Figura 7).

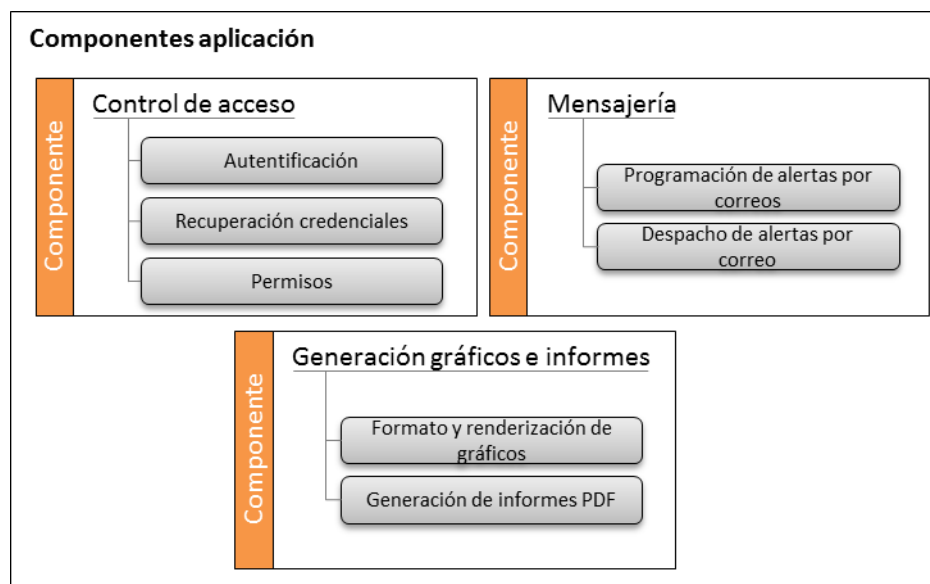


Figura 7: Componentes de la aplicación

Estos componentes brindan funcionalidades comunes a los módulos de aplicación, detallados en el siguiente esquema (Figura 8).

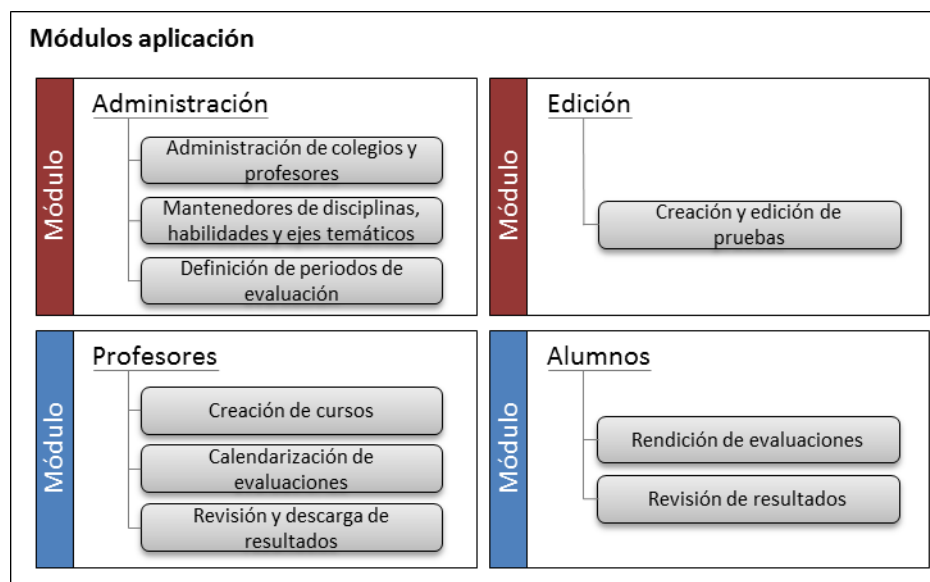


Figura 8: Módulos de la aplicación

Como se puede ver en la figura anterior, el módulo edición está destinado íntegramente a la creación y edición de pruebas, siendo accesible sólo por personal de la editorial con perfil *Editor*. La decisión de aislar esta funcionalidad y no incluirla en el módulo administración fue la separación de responsabilidades, ya que este último maneja características sensibles de la aplicación, como la habilitación de profesores y la activación y asignación de permisos para rendir evaluaciones a los colegios en la aplicación.

El modelo de datos de la aplicación comprende tablas para las principales entidades: *Usuarios*, *Cursos*, *Colegios*, *Preguntas*, *Pruebas* y *Resultados*, así como también tablas auxiliares para cálculo de estadísticas y tablas de tipos. A continuación se presenta el modelo de datos simplificado (Figura 9), compuesto por las tablas y asociaciones más relevantes de la aplicación.

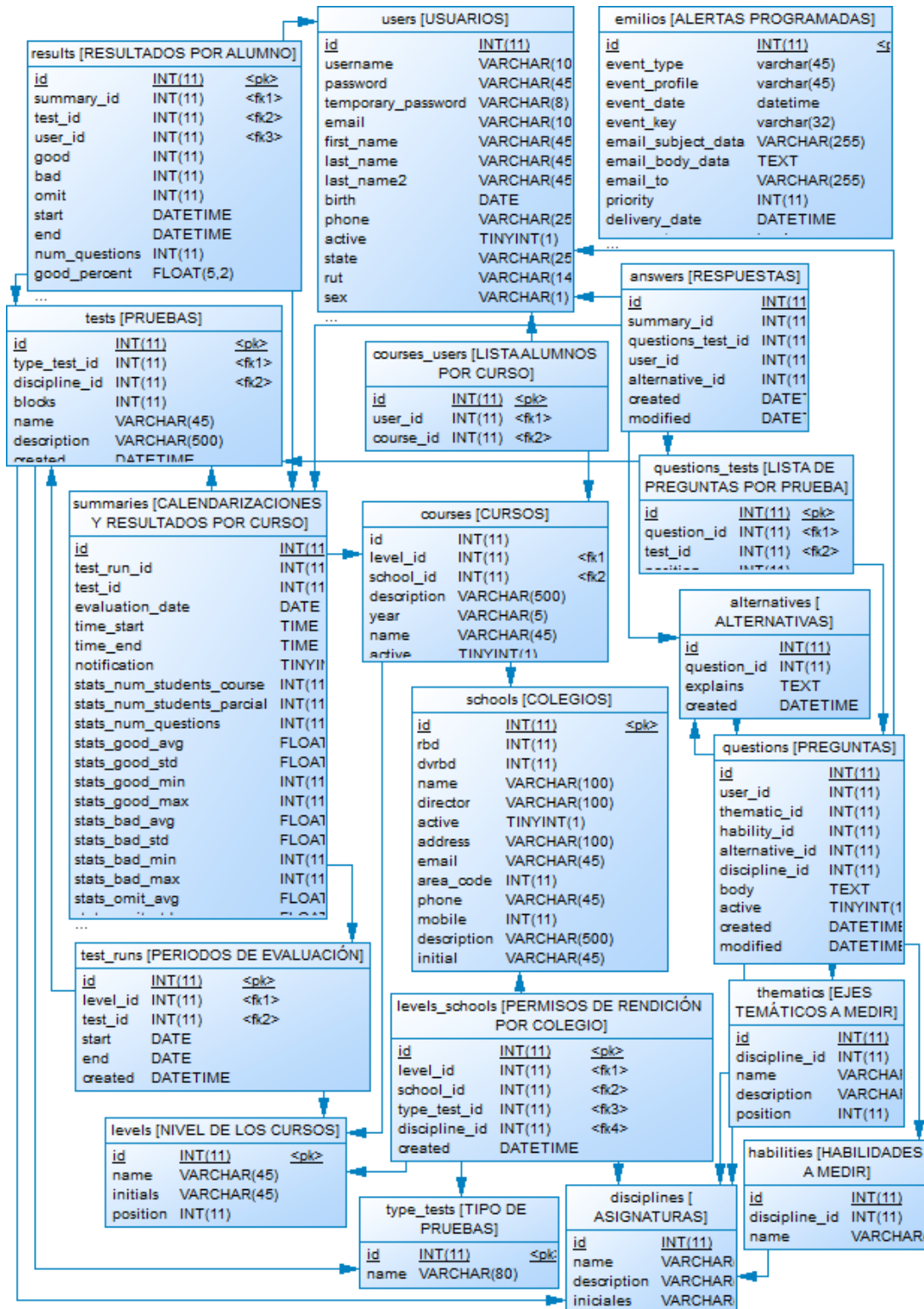


Figura 9: Modelo de datos simplificado

El modelo de datos fue construido siguiendo las convenciones de CakePHP, es decir:

- Nombres de tablas en inglés, plural y guiones bajos para separar palabras (por ejemplo: *schools*).
- Las llaves foráneas deben tener el nombre de la tabla referenciada en singular, seguido de *_id* (por ejemplo: *school_id*).
- Las tablas que indican una relación N:N entre otras dos tablas, deben llevar por nombre el de las tablas involucradas, separadas por un guión bajo, en orden alfabético (por ejemplo: *questions_tests* en vez de *tests_questions*).
- Todas las tablas necesitan un identificador primario llamado *id*, con excepción de aquellas descritas en el punto anterior.

3.4 Herramientas de desarrollo

Tras analizar las diversas alternativas, por motivos de agilidad de desarrollo y experiencia se optó por la utilización de las siguientes tecnologías:

- Como plataforma de hardware se optará por infraestructura en la nube de Amazon, explicado en detalle en la sección 3.2 Arquitectura de la Solución.
- El sistema operativo utilizado por el servidor será **Ubuntu Server 12.04 LTS**.
- Para el desarrollo por el lado del servidor se utilizará el lenguaje de programación **PHP**, haciendo uso del framework **CakePHP**, que sigue el patrón MVC.
- Para el manejo de datos se utilizará el gestor de base de datos **MySQL**.
- Para la interactividad de las interfaces, por el lado del cliente se hará uso del lenguaje de programación **JavaScript** con su librería **jQuery** [7].
- Para la construcción de las vistas se utilizará el framework CSS + JavaScript **TwitterBootstrap**.
- Para el despliegue de la aplicación en los distintos ambientes se utilizará a través del lenguaje **Ruby** la herramienta de ejecución de scripts en múltiples servidores **Capistrano**.
- Para el versionado del código se utilizará **Git**, utilizando además la metodología **Git-Flow**.

4 Diseño detallado de la aplicación

En esta sección se describen en detalle los componentes y módulos de la aplicación.

4.1 Componente control de acceso

Este componente es el encargado de manejar los permisos de usuario. Sigue una estructura de grupos y permisos, según puede desprenderse de las tablas que lo soportan mostradas a continuación (Figura 10).

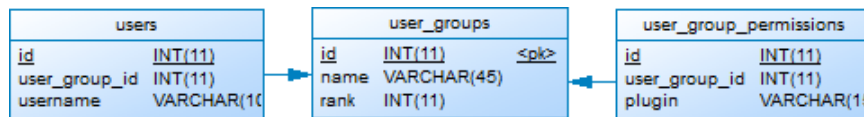


Figura 10: Modelo de datos simplificado, componente manejo de usuarios

Cada usuario de la aplicación pertenece a uno de los perfiles señalados en el capítulo anterior - *Administrador, Editor, Profesor y Alumno*- el que a su vez tiene acceso a un determinado conjunto de funcionalidades. El proceso control de acceso a funcionalidades se muestra a continuación (Figura 11).

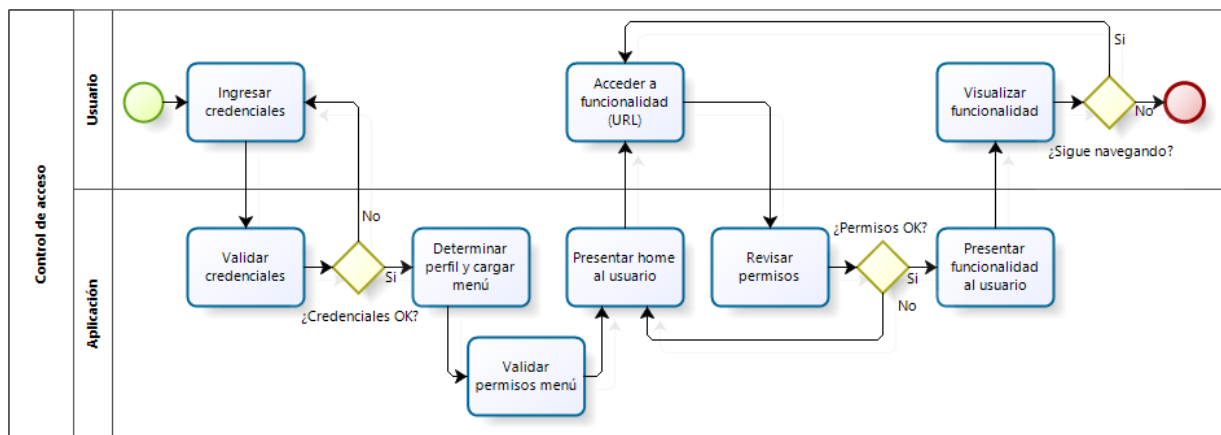


Figura 11: Proceso de control de acceso

Así, cuando un usuario ingrese con sus credenciales se cargará el menú asociado a su perfil y validará según los permisos establecidos, mientras que con cada acceso del usuario a una determinada funcionalidad, se volverá a realizar la validación.

4.2 Componente mensajería

La aplicación genera mensajes para distintos eventos que serán despachados vía correo. El listado de estos mensajes, los usuarios objetivo y el momento de despacho se indican en la siguiente tabla (Tabla 3).

Tabla 3: Mensajes enviados por la aplicación

Evento	Usuarios	¿Cuándo?
Nueva calendarización evaluación	Profesor, Alumno	En el momento que se crea
Modificación calendarización evaluación	Profesor, Alumno	En el momento que se modifica
Calendarización evaluación cancelada	Profesor, Alumno	En el momento que se cancela
Recordatorio calendarización evaluación	Profesor, Alumno	24 horas antes de la evaluación

Considerando que hay al menos un mensaje que debe ser programado para un momento futuro, y que eventualmente se agregarán mensajes ligados a otros eventos, este componente debe contener un proceso en segundo plano que revise un listado de mensajes pendientes y los despache según los criterios definidos. Dicho listado de mensajes será almacenado en la tabla *emilios*, mostrada a continuación (Figura 12).

emilios [ALERTAS PROGRAMADAS]		
id	INT(11)	PK
event_type	varchar(45)	
event_profile	varchar(45)	
event_date	datetime	
event_key	varchar(32)	
email_subject_data	VARCHAR(255)	
email_body_data	TEXT	
email_to	VARCHAR(255)	
priority	INT(11)	
delivery_date	DATETIME	

Figura 12: Sección modelo de datos, componente mensajería

En la tabla se puede observar el campo *email_to*, que contiene la dirección de correo donde se despachará el mensaje, es decir, para un evento que debe gatillar una alerta a múltiples usuarios, existirá un registro en la tabla *emilios* por cada uno de ellos.

4.3 Componente generación gráficos e informes

Cada vez que un alumno finalice una evaluación, se deben calcular estadísticas con distintos niveles de agregación, que como se verá más adelante serán: *Alumno*, *Curso* y *Nivel educacional*. Estas estadísticas serán desplegadas en forma de gráficos, que deberán poder ser desplegados online y además visualizables en un archivo PDF descargable. Con el fin de evitar la repetición de lógica, los archivos PDF serán generados a partir del mismo HTML que se visualiza online. Así, este componente se puede entender según muestra el siguiente diagrama (Figura 13).

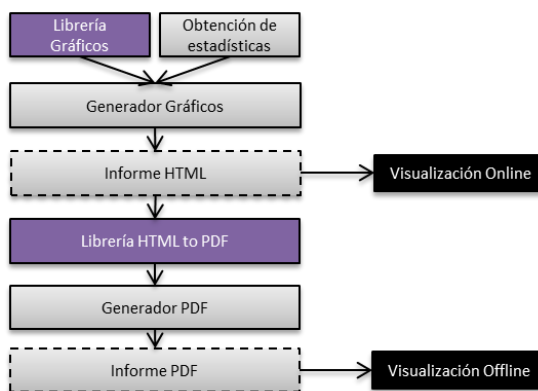


Figura 13: Componente generación gráficos e informes

4.4 Módulo administración

Este módulo provee a los administradores las funcionalidades para la administración de los colegios y profesores, mantenedores de disciplinas, habilidades y ejes temáticos, y definición de períodos de evaluación. A continuación se presenta una simplificación del modelo de datos con las tablas relevantes para este módulo (Figura 14).

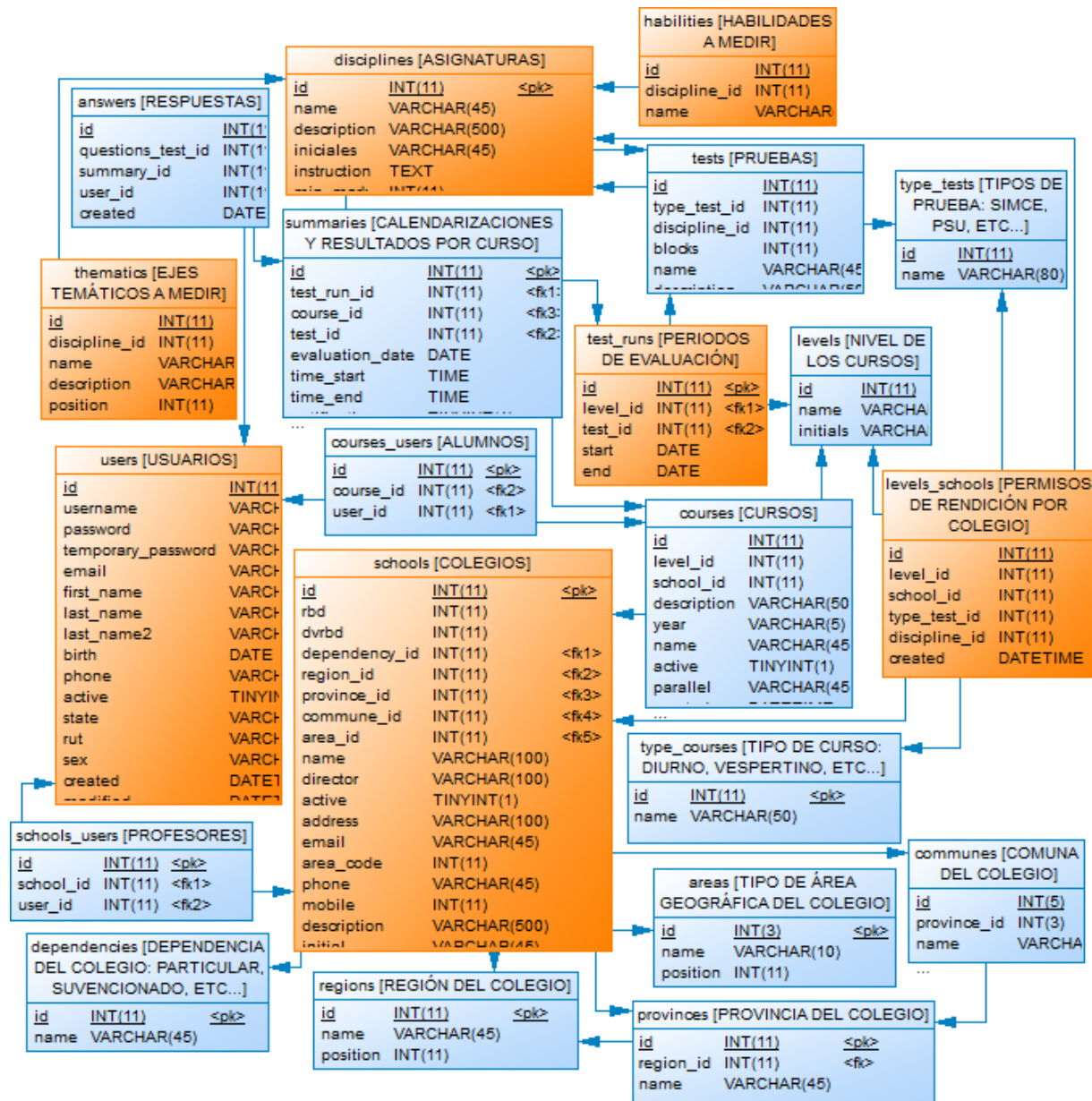


Figura 14: Modelo de datos simplificado, módulo administración

4.4.1 Administración de colegios y profesores

Aquellos colegios existentes al momento de realizar la presente memoria en el listado que publica el Ministerio de Educación, fueron previamente importados a la aplicación; sin perjuicio de lo anterior, nuevos colegios pueden ser registrados. Algunos de los datos que el ministerio provee sobre los colegios, y que son almacenados por la aplicación son: Nombre Establecimiento, RBD

(Rol Base de Datos MINEDUC), Ubicación y Dependencia (Particular, Subvencionado, etc.), entre otros. En un principio se consideró que serían profesores asociados a cursos específicos dentro de los establecimientos educacionales los usuarios responsables de calendarizar pruebas y revisar resultados, sin embargo, en el transcurso del trabajo se desestimó esta restricción, ya que la realidad en los colegios era que uno o dos funcionarios (en su mayoría profesores, pero también bibliotecarios y otros) serían los encargados de utilizar la aplicación de manera transversal a todos los cursos. De hecho, es por lo anterior que el perfil en realidad no se llama simplemente *Profesor*, sino que *Coordinador de Evaluaciones*, pero por simpleza en el presente trabajo es denominado profesor. Dado lo anterior, para que un usuario de la aplicación sea un profesor, éste tiene que estar registrado en la tabla *users* con perfil *Profesor*, y tener una asociación con la tabla *schools* en *schools_users*. La asociación no es mediante una llave foránea en *users* ya que se prevé que en un futuro un usuario profesor podría estar asociado a más de un colegio (especialmente en el caso de las Corporaciones Educacionales).

El acceso a las evaluaciones para los colegios estará asociado a utilización de los productos de la editorial. Así, establecimientos usuarios de material educativo de la editorial para la disciplina Matemática, nivel educacional 4to Básico, tendrán acceso a las evaluaciones SIMCE Matemática habilitadas para dicho nivel. Los permisos de rendición por colegio se encuentran definidos en la tabla *levels_schools*, mediante los campos *school_id*, *type_test_id*, *discipline_id* y *level_id*.

En este módulo además se puede visualizar la información de los cursos (colegio, nivel educacional, nómina de alumnos), mientras que su creación recae en el Módulo Profesores.

4.4.2 Mantenedores de disciplinas, habilidades y ejes temáticos

Los resultados de las evaluaciones son presentados categorizados por disciplina, sexo, habilidad y eje temático, siendo disciplina una propiedad de la prueba (campo *discipline_id* en tabla *tests*), sexo una propiedad del alumno (campo *sex* en tabla *users*), y habilidad y eje temático propiedades de las preguntas (campos *ability_id* y *thematic_id* en tabla *questions*). En este módulo se pueden agregar cada uno de estos parámetros para que estén disponibles al momento de crear preguntas y consecuentemente en la visualización de resultados.

4.4.3 Definición de períodos de evaluación

La última funcionalidad correspondiente a este módulo es la definición de períodos de evaluación. La editorial determina jornadas de evaluación masivas, generalmente 3 veces al año, con una duración de 1 mes. Así, el módulo permite la definición de dichos períodos asociados a una prueba en particular, lo que en la aplicación se denomina *Lanzamientos de Pruebas*, y que son registrados en la tabla *test_runs*. Esta tabla finalmente define una instancia de prueba, pudiendo una misma prueba ser evaluada en múltiples períodos. Se puede, además, ver que la tabla tiene el campo *level_id*, lo que indica que una prueba puede ser aplicada a distintos niveles educacionales (por ejemplo una prueba para un 4to Básico terminal también puede emplearse para un 5to Básico inicial).

4.5 Módulo edición

Este módulo provee a los editores las funcionalidades para la creación y edición de pruebas. A continuación se presenta una simplificación del modelo de datos con las tablas relevantes para este módulo (Figura 15).

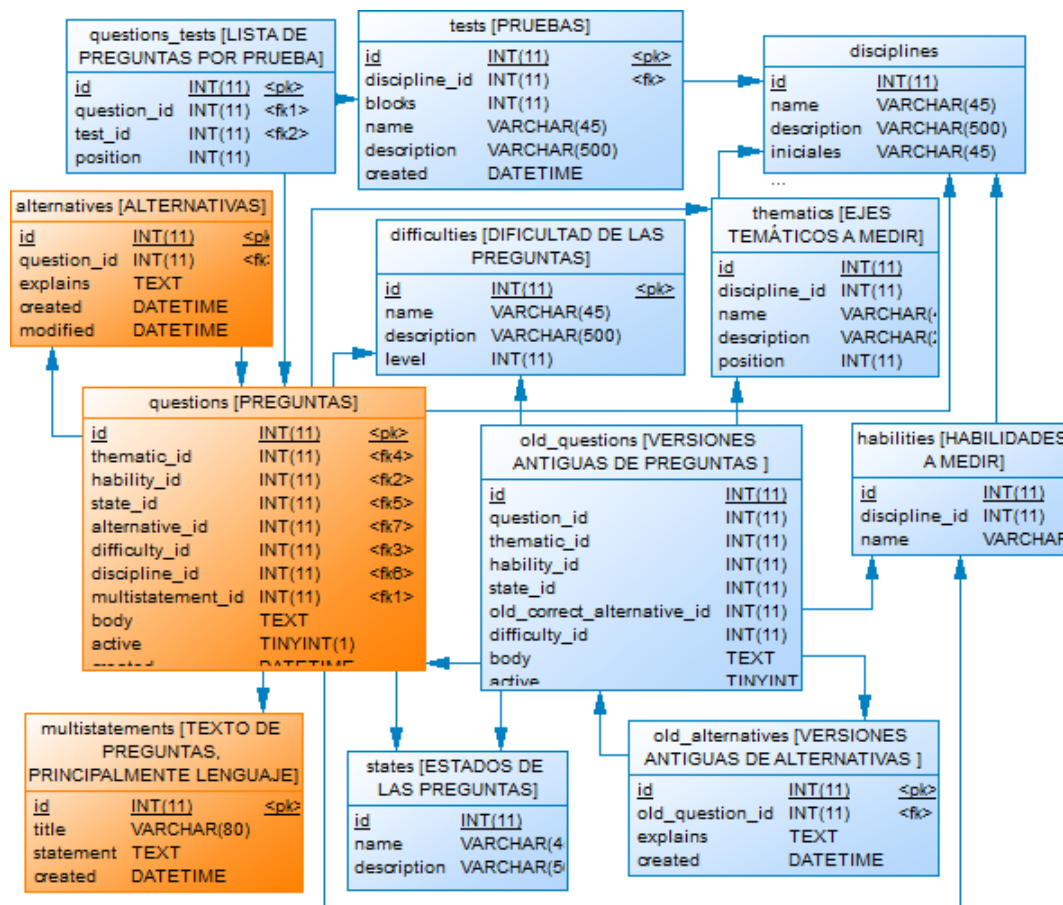


Figura 15: Modelo de datos simplificado, módulo edición

Una prueba está compuesta por preguntas (tablas *tests* y *questions*), las que a su vez tienen un conjunto de alternativas asociadas (tabla *alternatives*). Además, una o más preguntas pueden estar ligadas a un mismo texto (tabla *multistatement*), como es el caso de las preguntas de comprensión de lectura, en la disciplina de lenguaje. Con el fin de obtener resultados de rendimiento más detallados para profesores y alumnos, cada pregunta mide parámetros de habilidad y eje temático (tablas *abilities* y *thematics*), además de tener una dificultad (tabla *difficulties*) asociada, que sirve de referencia para el editor. En caso de que en un futuro se necesite acceder al historial de edición de una pregunta, se consideraron las tablas *old_questions* y *old_alternatives*. La tabla *questions_tests* define la asociación entre prueba y preguntas, de lo que se desprende que una pregunta puede pertenecer a más de una prueba, consiguiendo que sean reutilizables. Dicha tabla es la encargada de mantener la posición de la pregunta dentro de la prueba, mediante el campo *position*.

4.6 Módulo profesores

Este módulo provee a los profesores las funcionalidades para la creación de cursos, calendarización de evaluaciones y revisión y descarga de resultados. A continuación se presenta una simplificación del modelo de datos con las tablas relevantes para este módulo (Figura 16).

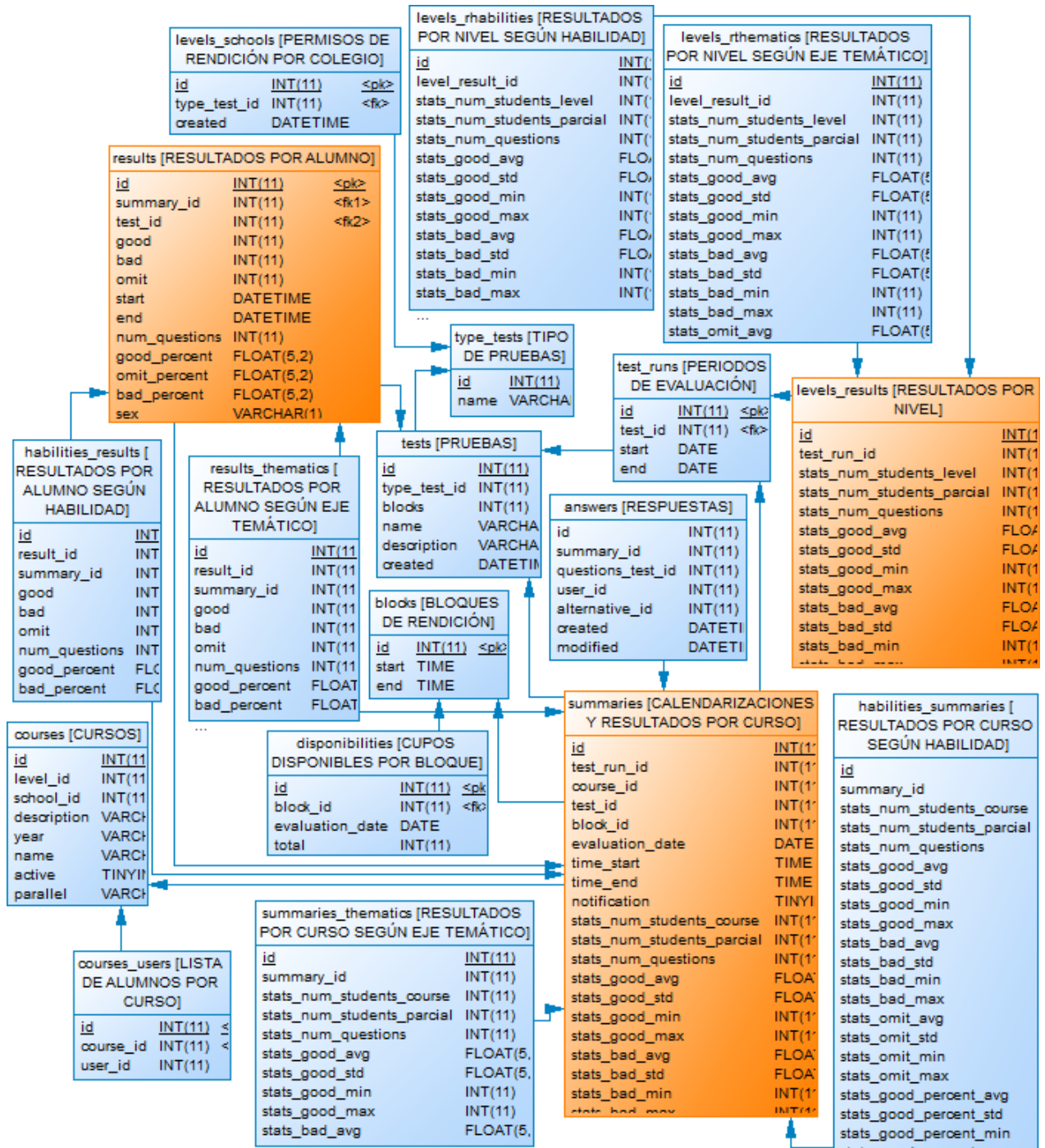


Figura 16: Modelo de datos simplificado, módulo profesores

4.6.1 Creación de cursos

La creación de un curso es el primer paso para que los alumnos puedan rendir una prueba. Con el fin de minimizar la posibilidad de error del profesor, y facilitar el traspaso de los datos se determinó que la aplicación recibirá un archivo Excel con el listado de alumnos de un curso según el siguiente formato (Figura 17).

	A	B	C	D	E	F	G
1	Los puntos son opcionales, el guión es obligatorio	Puede ser en mayúsculas o minúsculas	Puede ser en mayúsculas o minúsculas	Puede ser en mayúsculas o minúsculas	Ingresar SÓLO "F" para Femenino o "M" para Masculino	Ingresar SÓLO desde 1B a 8B (para enseñanza básica) o desde 1M a 4M (para enseñanza media)	Agregar SÓLO una letra
2	RUT	NOMBRES	APELLIDO PATERNO	APELLIDO MATERNO	SEXO	NIVEL	PARALELO
3							
4							

Figura 17: Formato archivo de importación de alumnos en Excel

Dicho archivo se procesará como muestra a continuación la Figura 18.

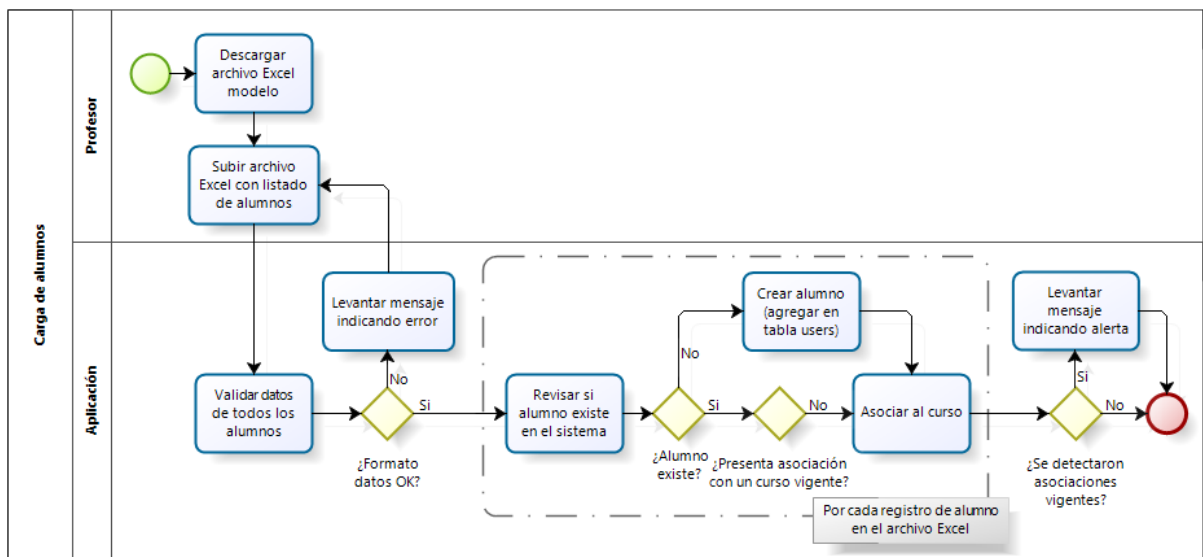


Figura 18: Proceso de carga de alumnos

El paso de decisión “¿Presenta asociación con un curso vigente?” valida que en caso de existir un curso (tabla *courses*) del presente año, exista ya un alumno (tabla *users*) de *rut* igual al indicado en el archivo Excel, tal que presenta una asociación mediante la tabla *courses_users* con dicho curso. Si el alumno ya está asociado a un curso se le informará al usuario mediante una alerta. En caso de que no existan ni el curso ni el alumno, se crearán los registros correspondientes en las tablas *courses*, *users* y *courses_users*.

4.6.2 Calendarización de evaluaciones

Si el colegio presenta permisos de rendición para los cursos ingresados en la aplicación, y hay un período de evaluación vigente para ellos, el profesor tendrá la posibilidad de calendarizar

evaluaciones. Con el fin de tener cierto control sobre la carga de la aplicación, se definirá un parámetro que indicará cuántos alumnos pueden estar rindiendo una prueba al mismo tiempo, llamado *cupo*. Cada vez que se calendarice (tabla *summaries*) una evaluación, se descontará la cantidad de alumnos existentes en el curso calendarizado de los cupos disponibles (tabla *disponibilities*) para los bloques (tabla *blocks*) que dure la prueba. Dado lo anterior, si el cupo por bloque se define como 300 alumnos concurrentes, para una prueba de duración 2 bloques que no presenta calendarizaciones anteriores, cuando un profesor calendarice un curso de 35 alumnos a las 14:00 horas, los cupos disponibles quedarán como sigue: de 14:00 a 14:59 265 cupos y de 15:00 a 15:59 265 cupos. La idea de este parámetro es tener un mecanismo para medir la capacidad de la aplicación y obligar a que la demanda sea lo más uniforme posible. Si por algún motivo existen alumnos que no alcancen a rendir una evaluación, se permitirá la recalendarización del curso, pudiendo rendir la prueba sólo aquellos alumnos que no presenten resultados (tabla *results*).

4.6.3 Revisión y descarga de resultados

Cuando un alumno termina una evaluación, desde la tabla *answers* se deben contar sus respuestas buenas, malas y omitidas, dando origen a un registro de resultado en la tabla *results*. Estos resultados además son segmentados según las habilidades y ejes temáticos medidos en la evaluación, dando origen a un registro por cada habilidad en la tabla *habilities_results* y a un registro por cada eje temático en la tabla *results_thematics*. De forma análoga a la recién descrita, resultados para curso y nivel educacional deben ser calculados y segmentados según habilidades y ejes temáticos, utilizándose las tablas *summaries*, *habilities_summaries* y *summaries_thematics* para los resultados del curso; y *level_results*, *levels_rhabilities* y *levels_rthematics* para resultados de nivel educacional. A continuación se muestra un esquema que define la estructura de resultados planteada (Figura 19).

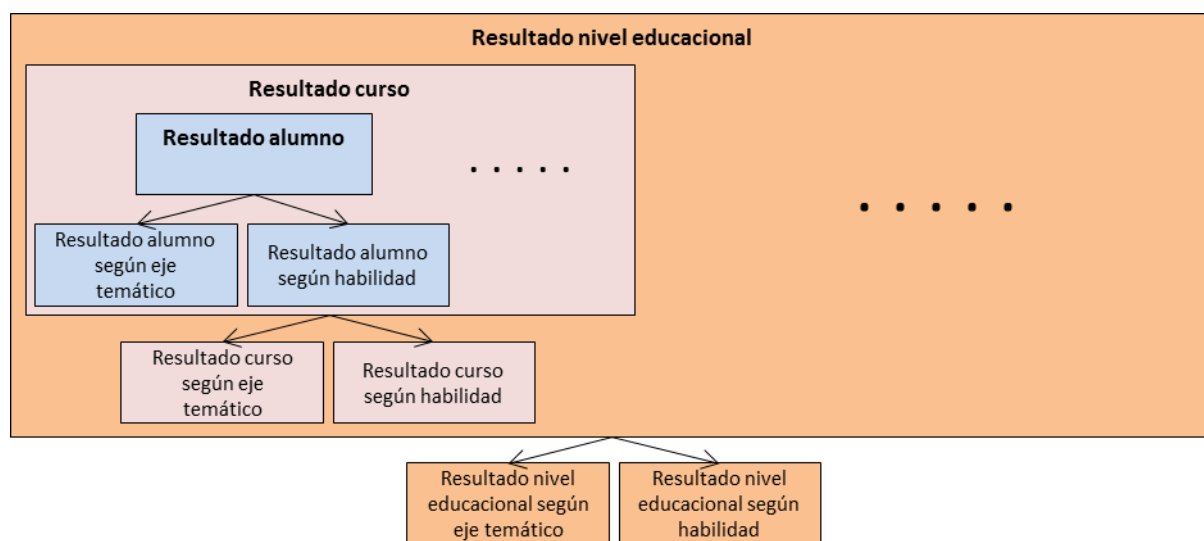


Figura 19: Estructura de resultados de evaluaciones

Los indicadores para curso y nivel son expresados en términos de promedios.

4.7 Módulo alumnos

Este módulo provee a los alumnos las funcionalidades para la rendición de pruebas y revisión de los resultados obtenidos. A continuación se presenta una simplificación del modelo de datos con las tablas relevantes para este módulo (Figura 20).

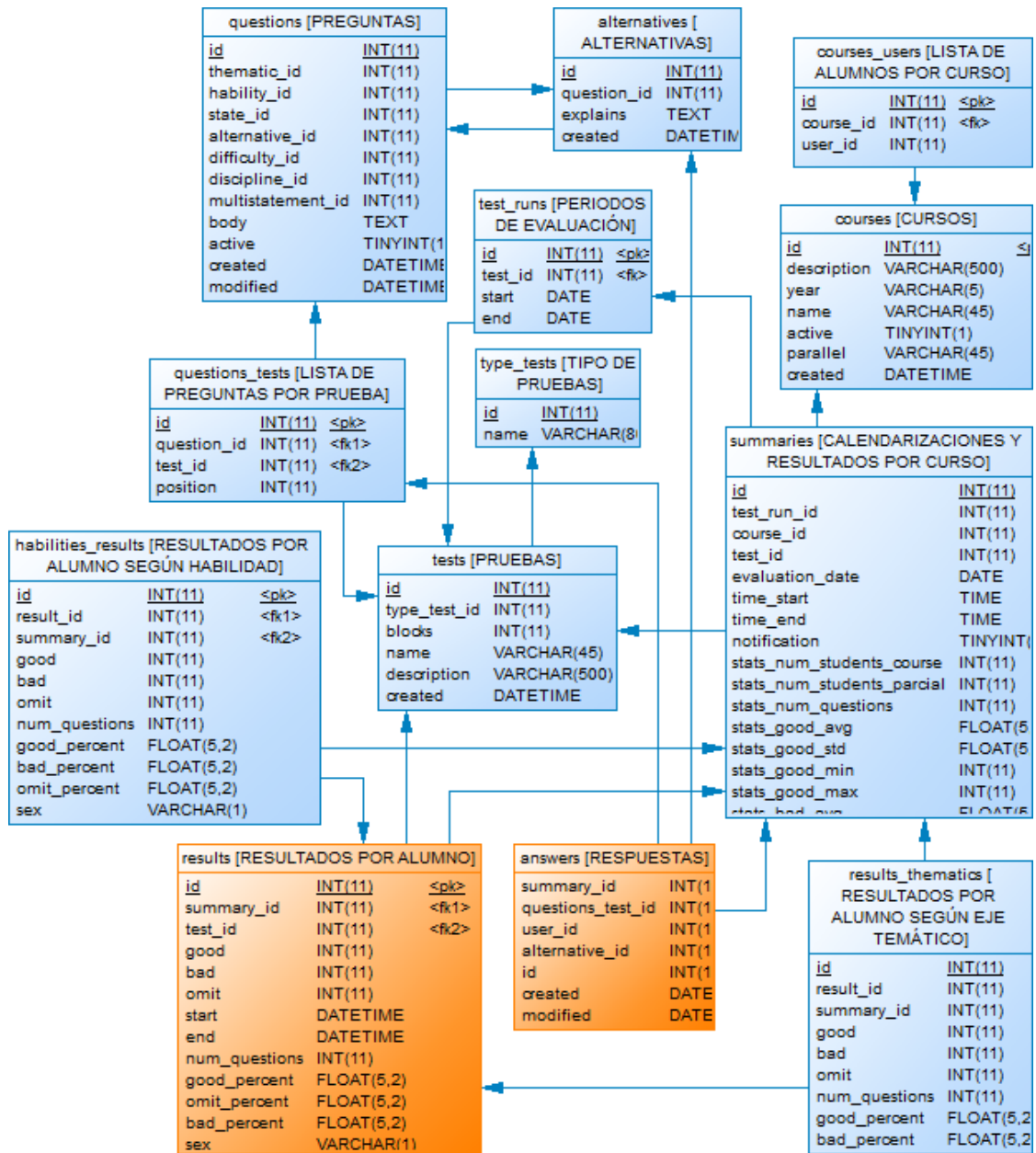


Figura 20: Modelo de datos simplificado, módulo alumnos

Un alumno puede rendir una evaluación sólo en el horario definido por el profesor (registrado por la tabla *summaries*). Cada respuesta que da el alumno se almacena en la tabla *answers*. Cuando el alumno determina que ha completado la prueba, o bien el tiempo de resolución se ha agotado, se calcula su resultado generando un registro en la tabla *results*, almacenando respuestas buenas, malas y omitidas en términos absolutos y porcentuales (campos *good*, *bad*, *omit*, *good_percent*, *bad_percent* y *omit_percent*), las horas de inicio y fin (campos *start* y *end*), el número de preguntas

que tenía la prueba que acaba de responder (campo *num_questions*) y el sexo del alumno (campo *sex*). Los datos mencionados son calculables en cualquier momento a partir de la información de la prueba, las respuestas y el alumno; sin embargo para no tener que repetir este cálculo, y facilitar la construcción de estadísticas con mayor nivel de agregación, estos son obtenidos una única vez y almacenados por la aplicación. Adicionalmente, cómo se explicó en el punto anterior, estos resultados son segmentados por habilidad y eje temático (tablas *abilities_results* y *results_thematics*).

5 Implementación de la solución

En esta sección se detallan los principales aspectos de la implementación del diseño expuesto, mostrando las pantallas con mayor relevancia de la aplicación.

5.1 Componente control de acceso

El manejo de usuarios se implementa en base al plugin para CakePHP *SparkPlug*, que permite integrar a la aplicación las funcionalidades de *Ingreso de usuarios*, *Cambio de contraseña*, *Recuperación de contraseña* y -la más importante- *Control de acceso*. El ingreso de usuarios se realiza mediante credenciales previamente establecidas por el administrador, a través de un formulario ubicado en la página inicial de la aplicación, según se observa en la imagen siguiente (Figura 21).

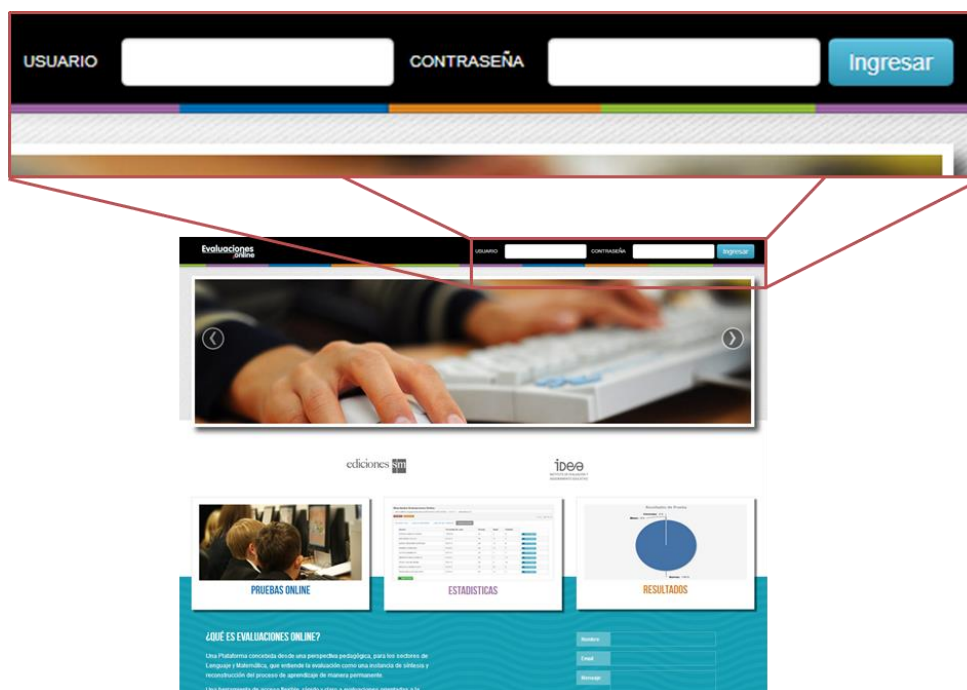


Figura 21: Pantalla de ingreso

Una vez proporcionadas las credenciales por el usuario, se realiza el procedimiento de autenticación y determinación de su perfil, el que sirve de identificador para la carga del menú correspondiente. Dicho menú se obtiene a partir de un archivo YML¹ con el nombre del perfil (por ejemplo: *profesor.yml*), el que define un árbol de URLs según muestra el Extracto de código 1.

¹ YAML es el acrónimo recursivo de *YAML Ain't Markup Language* y corresponde a un formato de serialización de datos legible por humanos inspirado en lenguajes como XML, C, Python, Perl.

```

nav:
  Inicio:
    url: /landing/coordinator
  Cursos:
    url: /courses/index
  subnav:
    Ver Todos los Cursos:
      url: /courses/index
    Agregar Estudiantes:
      url: /schools/loadstudents
  ...

```

Extracto de código 1: Menú perfil profesor (coordinador_ evaluaciones.yml)

El menú anterior es visualizado según muestra la Figura 22.

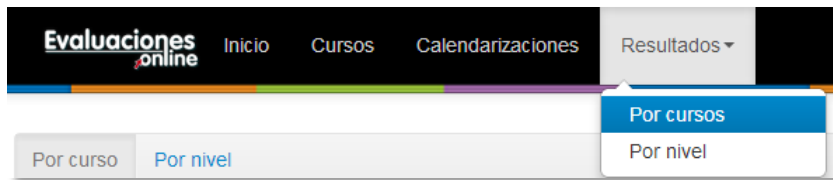


Figura 22: Menú de usuario

Cada funcionalidad disponible para los usuarios de la aplicación corresponde a un método de una clase controlador. Cada clase controlador hereda de la clase `AppController`, que a su vez hereda de `Controller`. Esta última, perteneciente al core del framework, define métodos que son llamados antes y después de ciertas actividades en el ciclo de una petición (visto en la Figura 1), denominados *callbacks*, característica que se utilizará para la validación del acceso de los usuarios a las funcionalidades de la aplicación. A continuación en la Figura 23 se presenta el ciclo extendido de una petición en CakePHP.

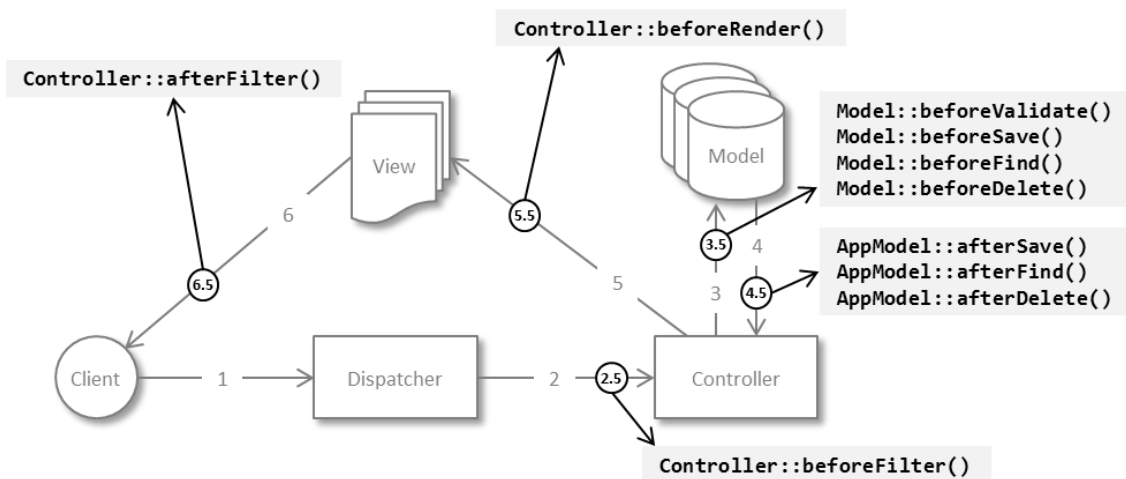


Figura 23: Ciclo extendido de una petición en CakePHP

Dado lo anterior, se tiene que `AppController->beforeFilter()` nos provee de punto único de control para el acceso a las funcionalidades de la aplicación, permitiendo que la validación se realice según muestra el Extracto de código 2.


```

class AppController extends Controller {
    function beforeFilter() {
        SparkPlugIt($this);
    }
    ...
}

```

Extracto de código 2: Control de acceso (app_controller.php)

La función SparkPlugIt recibe al objeto controlador, el que contiene toda la información de la petición, encargándose de autorizar o prohibir el acceso del usuario a la funcionalidad solicitada.

5.2 Componente mensajería

Según se explicó en el capítulo anterior, la aplicación debe enviar mensajes vía correo a los usuarios ante diversos eventos. Como ésta se trata de una lógica habitual en un gran número de aplicaciones, se decidió encapsularla en un plugin para CakePHP denominado *Emailator*. Los principales elementos de este plugin son:

- **Config**: Archivo de configuración que define los eventos.
- **EmilioComponent**: Clase componente que proporciona a los controladores la funcionalidad de programar mensajes.
- **EmailatorShell**: Clase que ejecuta la lógica de envío y que será instanciada por un cron cada 5 minutos.

Los eventos, con sus respectivos tiempos de envío y mensaje (vistos en la Tabla 3), se definen como sigue (Extracto de código 3).

```

Configure::write('Emailator.events', array(
    'new_summary' => array(
        'coordinador_evaluaciones' => array(
            'template' => 'email_new_summary_coordinator',
            'event_date_delay' => '+0 seconds',
            'subject_format' => 'Se ha programado una nueva evaluación'
        ),
        'alumno' => array(
            'template' => 'email_new_summary_student',
            'event_date_delay' => '+0 seconds',
            'subject_format' => 'Hola %s, tienes una nueva evaluación online programada!'
        )
    ),
    'edit_summary' => array(
        'coordinador_evaluaciones' => array(
            'template' => 'email_edit_summary_coordinator',
            ...

```

Extracto de código 3: Definición de eventos para envío de mensajes (config.php)

Mientras que el componente EmilioComponent se utiliza según se muestra a continuación (Extracto de código 4).

```

$eventData['event_type'] = 'new_summary';
$eventData['event_profile'] = 'alumno';
$eventData['event_date'] = date(Configure::read('Emailator.settings.date_format'));
...
//se agrega el nombre y correo de cada uno de los alumnos
foreach($users as $user) {
    ...
    $emailsData['email_to'] = $data;
}
$this->Emilio->create($eventData, $emailsData);
...

```

Extracto de código 4: Utilización de EmilioComponent para el envío de mensajes (emilio.php)

Un ejemplo de mensaje se puede apreciar a continuación (Figura 24).

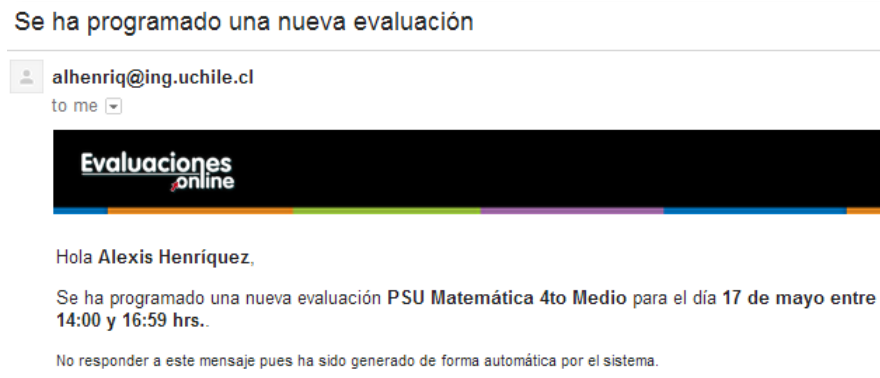


Figura 24: Correo enviado por la aplicación

5.3 Componente generación gráficos e informes

El diagrama mostrado en Figura 13, definido en términos de elementos del framework, consiste en funciones a nivel de modelo para la obtención de estadísticas, la clase helper ChartsHelper que utilizará la librería **Highcharts** [8] para la generación de gráficos, una vista que renderizará dichos gráficos en un informe HTML y la clase vista WkHtmlToPdfView que, haciendo uso de la librería de conversión HTML a PDF **wkhtmltopdf** [9], permitirá obtener el informe PDF descargable.

Los gráficos que se utilizarán serán de barra, de barra porcentual y torta, según se observa en el Extracto de código 5.

```

class ChartsHelper extends AppHelper {
    function bar($id, $title, $subtitle, $yaxis, $categories, $series) {...}
    function bar100($id, $title, $subtitle, $yaxis, $categories, $series) {...}
    function pie($id, $title, $series) {...}
    ...
}

```

Extracto de código 5: Helper para renderización de gráficos (charts.php)

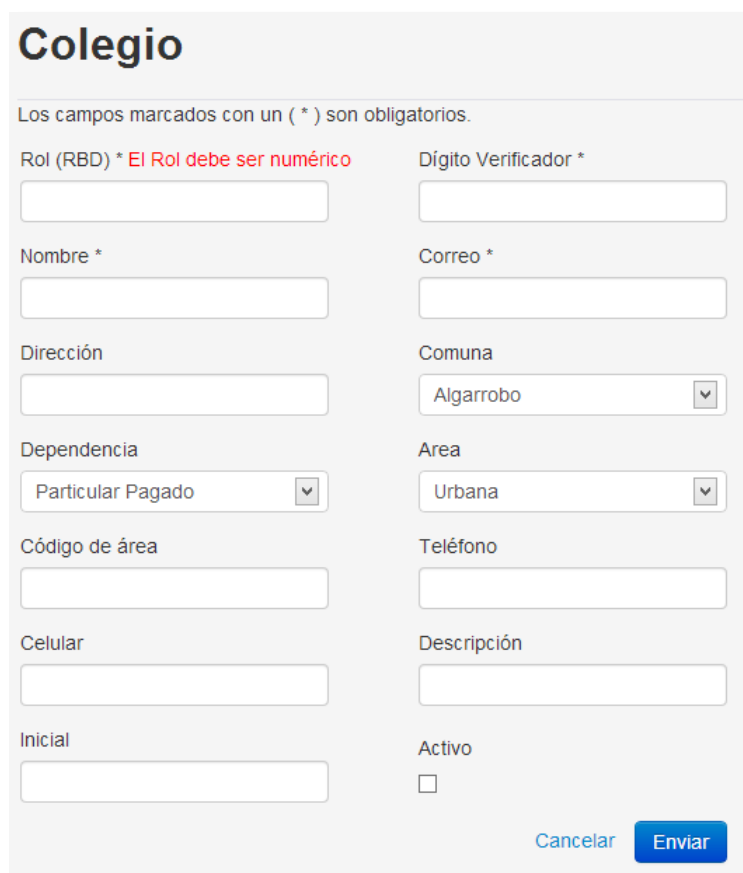
Esta clase es luego utilizada en las vistas, que son las que reciben los datos para generar los gráficos.

5.4 Módulo administración

A continuación se destacan los aspectos más relevantes de la implementación del módulo administración: administración de los colegios y profesores, mantenedores de disciplinas, habilidades y ejes temáticos, y definición de períodos de evaluación; mostrando sus principales pantallas.

5.4.1 Administración de colegios y profesores

Como se mencionó anteriormente, los colegios existentes en el listado que publica el Ministerio de Educación fueron previamente importados, por lo que nuevos colegios deben ser ingresados a través de la aplicación. A continuación se muestra el formulario de ingreso para nuevos colegios (Figura 25).



Colegio

Los campos marcados con un (*) son obligatorios.

Rol (RBD) * El Rol debe ser numérico	Dígito Verificador *
<input type="text"/>	<input type="text"/>
Nombre *	Correo *
<input type="text"/>	<input type="text"/>
Dirección	Comuna
<input type="text"/>	Algarrobo <input type="button" value="v"/>
Dependencia	Area
Particular Pagado <input type="button" value="v"/>	Urbana <input type="button" value="v"/>
Código de área	Teléfono
<input type="text"/>	<input type="text"/>
Celular	Descripción
<input type="text"/>	<input type="text"/>
Inicial	Activo
<input type="text"/>	<input type="checkbox"/>

Cancelar

Figura 25: Formulario agregar colegio

En la imagen anterior se observa que en caso de existir un error de validación, éste es indicado con un texto en rojo sobre el campo correspondiente. Dicha validación se realiza en el lado del servidor a nivel de modelo, como muestra el Extracto de código 6.

```

class School extends AppModel {
    public $validate = array(
        'rbd' => array(
            'numeric' => array(
                'rule' => 'numeric',
                'message' => 'El Rol debe ser numérico',
            ),
            'between' => array(
                'rule' => array('between', 1, 6),
                'message' => 'El Rol debe tener un largo entre 1 y 6 caracteres',
            )
        ),
    ),
    ...
}

```

Extracto de código 6: Validación a nivel de modelo (school.php)

Las reglas de validación definidas en el modelo son aplicadas antes de cualquier operación de guardado.

Para todos los listados de elementos en la aplicación se utiliza una tabla con filtros en su cabecera, la que facilita la búsqueda de elementos específicos. Estos filtros fueron construidos utilizando 2 plugins: Search y Chosen. El primero se trata de un plugin para CakePHP que tiene la característica de ser compatible con la paginación que el framework proporciona por defecto, mientras que el segundo corresponde a un plugin de jQuery que permite convertir cualquier *input* de tipo *select* en una barra de búsqueda. A continuación se presenta la pantalla de listado de colegios (Figura 26).

The screenshot shows the 'Evaluaciones online' interface. At the top, there is a navigation bar with links: Inicio, Cursos, Mantenedores, Colegios, Lanzamientos de Pruebas, Calendarizaciones, and Usuario. Below the navigation bar, there are two buttons: 'Ver Todos los Colegios' and 'Agregar Colegio'. The main content is a table with the following columns: Nombre, región, comuna, dependencia, activo, and Acciones. The table contains three rows of data. A search dropdown menu is open over the first row, showing a list of school names. The 'Acciones' column for each row contains buttons for 'Ver', 'Editar', 'Eliminar', and 'Acceso'.

Nombre	región	comuna	dependencia	activo	Acciones
...	-- TODO... --	-- TODO... --	-- TODO... --	-- --	Filtrar
...	Arica y Parinacota	Arica	Municipal	No	Ver, Editar, Eliminar, Acceso
...	Arica y Parinacota	Arica	Municipal	No	Ver, Editar, Eliminar, Acceso
...	Arica y Parinacota	Arica	Municipal	No	Ver, Editar

Figura 26: Listado de colegios

El botón *Acceso* en la imagen anterior lleva a la pantalla de asignación de permisos para el colegio correspondiente (Figura 27).

Evaluaciones online Inicio Cursos Mantenedores Colegios Lanzamiento		
Pruebas Habilitadas		
LICEO INSTITUCIÓN DEL JUAN BOSCH CUNAVAR tiene acceso a las siguientes pruebas por nivel y disciplina:		
Nivel	Lenguaje	Matemática
Primero Básico		
Segundo Básico		
Tercero Básico		
Cuarto Básico		
Quinto Básico		

Figura 27: Asignación de permisos a colegio

Los permisos son desplegados en una tabla donde las filas indican el nivel educacional, las columnas la disciplina y cada celda contiene los distintos tipos de prueba disponibles para los parámetros ya mencionados. Con el fin de que el usuario no pierda de vista la perspectiva general cada vez que activa/desactiva un permiso, esta acción se lleva a cabo mediante AJAX², según se aprecia a continuación (Extracto de código 7).

```

$( '.test' ).bind( 'click', function() {
    $.ajax( {
        url: '/schools/activeTest',
        async: true,
        success: function( data ) {
            if( data.response == 'Ok' ) {
                message = "Se ha activado SIMCE " + data.discipline + " " + data.level;
                $( imagen ).attr( 'src', base + 'img/icons/simce_viole_ON.png' );
            } else {
                message = "Se ha desactivado SIMCE " + data.discipline + " " + data.level;
                $( imagen ).attr( 'src', base + 'img/icons/simce_viole_OFF.png' );
            }
        }
    }
    ...

```

Extracto de código 7: Llamada AJAX para activar/desactivar permisos a un colegio (levels.js)

El código anterior está escrito en JavaScript haciendo uso de la librería jQuery, lo que permite independencia de la implementación específica de XMLHttpRequest³ que posea el navegador. En

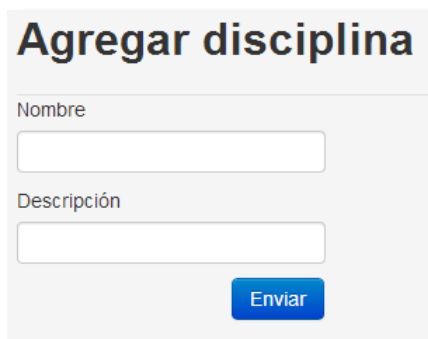
² AJAX es el acrónimo de *Asynchronous JavaScript And XML*, técnica de programación web que permite ejecutar acciones en el navegador mientras se mantiene la comunicación asíncrona con el servidor en segundo plano, posibilitando realizar cambios sobre las páginas sin necesidad de recargarlas.

³ XMLHttpRequest es un objeto JavaScript -diseñado por Microsoft para Internet Explorer y luego adoptado como estándar- [15] que provee un mecanismo fácil para la obtención de datos desde una URL sin la necesidad de recargar la página. Como muchas funcionalidades de JavaScript presenta algunas diferencias de implementación dependiendo del navegador, por lo que la utilización de una librería que proporcione una única interfaz de acceso -como jQuery- se ha hecho una práctica común y necesaria.

él se muestra cómo se realiza una petición asíncrona a la página */schools/activeTest* para luego, en caso de éxito, activar o desactivar el ícono de la prueba según corresponda.

5.4.2 Mantenedores de disciplinas, habilidades y ejes temáticos

En caso de que en futuro se adicionen nuevas disciplinas (recordemos que por ahora sólo se consideran Lenguaje, Matemática, Ciencias e Historia) se proporciona un formulario para este fin (Figura 28).



El formulario 'Agregar disciplina' tiene un título 'Agregar disciplina' en negrita. Abajo del título hay dos campos de texto: 'Nombre' y 'Descripción'. Debajo de los campos hay un botón azul con el texto 'Enviar'.

Figura 28: Formulario agregar disciplina

A continuación se muestra la pantalla que despliega el listado de disciplinas (Figura 29).



La pantalla muestra el menú de navegación 'Mantenedores' con submenús 'Disciplinas', 'Habilidades' y 'Ejes Temáticos'. El título principal es 'Disciplinas'. Hay una barra de acciones con 'Ver Todas las Disciplinas' y 'Agregar Dis...'. La tabla tiene tres columnas: 'Nombre', 'descripción' y 'Acciones'. Se ven las disciplinas 'Lenguaje' y 'Matemática'. Hay un filtro y botones de 'Ver', 'Editar' y 'Eliminar' para cada una.

Nombre	descripción	Acciones
-- TO... --	-- TODOS --	Filtrar
Lenguaje	Lenguaje	Ver, Editar, Eliminar
Matemática	Matemática	Ver, Editar, Eliminar

Figura 29: Listado de disciplinas

Formularios y pantallas de listado análogas a las anteriores para habilidades y ejes temáticos fueron construidas.

5.4.3 Definición de períodos de evaluación

Para cada prueba presente en la aplicación se puede definir un período de evaluación, según muestra la siguiente imagen (Figura 30).

El formulario 'Nuevo lanzamiento de prueba' contiene los siguientes campos:

- Nivel:** Selector desplegable con 'Primero Básico'.
- Prueba:** Selector desplegable con '2013 SIMCE Naturaleza Cua'.
- Inicio:** Campos para mes ('Junio'), día ('6') y año ('2013').
- Final:** Campos para mes ('Julio'), día ('6') y año ('2013').
- Botón:** 'Inicio'.

Figura 30: Formulario definición de período de evaluación

Desde el listado de períodos de evaluación (o “Lanzamientos de prueba” como se le denomina en la interfaz), se pueden descargar los resultados detallados de un período de evaluación, para su validación por parte de la editorial, como se observa en la siguiente imagen (Figura 31).

La interfaz muestra un menú de navegación con las siguientes opciones: Evaluaciones online, Inicio, Cursos, Mantenedores, Colegios, Lanzamientos de Pruebas, y Calendarizaciones. Debajo del menú, hay dos botones: 'Ver Todos los Lanzamientos de Pruebas' y 'Realizar Nuevo Lanzamiento de Prueba'. El título principal de la sección es 'Lanzamientos de Pruebas'.

nivel	prueba	inicia	termina	creado	Acciones
-- TODO... ▾	-- TODO... ▾				Filtrar
Cuarto Básico	2012 SIMCE 1 Lenguaje Cuarto Básico	14-05-2012	07-06-2012	07-04-2012 21:06	Descargar Resultados
Cuarto Básico	2012 SIMCE 1 Matemática Cuarto Básico	14-05-2012	07-06-2012	07-04-2012 21:06	Descargar Resultados
Cuarto Básico	2012 SIMCE 1 Naturales Cuarto Básico	14-05-2012	07-06-2012	07-04-2012 21:07	Descargar Resultados
Cuarto Básico	2012 SIMCE 1 Sociedad Cuarto Básico	14-05-2012	07-06-2012	07-04-2012 21:07	Descargar

Figura 31: Listado de períodos de evaluación

Dichos resultados son descargados en formato Excel y contienen todas las respuestas (para todos los colegios y alumnos) de una prueba en un período de evaluación dado, y se obtienen del cruce de varias tablas, como se desprende del código a continuación (Extracto de código 8).

```

initQuery();
addStatement('SELECT schools.rbd, schools.name, users.rut, ');
$questionNumber = 1;
foreach($questions_id as $question_id) {
    $colsTestRunResults[] =
        "SUM(IF(questions_test_id = {$question_id}, alternative_id, 0))
        AS question_{$questionNumber}";
    $questionNumber++;
}
addStatement(implode($colsTestRunResults, ','));
addStatement('FROM answers');
addStatement('INNER JOIN summaries ON (answers.summary_id = summaries.id)');
addStatement('INNER JOIN users ON (answers.user_id = users.id)');
addStatement('INNER JOIN courses_users ON (users.id = courses_users.user_id)');
addStatement('INNER JOIN courses ON (courses_users.course_id = courses.id)');
addStatement('INNER JOIN schools ON (courses.school_id = schools.id)');
addStatement("WHERE summaries.test_run_id = {$test_run_id}");
addStatement('GROUP BY users.rut ORDER BY schools.rbd');
$answersByTestRunId = runQuery();

```

Extracto de código 8: Consulta SQL para descarga de resultados de un período (test_run.php)

Como se puede observar, se utiliza de forma conveniente la función SUM en conjunto con IF para obtener sólo el identificador de la respuesta asociada a la pregunta correspondiente (el resto serán 0's que no aportarán a la suma). El archivo resultante se aprecia en la siguiente imagen (Figura 31).

	A	B	C	D	E
1	school_rbd	school_name	student_rut	question_1	question_2 qu
2	Pauta	Pauta	Pauta	24	28
3	579	COLEGIO SEMINARIO CONCEPCION	21120487-3	24	28
4	579	COLEGIO SEMINARIO CONCEPCION	21120487-3	24	27
5	579	COLEGIO SEMINARIO CONCEPCION	21120488-1	24	29
6	579	COLEGIO SEMINARIO CONCEPCION	21120488-2	23	28
7	579	COLEGIO SEMINARIO CONCEPCION	21120488-4	23	29
8	579	COLEGIO SEMINARIO CONCEPCION	21120488-2	24	29
9	579	COLEGIO SEMINARIO CONCEPCION	21120488-1	24	29

Figura 32: Resultados detallados de un período de evaluación en Excel

La primera fila contiene la pauta de evaluación, es decir, el identificador de la respuesta correcta para cada pregunta.

5.5 Módulo edición

El editor tiene la facultad de crear/editar preguntas en la aplicación mediante el siguiente formulario (Figura 33).

Editar pregunta

Prueba

2012 SIMCE 1 Matemática Cuarto Básico
2012 SIMCE 1 Matemática Primero Medio
2012 SIMCE 1 Matemática Segundo Medio
2012 PSU 1 Matemática Tercero Medio

Disciplina: Matemática

Dificultad: Moderada

Temática: Estimación de cantidades

Estado: aceptada

Habilidad: Estimar

Activo:

Contenido

¿Cuántas bolitas del mismo tamaño que las del frasco 1 podría contener el frasco 2 aproximadamente?

Frasco 1: 50 bolitas

Frasco 2: ?

Alternativa # 0

Correcta

50

Figura 33: Formulario de creación/editación de preguntas

En él las alternativas se van creando dinámicamente a través de AJAX y, sus textos, así como el de la pregunta, utilizan el editor WYSIWYG⁴ CKEditor [10], en conjunto con KCFinder [11] para la incrustación de imágenes. A continuación se muestra la vista previa de una prueba (Figura 34).

⁴ Acrónimo de What You See Is What You Get, se refiere a editores de texto que permiten aplicar formato de manera visual, obteniendo una vista previa del resultado final.

Evaluaciones online Inicio Textos de Preguntas Preguntas Pruebas

Prueba #70 PSU Disciplina: Lenguaje Creado: 2013-03-15

Nombre: 2013 PSU Lenguaje Cuarto Medio

Ver Todo

Preguntas

Pregunta #1 Habilidad: Caracterizar Temática: Conceptos básicos Editar Desasociar Eliminar

Pregunta #2 Habilidad: Identificar Temática: Conceptos básicos Editar Desasociar Eliminar

Pregunta #3 Habilidad: Inferir Temática: Conceptos básicos Editar Desasociar Eliminar

Texto asociado: Lenguaje IV texto 1

¿Cuál es la finalidad del texto 1?

Alternativas:

- Convencer al director de corregir el lenguaje de sus publicaciones.
- Refutar el uso de "como" en contextos innecesarios.
- Informar sobre las nuevas formas de usar la palabra "como".
- Explicar a los lectores cuándo deben utilizar la palabra "como".
- Educar a la población en el buen uso de la lengua.

Pregunta #4 Habilidad: Interpretar Temática: Conceptos básicos Editar Desasociar Eliminar

Pregunta #5 Habilidad: Caracterizar Temática: Conceptos básicos Editar Desasociar Eliminar

Figura 34: Vista previa de una prueba

Esta interfaz presenta las preguntas en forma de acordeón vertical, permitiendo la visualización del detalle de cada pregunta sin perder de vista al resto de ellas. La habilidad y temática medidas se muestran al lado del número de la pregunta como etiquetas, mientras que los botones en azul con flechas hacia arriba y abajo permiten cambiar la posición de la pregunta dentro de la prueba.

A medida que las preguntas registren respuestas, estarán disponibles sus indicadores de dificultad, homogeneidad y discriminación; según se aprecia en la siguiente imagen (Figura 35).

Evaluaciones online Inicio Textos de Preguntas Pregunta	
2012 SIMCE 1 Matemática Primero Medio	
Pregunta N° 158 Editar	
Autor	Editor Editor
Temática	Números
Habilidad	Aplicar, Interpretar
Estado	aceptada
Índice de Dificultad	0.3710
Correlacion Pearson (Índice de Homogeneidad)	0.3777
Correlacion Pearson Corregida (Índice de Homogeneidad Corregido o Índice de Discriminación)	0.3135

Figura 35: Detalle pregunta e indicadores de validación

Estos indicadores permiten al editor mejorar continuamente los instrumentos de evaluación.

5.6 Módulo profesores

A continuación se destacan los aspectos más relevantes de la implementación del módulo profesores: creación de cursos, calendarización de evaluaciones y revisión y descarga de resultados; mostrando sus principales pantallas.

5.6.1 Creación de cursos

El primer paso para poder tomar una evaluación online es subir la nómina de alumnos. Para esto, el profesor debe descargar el archivo Excel de ejemplo disponible en la aplicación y luego subirlo con la información de los alumnos, según se aprecia en la imagen siguiente (Figura 36).



Figura 36: Formulario de ingreso de alumnos mediante archivo Excel

El botón *Ver permisos* permite al profesor visualizar las evaluaciones disponibles para su colegio, en un formato similar al mostrado en la Figura 27. Para la lectura del archivo en Excel (cuyo formato se define en la Figura 17) se hace uso de la librería PHPEXcel [12], la que permite una sencilla manipulación del archivo según muestra el Extracto de código 9.

```

$rut      = $objWorksheet->getCellByColumnAndRow(0, $row)->getValue();
$first_name = $objWorksheet->getCellByColumnAndRow(1, $row)->getValue();
$last_name  = $objWorksheet->getCellByColumnAndRow(2, $row)->getValue();
...

```

Extracto de código 9: Lectura archivo Excel (school.php)

Una vez se han ingresado las nóminas de alumnos, los cursos creados se visualizan en la siguiente pantalla (Figura 37).

Evaluaciones online			
Inicio	Cursos	Calendarizaciones	
Ver Todos los Cursos	Agregar Estudiantes		
Nivel	Paralelo	Año	Acciones
-- TODOS --			Filtrar
Cuarto Básico	A	2012	Ver
Cuarto Básico	B	2012	Ver

Figura 37: Listado de cursos

Mediante el botón *Ver*, se puede acceder al detalle del curso correspondiente (Figura 38).

Identificador	RUT	Nombre	Apellido Pa
1011871	21337081-4	Carolina Jazmin	Almendros
1011872	21142516-4	Cristobal Andres	Araya
1011873	21178381-5	Danae Betzabe	Araya
1011874	21226216-1	Katerina Alejandra	Ariles
1011875	21073794-4	Catalina Belen	Arredondo
1011876	21140381-3	Maria Ignacia	Ariles
1011877	21296703-1	Valentina Javiera	Castillo

Figura 38: Detalle de un curso

Con la creación de un curso se generan credenciales de usuario para cada uno de los alumnos que lo componen.

5.6.2 Calendarización de evaluaciones

Una vez el profesor ha ingresado los cursos, puede ver una lista de las evaluaciones que están disponibles para su calendarización y seleccionar su fecha de aplicación, según se observa en la Figura 39.

Calendarizar 2013 SIMCE 1 Matemática Cuarto Básico

2013-07-09

Enviar

Seleccione bloque

- Seleccione bloque
- Bloque 1 / 07:00:00
- Bloque 2 / 08:00:00
- Bloque 3 / 09:00:00
- Bloque 4 / 10:00:00
- Bloque 5 / 11:00:00
- Bloque 8 / 14:00:00
- Bloque 9 / 15:00:00
- Bloque 12 / 18:00:00
- Bloque 13 / 19:00:00
- Bloque 14 / 20:00:00

Figura 39: Formulario de cambio de fecha

La ausencia de los bloques 6, 7, 10 y 11 en el *select* desplegado en la imagen anterior, se debe a que la cantidad de alumnos que tiene el curso a calendarizar es mayor a la capacidad actual de dichos bloques. La aplicación permite el cambio en la fecha calendarizada en cualquier momento

dentro del período de evaluación vigente, sin embargo aquellos alumnos que registren resultados no pueden volver a rendir una misma evaluación. Esto es útil en caso de que por algún motivo uno o más alumnos no hayan alcanzado a rendir una prueba en la fecha reservada.

5.6.3 Revisión y descarga de resultados

A medida que los alumnos finalizan las evaluaciones, las estadísticas a nivel de curso pueden ser visualizadas por el profesor; en caso de que éstas sean parciales, se indica mediante una alerta que especifica cuántos alumnos son considerados en el análisis. La imagen siguiente muestra los resultados de una evaluación para un curso completo (Figura 40).



Figura 40: Resultados generales a nivel de curso

Los resultados anteriores corresponden a los promedios de un curso Cuarto Básico B para un ensayo SIMCE de la disciplina Sociedad. Como se mencionó en el capítulo anterior, también se presentan resultados desglosados según las habilidades y ejes temáticos, así como por género, según se observa en la imagen siguiente (Figura 41).



Figura 41: Gráfico de resultados detallados a nivel de curso

Del gráfico se desprende que en promedio los alumnos tuvieron un mejor desempeño en la habilidad de *Reconocer*, sin embargo, si se considera el género, los mejores resultados fueron obtenidos por las mujeres para la habilidad *Evaluar*.

Finalmente, como se muestra a continuación, los profesores pueden obtener los resultados a nivel de alumno (Figura 42).

Evaluaciones online Inicio Cursos Calendarizaciones Resultados ▾					
Resumen Curso Logro por Habilidades Logro por Ejes Temáticos Logro por Alumno					
Alumno	Porcentaje de Logro	Buenas	Malas	Omitidas	
Guillermo Andres Abalos	66.67 %	20	10	0	Ver Resultado
Vanessa Alejandra Araya	43.33 %	13	17	0	Ver Resultado
Cristobal Daniel Castro	66.67 %	20	10	0	Ver Resultado
Emilia Jose Chacon	63.33 %	19	11	0	Ver Resultado
Franco Alejandro Contreras	53.33 %	16	14	0	Ver Resultado
Jose Patricio Correa	43.33 %	13	17	0	Ver Resultado
Cristobal Martin Estay	60.00 %	18	12	0	Ver Resultado
Juan Jose Gallardo	63.33 %	19	11	0	Ver Resultado
Boris Hernan Garcia	50.00 %	15	15	0	Ver Resultado
Sergio Nicolas Guerrero	73.33 %	22	8	0	Ver Resultado

Figura 42: Resultados detallados de un curso a nivel alumno

Los resultados se muestran en forma de porcentaje de logro, otorgando libertad de interpretación al profesor. El resultado de cada alumno -visto en el punto siguiente- también es accesible por el profesor mediante el botón *Ver Resultado*.

Además de poder visualizar los resultados de una evaluación desde la aplicación, el profesor tiene la posibilidad de descargarlos en un archivo PDF para su revisión offline. Este PDF contiene la misma información que es posible de ver online, es decir: Resultados generales; Resultados por habilidad; Resultados por habilidad y género; Resultados por eje temático; Resultados por eje temático y género; y Resultados por alumno. A continuación se presenta una página del informe en PDF (Figura 43).

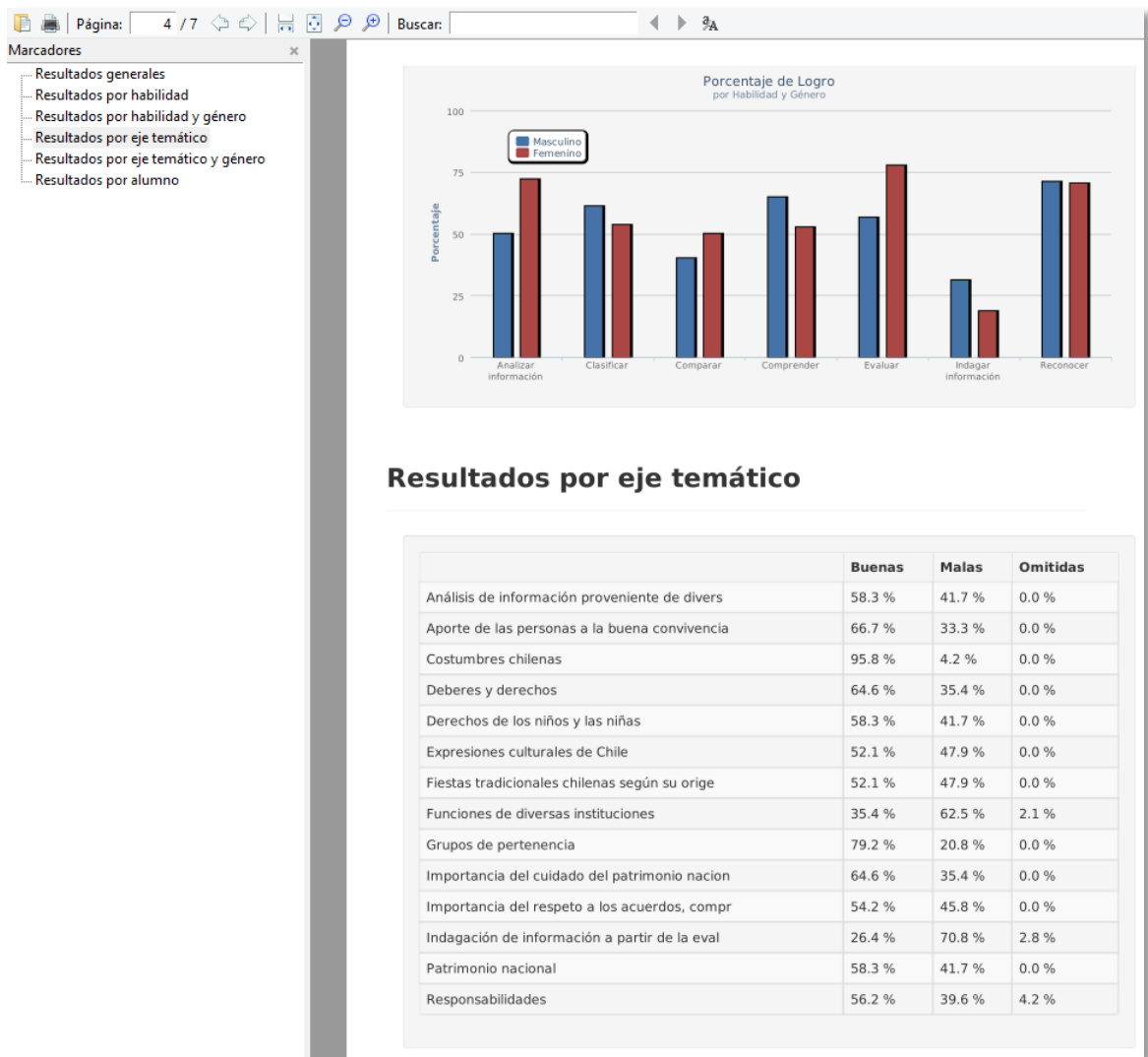


Figura 43: Informe en PDF

La información estadística en conjunto con los gráficos –que son renderizados mediante JavaScript en SVG⁵–, es almacenada en un archivo HTML el que luego es procesado por la librería *wkhtmltopdf* y convertido a PDF. Esta librería a su vez hace uso de Xvfb⁶, programa que permite el procesamiento gráfico por el lado del servidor. Como los gráficos tienen una sutil animación al momento de su despliegue, para su correcta conversión a PDF se introduce una instrucción de retraso en la renderización, según se observa a continuación en el Extracto de código 10.

```
xvfb-run /usr/bin/wkhtmltopdf --javascript-delay 1200 "informe.html" "informe.pdf"
```

Extracto de código 10: Conversión HTML a PDF (*wk_html_to_pdf.php*)

⁵ Acrónimo de Scalable Vector Graphics, corresponde a una especificación para describir gráficos vectoriales bidimensionales en formato XML utilizada por navegadores.

⁶ X virtual framebuffer, es un servidor X11 que ejecuta todas las operaciones gráficas en memoria, sin necesidad de hardware de pantalla. [16]

5.7 Módulo alumnos

Las acciones disponibles para los alumnos son la rendición de pruebas y revisión de sus resultados. Al momento de ingresar, si el alumno tiene una evaluación activa, se despliega el botón *Comenzar Prueba*, según muestra la imagen siguiente (Figura 44):

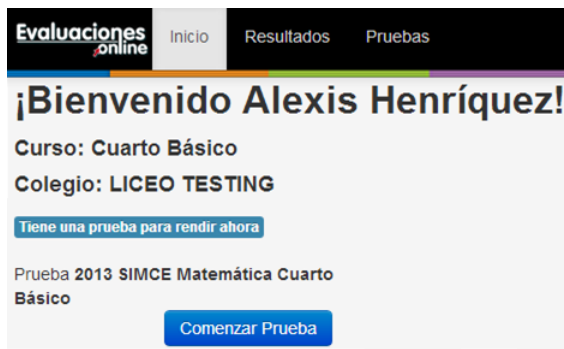


Figura 44: Pantalla de ingreso alumno

Dicho botón lo lleva a una interfaz libre de distracciones donde sólo puede observar las preguntas de la prueba a rendir, la que es cargada completamente en una única petición, desplegando una pregunta a la vez, como se observa a continuación en la Figura 45.

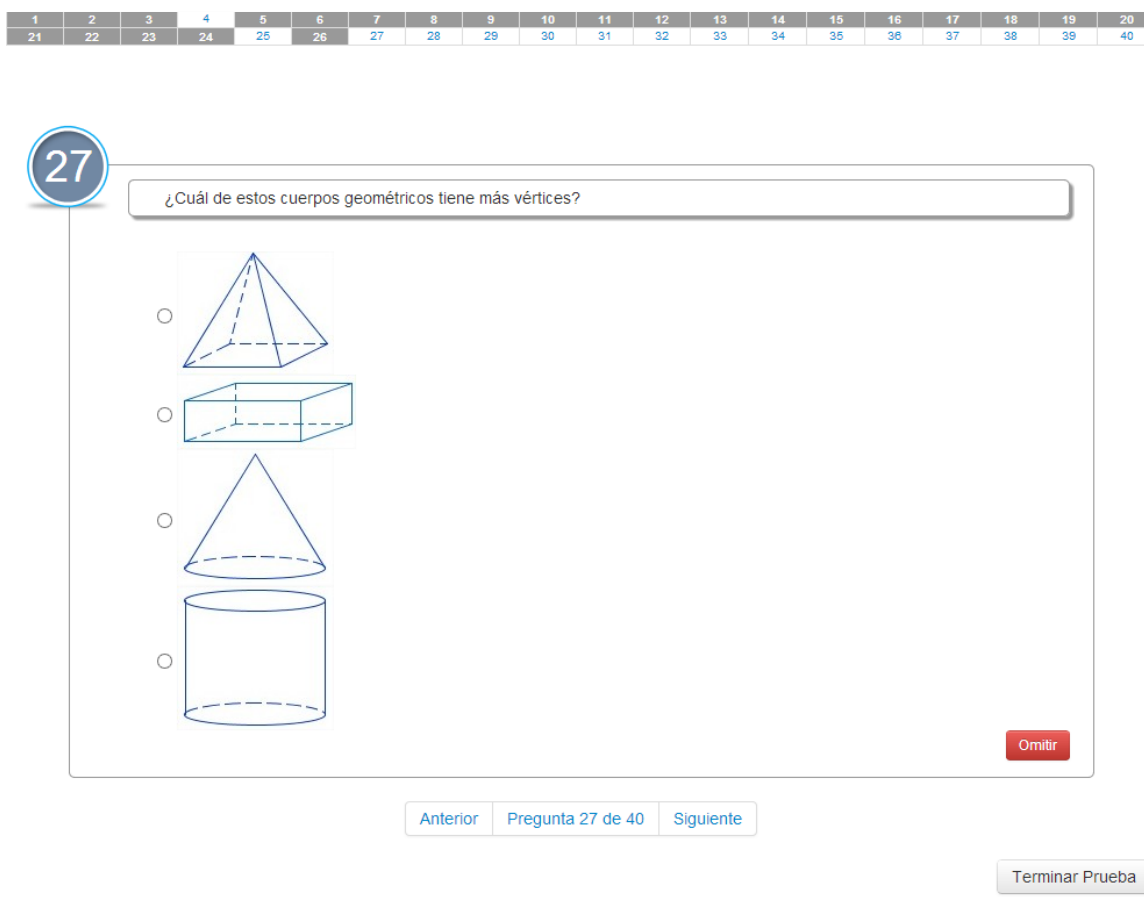


Figura 45: Pantalla de rendición de prueba

Las respuestas se envían vía AJAX al momento en que son registradas para evitar posibles pérdidas por errores de conexión. En la parte superior de la pantalla se despliega un índice rápido de la prueba, que sirve además para discriminar aquellas preguntas que aún no han sido respondidas (en fondo blanco), de las ya contestadas (en fondo gris).

Finalmente, cuando el alumno finaliza la prueba (haciendo clic en el botón *Terminar Prueba*), se despliegan sus resultados (Figura 46).

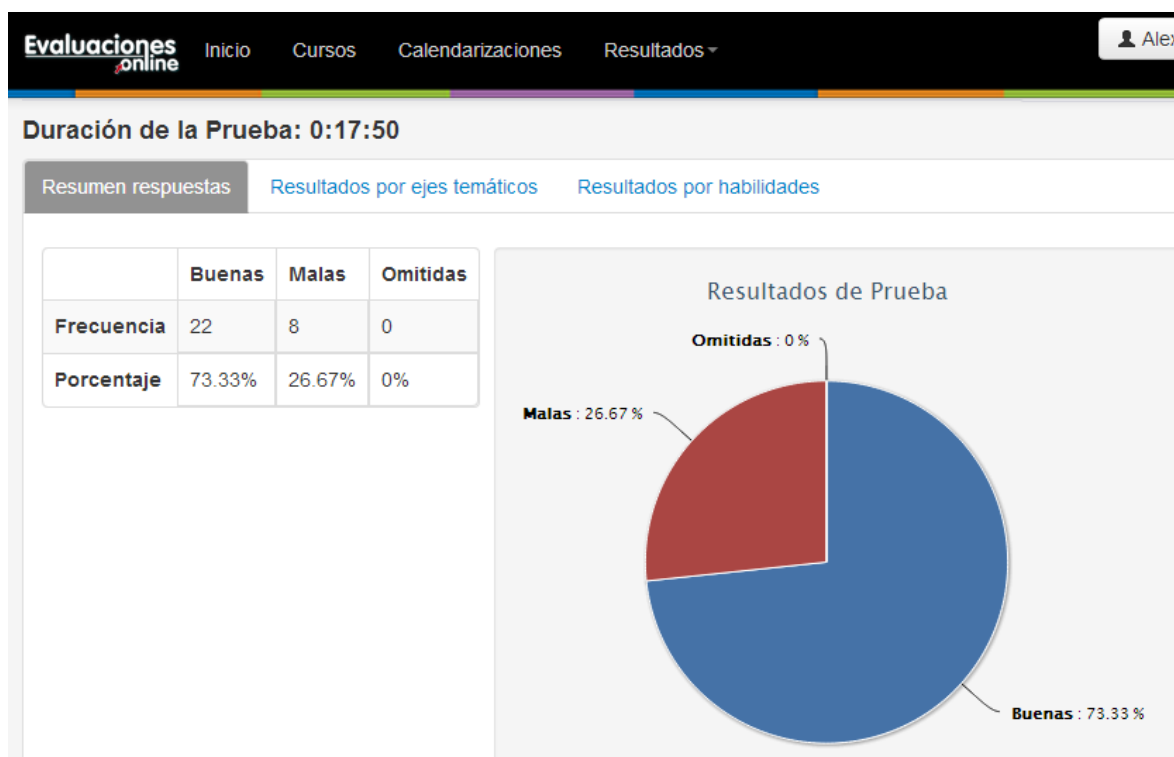


Figura 46: Resultados de un alumno

Como se aprecia en la imagen los resultados tiene el mismo formato que el visto para profesores, con la diferencia que sólo contienen la información del alumno.

5.8 Consideraciones de rendimiento

Con el objetivo de minimizar las falencias de rendimiento en la aplicación, se tomaron las medidas descritas a continuación.

- **Optimización de consultas:** CakePHP implementa el patrón ActiveRecord, el que a través de objetos ligados a registros en la base de datos, encapsula la lógica de las consultas, posibilitando la interacción con la base de datos solamente a través de dichos objetos y sus métodos [13]. Si bien la utilización de este patrón facilita diversos aspectos de la implementación (como la abstracción del lenguaje SQL o la legibilidad y reutilización de código), puede generar una sobrecarga de consultas (múltiples JOINS) y datos, lo que incrementa el peso de cada respuesta de la aplicación. Es por esto fundamental, especialmente para consultas complejas y/o extensas, especificar las relaciones y campos que se requieren, con el fin de evitar cualquier carga adicional. Para esto el framework

provee una serie de mecanismos como desvinculación de modelos y definición de la profundidad de consulta que fueron utilizados en la implementación.

- **Caché de cálculos y vistas de mayor tráfico:** Se detectaron 2 importantes funcionalidades que tenían el potencial de ser ineficientes. La primera de ellas fue la visualización de estadísticas y gráficos, que involucra un cálculo que cruza varias tablas de la base de datos. La segunda corresponde a la vista de rendición de prueba, asociada al perfil *Alumno*, que será la funcionalidad que presente la mayor concurrencia de toda la aplicación. Debido a esto se optó por almacenar un caché de datos, en el caso de las estadísticas y gráficos, en base de datos; mientras que para la vista de rendición de prueba, presentando una vista ya construida en vez de renderizarla cada vez.
- **Distribución de carga:** La mayor fuente de carga para la aplicación será la rendición de pruebas, que implica una masiva llegada de alumnos en un mismo período de tiempo. Es por esto que, para poder controlar la capacidad del servidor, se limitó la cantidad de alumnos que pueden rendir evaluaciones al mismo tiempo. Como se explicó en 4.6.2, se utiliza el parámetro *cupos* para conseguir dicho objetivo. La idea es maximizar el cupo manteniendo la utilización de recursos del servidor (y por tanto la capacidad de respuesta de la aplicación) en valores razonables.

6 Resultados y conclusiones

A continuación se presentan los resultados y conclusiones obtenidos.

6.1 Ampliación de la cobertura de evaluación

Este es el más importante y visible de los resultados. Mediante el proceso tradicional de aplicación, explicado anteriormente en el punto 2.3, se logra abarcar un máximo de 4 cursos diarios. Si se considera que un curso en promedio tiene 29 alumnos (dato obtenido sobre un universo de 747 cursos), se obtiene que mediante el proceso tradicional se pueden rendir 116 evaluaciones diariamente.

A través de la aplicación se han recibido más de 5.000 alumnos, con un total cercano a las 17.000 evaluaciones rendidas (Figura 47).



Figura 47: Gráfico pruebas rendidas por día

El promedio diario de rendición en el periodo señalado fue de 316 evaluaciones diarias, lo que supera en 200 evaluaciones la capacidad mediante el proceso tradicional. Además, se presentó un máximo de 1060 evaluaciones rendidas el día 23 de agosto del año 2012, es decir, casi 10 veces la capacidad del proceso tradicional.

El promedio de resolución de una evaluación ha pasado de 1 hora 33 minutos a 52 minutos (Figura 48), lo que sumado a que los puntajes obtenidos por los alumnos han mejorado con el paso del tiempo (Figura 49), puede suponer un mayor acostumbramiento y una adopción cada vez más seria de la herramienta.

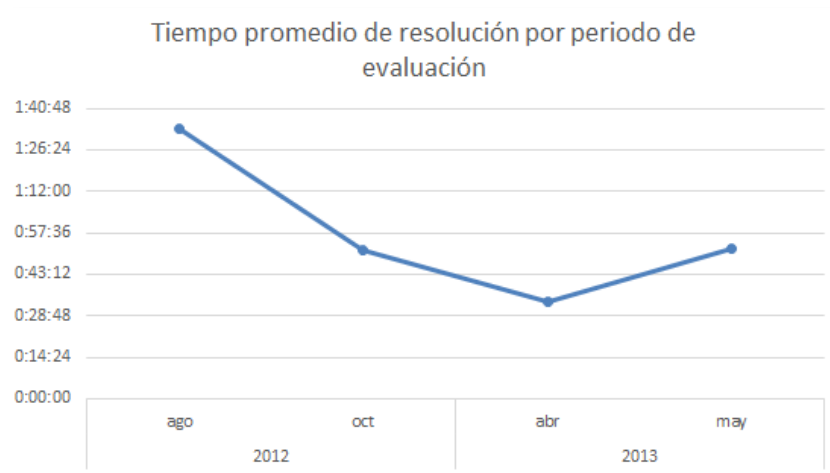


Figura 48: Gráfico tiempo promedio de resolución por período de evaluación

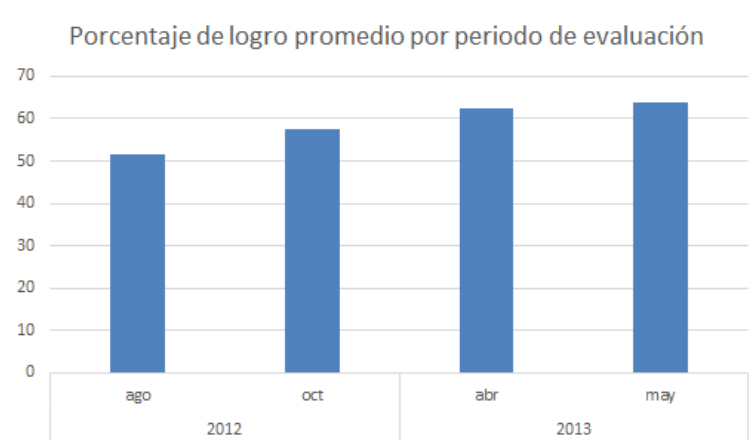


Figura 49: Gráfico porcentaje de logro promedio por período de evaluación

6.2 Eliminación del proceso de elaboración de informes e inmediatez de los resultados

Otro significativo avance ha sido la automatización del proceso de generación de informes, el que antes era realizado de forma manual por la editorial según se explicó en el punto 2.3 y que hubiese sido insuficiente para la cantidad de evaluaciones que la aplicación ha logrado recibir: la máxima capacidad de la editorial para un periodo dado son 80 informes, mientras que los informes automáticos entregados por la aplicación en promedio por periodo han sido 191.

Además, estos informes están a disposición del profesor inmediatamente después de finalizada la evaluación, acortando de manera patente los tiempos de espera respecto al método tradicional, el que demoraba entre 4 a 6 semanas tras rendida la evaluación en enviar el informe impreso a los establecimientos educacionales.

6.3 Análisis automático de los instrumentos de evaluación

El Instituto IDEA está constantemente validando los instrumentos de evaluación analizando cada pregunta según diversos criterios. Este proceso era llevado de forma manual e implicaba un gran trabajo de recopilación de información, digitación y procesamiento, mediante Excel y SPSS. Ahora, a través de la aplicación, estos indicadores son obtenidos de forma automática, dinámica y sobre una muestra mayor y creciente en el tiempo. Además, ya no es necesario recalcular los indicadores, estos se obtienen en el momento considerando todos los resultados presentes en la aplicación.

6.4 Reducción de costos

Recordando lo visto en 2.3, las etapas del proceso tradicional de evaluación son las siguientes:

1. Construcción de material
2. Aplicación presencial
3. Digitalización de resultados y generación de informes

La adopción de la aplicación ha implicado una reducción de costos para la editorial, quien ahora dirige sus esfuerzos sólo a la parte inicial del proceso, la construcción de material, siendo las etapas 2 y 3 manejadas completamente por la aplicación.

6.5 Conclusiones y trabajo a futuro

Los objetivos planteados al inicio del proyecto se han cumplido de forma satisfactoria. La cobertura de los ensayos SIMCE y PSU realizados por el Instituto IDEA y Ediciones SM se ha ampliado de forma importante (según se indica en 6.1); las estadísticas e informes automáticos han podido equipararse al generado de forma manual, con la gran ventaja de que se obtienen de forma inmediata; se ha logrado calcular los indicadores para la validación de los instrumentos de evaluación utilizando los datos generados por la aplicación; y la aplicación se ha construido de forma tal que permite su escalamiento a un número mayor de usuarios.

A lo largo del proyecto fue fundamental la comprensión de los procesos de negocio de la editorial, lo que permitió, especialmente en etapas iniciales, la correcta priorización y dirección del trabajo. Fue así como la primera versión de la aplicación no consideraba gráficos ni reportes, sólo permitía realizar lo medular: que los alumnos rindieran evaluaciones y obtuvieran su puntaje; lo que fue suficiente para partir y dar a conocer el servicio. Producto del mismo conocimiento de los procesos de la editorial es que se detectan puntos donde se podría haber logrado una mejor integración con aplicación, como lo es el caso de la asignación de permisos a los colegios, cuestión muy ligada al proceso de venta.

Una importante lección aprendida fue: un sistema no es un servicio. En el transcurso del proyecto siempre se pensó en la aplicación y su interacción con los usuarios, pero no en la interacción de dichos usuarios como clientes de la editorial, con la editorial misma. En particular y como ejemplo, esto quedó patente cuando la aplicación se puso en marcha: la editorial recibió un gran número de llamadas telefónicas de índole diverso, desde necesidades de soporte hasta solicitudes de

información. Fue así como una capa de servicio no considerada hasta el momento tuvo que añadirse al proceso.

En cuanto a la metodología de desarrollo, si bien se utilizaron buenas prácticas como estandarizar el proceso de versionado de código mediante Git-Flow o utilizar Capistrano como herramienta de automatización para el despliegue de la aplicación en diversos ambientes, un importante punto que no pudo ser abarcado en el presente trabajo fue el testing. Se espera en un futuro considerar este aspecto ligándolo a un servidor de integración continua.

Finalmente, cabe mencionar que para principios del año 2014 nuevas funcionalidades estarán disponibles en la aplicación, entre ellas destacan: Importación de resultados externos para generación de informes, Constructor automático de pruebas para los profesores y Generación de pruebas para rendición offline.

7 Bibliografía

- [1] Moodle Pty Ltd, «http://docs.moodle.org/24/en/Quiz_module,» [En línea].
- [2] ProProfs, «<http://www.proprofs.com/>,» [En línea].
- [3] Ministerio de Educación de Chile, Objetivos Fundamentales y Contenidos Mínimos Obligatorios de la Educación Básica y Media.
- [4] F. Abad, J. Garrido, J. Olea y V. Ponsoda, Introducción a la Psicometría, Teoría Clásica de los Tests y Teoría de la Respuesta al Ítem.
- [5] Cake Software Foundation, Inc., «CakePHP: the rapid development php framework,» [En línea]. Available: <http://cakephp.org/>.
- [6] Amazon Web Services, Inc., «Amazon Web Services,» [En línea]. Available: <http://aws.amazon.com/>.
- [7] jQuery Foundation, «jQuery: The Write Less, Do More, JavaScript Library,» [En línea]. Available: <http://jquery.com/>.
- [8] Highsoft Solutions AS, «Highcharts - Interactive JavaScript charts for your webpage,» [En línea]. Available: <http://www.highcharts.com/>.
- [9] J. Truelsen, «wkhtmltopdf - Convert html to pdf using webkit (qtwebkit),» [En línea]. Available: <https://code.google.com/p/wkhtmltopdf/>.
- [10] F. Knabben, «CKEditor.com | The best web text editor for everyone,» [En línea]. Available: <http://ckeditor.com/>.
- [11] P. Tzonkov, «KCFinder web file manager,» [En línea]. Available: <http://kcfinder.sunhater.com/>.
- [12] PHPExcel, «PHPExcel - OpenXML - Read, Write and Create Excel documents in PHP,» [En línea]. Available: <http://phpexcel.codeplex.com/>.
- [13] M. Fowler, D. Rice, M. Foemmel, E. Hieatt, R. Mee y R. Stafford, Patterns of Enterprise Application Architecture, Addison Wesley, 2002.
- [14] E. Dorrego, «Educación a distancia y evaluación del aprendizaje,» 2006.
- [15] Mozilla Developer Network, «Web API reference | MDN,» [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>.

- [16] D. Wiggins, «XVFB,» [En línea]. Available:
<http://www.x.org/releases/X11R7.6/doc/man/man1/Xvfb.1.xhtml>.