



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

# **SISTEMA DE ADMINISTRACIÓN DE PROGRAMAS Y CURSOS DE EXTENSIÓN PARA EL DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN  
COMPUTACIÓN

**EDUARDO ESTEBAN ESCOBAR UMAÑA**

PROFESOR GUÍA:

SERGIO OCHOA DELORENZI

MIEMBROS DE LA COMISIÓN:

JORGE PÉREZ ROJAS

PABLO BARCELO BAEZA

SANTIAGO DE CHILE

2013

## Resumen

El programa de Educación Continua y Capacitación (PEC) del Departamento de Ciencias de la Computación (DCC) en la actualidad ofrece cursos de extensión, postítulo y diplomados, y un magíster profesional a ingenieros en computación y áreas afines. Estos programas se dictan en horario vespertino y se ofrecen tanto a empresas (en la modalidad de programas cerrados), como a profesionales independientes (en la modalidad de programas abiertos). Más de 200 profesionales pasan cada año por los programas del PEC.

Debido al reconocimiento que el mercado ha hecho a los programas del DCC, la cantidad de cursos y estudiantes se ha incrementado en forma sostenida durante los últimos 5 años. Por lo tanto, el PEC ha pasado a ser un componente importante que no sólo vincula al DCC con la industria, sino que también representa una importante fuente de recursos económicos para el Departamento.

Las operaciones del área se manejan hasta el día de hoy de manera un tanto artesanal, debido a la falta de un sistema de apoyo a sus actividades. Por lo tanto, los procesos del área están propensos a errores, son poco versátiles, además de lentos y caros. Sin embargo, la principal limitante es la imposibilidad de seguir creciendo en forma sostenida debido a la complejidad y volumen de las actuales operaciones del área.

Esta situación claramente no es sostenible en el tiempo, y pone en riesgo los buenos resultados obtenidos hasta el momento. Si bien hoy en día la operación del área se lleva a cabo de manera aceptable, ésta depende en gran medida de la disponibilidad y dedicación del personal involucrado. Esto es algo que claramente el DCC debería querer evitar a fin de asegurar el funcionamiento continuo y la sustentabilidad de esta área de servicios.

Por otra parte, los eventuales problemas en la logística del PEC generan grandes desajustes y afectan la imagen del DCC frente a los estudiantes y las empresas clientes. Por lo anterior, se ha vuelto necesario automatizar algunos de los procesos del área, para que la operación del PEC resulte más sencilla, robusta y eficiente; y al mismo tiempo independiente (en cierta medida) del *know-how* de su personal.

En el presente trabajo de memoria se desarrolló un sistema Web, que permite administrar los programas, cursos, profesores y calendarios de clases. Estas actividades del PEC son las que más limitan el crecimiento del área, por lo que su sistematización debería contribuir a mejorar en forma sustantiva las operaciones del área.

El software desarrollado ha sido evaluado en forma preliminar por el personal del PEC, por su coordinador, y también por un profesor que dicta regularmente clases en dicho programa. Los resultados obtenidos indican que el sistema desarrollado se adapta a la mecánica del PEC, y que facilitará en gran parte la mecánica con la que hoy en día se maneja el área. El próximo paso es realizar una marcha blanca para poder dejarlo en producción durante enero próximo.

## **Agradecimientos**

Principalmente agradecer a mis padres, Agustín y Marcelina, quienes me han apoyado incondicionalmente en mi proceso como estudiante universitario, que siempre han confiado en mi a pesar de todo, y se han esforzado para que cada uno de sus hijos pueda cumplir sus sueños y sean felices.

También quisiera agradecer a mis amigos, compañeros de universidad, colegas de trabajo y todos aquellos quienes me han acompañado en este largo proceso universitario, con todos aquellos que pasamos horas de estudio, y horas programando, a todos aquellos que me aguantaron en mis malos ratos y me apoyaron en momentos complicados para poder seguir adelante en esta ardua tarea, gracias por ese apoyo incondicional.

Y finalmente, y no por eso menos importante, quisiera agradecer a mi profesor guía Sergio Ochoa, quien fue un apoyo fundamental en que mi proceso de memoria se pudiese llevar adelante, que me ayudó y orientó como nadie en este proceso tan complejo.

## Tabla de contenido

1. Introducción.....	1
1.1. Justificación de la Memoria.....	2
1.2. Objetivos de la Memoria.....	3
1.3. Resumen de la Situación Actual del PEC.....	4
2. Requisitos de la Solución.....	6
2.1. Principales Procesos a Apoyar.....	6
2.2. Principales Requisitos de la Solución.....	8
2.2.1. Mantenedores.....	8
2.2.2. Calendarios.....	8
2.2.3. Usuarios Soportados.....	9
2.2.4. Requisitos de Restricción.....	9
2.2.5. Requisitos de Calidad.....	9
2.2.6. Estadísticas.....	10
3. Diseño de la Solución.....	11
3.1. Diseño de la Arquitectura Física.....	11
3.2. Diseño de la Arquitectura del Software.....	12
3.3. Modelo de Datos de la Solución.....	14
3.4. Tipos de Usuarios Soportados.....	16
4. Análisis de Alternativas para la Implementación.....	18
4.1. Tecnologías Básicas.....	18
4.2. Opciones de Calendario.....	18
5. Implementación de la Solución.....	23
5.1. Mapa de Navegación.....	23
5.2. Principales Funcionalidades Según Tipo de Usuario.....	24
5.2.1. Funcionalidad Disponible para el Funcionario PEC.....	24
5.2.2. Funcionalidad Disponible para el Coordinador PEC.....	31
5.2.3. Funcionalidad Disponible para los Profesores.....	32
5.3. Resultados Preliminares.....	33

6. Conclusiones y Trabajo a Futuro.....	35
7. Bibliografía y Referencias.....	38
Anexos .....	39
Anexo A – Script Generador de la Base de Datos en MySQL.....	39

## Índice de Figuras

Figura 1.	Ejemplo de un calendario de clases para un cierto profesor..	05
Figura 2.	Diagrama de Casos de Uso del Sistema.....	06
Figura 3.	Arquitectura del Ambiente Operacional.....	11
Figura 4.	Arquitectura de Software de la Solución.....	12
Figura 5.	Modelo de Datos de la Solución.....	14
Figura 6.	Procesos Básicos realizados por el Funcionario PEC.....	17
Figura 7.	Calendario de muestra de JQuery Event Calendar.....	19
Figura 8.	Calendario de muestra Calendarix.....	20
Figura 9.	Calendario de muestra de FullCalendar.....	20
Figura 10.	Calendario de muestra de MRBS.....	21
Figura 11.	Mapa de Navegación de la Aplicación.....	23
Figura 12.	Vista de login de usuario.....	24
Figura 13.	Vista Mensual del Calendario visto por un Funcionario PEC...	25
Figura 14.	Menú Lateral de la aplicación.....	26
Figura 15.	Vista del Mantenedor de Personas.....	27
Figura 16.	Vista del Mantenedor de Programas.....	27
Figura 17.	Vista del Mantenedor de Cursos.....	28
Figura 18.	Vista del Mantenedor de Dictados.....	28
Figura 19.	Vista del Mantenedor de Salas de Clase.....	29
Figura 20.	Vista Semanal del Calendario vista por un Funcionario PEC..	29
Figura 21.	Vista de edición de una clase.....	30
Figura 22.	Filtros de Calendario por Programa y Profesor.....	30
Figura 23.	Vista de lista de Cursos.....	30
Figura 24.	Vista del Perfil del Estudiante.....	31

Figura 25.	Vista Estadística – Número de Estudiantes por Año.....	31
Figura 26.	Vista Estadística – Hora impartidas por Año.....	32
Figura 27.	Lista de Cursos Dictados - Vista Profesor.....	33
Figura 28.	Lista de Estudiantes de un Curso – Vista Profesor.....	33

## 1. Introducción

El programa de Educación Continua y Capacitación (PEC) del Departamento de Ciencias de la Computación (DCC), perteneciente a la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile fue implementado en el año 1998. A través de este programa el DCC ofrece cursos de extensión, diplomas de postítulo y un magíster profesional. Tanto los cursos como los programas antes mencionados están orientados a profesionales de la industria que desean actualizar sus conocimientos.

El DCC a través del PEC, imparte diplomas de postítulo en cinco áreas diferentes (ingeniería de software, calidad de software, seguridad computacional, gestión informática, desarrollo de proyectos de TI). Cada uno de estos programas poseen múltiples cursos, los cuales tienen a su vez un código, unidades docentes (UD), duración, horarios, profesores y estudiantes asociados, al igual que cualquier otro curso del DCC.

Tanto los programas como los cursos de extensión se ofrecen en un formato cerrado (o exclusivo para una empresa cliente), y también en un formato abierto a profesionales de empresas y organismos públicos. Este canal de servicios le permite al Departamento obtener recursos económicos, y mantener un vínculo con la industria, por lo que es muy importante que funcione bien y que pueda crecer en el futuro.

La administración del PEC se realiza de manera bastante artesanal, especialmente en lo que se refiere a la administración de calendarios de cursos, horarios, asignación de salas de clases, profesores, etc. Esta información se lleva de manera manual, utilizando planillas en papel y en Excel, lo cual lo hace lento y propenso a errores.

Hace algunos años atrás, cuando la actividad del PEC era poca, esta forma de gestión era suficiente. Sin embargo, debido al crecimiento del área se vuelve necesario el poder contar con un sistema de gestión de cursos y programas que no sólo facilite la gestión interna, sino que también ayude a reducir o evitar los errores producidos por el procesamiento manual de esta información. Hoy en día el PEC dicta más de 1000 horas de clases por año, que involucran a más de 200 estudiantes, en más de 40 cursos. Claramente estas dimensiones de operación requieren el apoyo de un sistema automatizado, basado en la Web, dónde no sólo se pueda realizar la operatoria diaria, sino también la obtención de información en línea para apoyar la toma de decisiones del área.

Otra limitación importante del sistema de gestión actual del PEC es la dependencia que éste tiene de las personas que día a día lo ejecutan. El día que alguien se enferma, el proceso se vuelve más vulnerable y propenso a errores, puesto que la información necesaria para realizar las tareas se encuentra distribuida en los computadores del PEC, y la sincronización de ésta se realiza a través de un proceso manual.

Una posibilidad para abordar el problema planteado es modificar una herramienta existente, como por ejemplo U-Cursos. Cabe destacar que U-Cursos al ser una herramienta comercial, su código no está disponible para que se le puedan hacer modificaciones por parte de terceros, por lo que no puede adaptarse fácilmente a la dinámica y a la operatoria del área de Educación Continua. Actualmente se utiliza U-Cursos en cierto grado para el manejo de los cursos, pero al querer obtener informes o información específica para los trámites propios del área, esta herramienta se vuelve limitada, puesto que no fue diseñada para cumplir con ese rol.

Para tratar de paliar el problema planteado, en este trabajo de memoria se desarrolló una aplicación Web que automatiza gran parte de la administración de programas y cursos de extensión del PEC. Este sistema no sólo ordena y facilita la gestión interna del área, sino que también le permite crecer y evolucionar en forma más segura y controlada. Además hace que este proceso se vuelva más independiente de las personas que lo ejecutan diariamente.

### **1.1. Justificación de la Memoria**

Como se mencionó en la sección anterior, esta memoria pretende facilitar la gestión interna del Programa de Educación Continua del DCC mediante el diseño e implementación de un sistema de información. El propósito de este sistema es poder gestionar de manera más sencilla los procesos del área, a través del uso de tecnología.

El producto desarrollado involucró diferentes desafíos, tanto de diseño como de construcción. La aplicación debía contar con una interfaz Web que fuese soportada por múltiples navegadores, además debía ser fácil de entender y de usar. El sistema fue construido con una arquitectura en capas, que le otorga robustez y seguridad, ya que la operatoria diaria del área requiere de estos requisitos.

Por otro lado, la aplicación debía correr en los servidores del DCC (particularmente en Anakena), pues al menos en su versión inicial, cualquier software de gestión interna del PEC debería correr en la intranet del DCC, por motivos de seguridad. Para hacer esto se utilizaron los mecanismos de autenticación de usuarios de dicha intranet, permitiendo el ingreso de usuarios al sistema según un rol (o perfil) asociado a cada uno de ellos (por ejemplo, coordinadores, profesores, personal del PEC, etc.).

Una vez que el usuario ha ingresado al sistema, dependiendo de su rol, se le brinda el acceso a las diferentes herramientas que están disponibles para gestionar los procesos del PEC. Por ejemplo, un profesor puede especificar sus horarios disponibles para hacer clases y ver su calendario de clases, mientras que el personal del PEC tiene las facultades de gestionar cursos, profesores, salas de clases, horarios, etc. Para lo anterior se debe mantener una base de datos para almacenar los distintos usuarios y roles de

cada uno de ellos, con el propósito de otorgar seguridad al sistema. La trazabilidad de todas las operaciones del área es un tema relevante para esta aplicación.

El sistema debe también recomendar salas de clases, profesores y horarios, manejando automáticamente los topes correspondientes. Además la herramienta debe incluir mecanismos de chequeo cruzado para los calendarios de clases generados diariamente. Todo esto facilita a la gestión diaria del PEC, volviéndola más eficiente y efectiva, permitiendo a su personal enfocarse en problemas estratégicos y no sólo en los operativos.

La aplicación debe manejar calendarios con múltiples vistas y filtros (por profesor, fechas, programas, etc.), para que los coordinadores de programas, profesores, estudiantes y el personal del PEC puedan tener amplia visibilidad de los calendarios, según lo que su rol le permita. Se espera que el coordinador del área pueda además usar este sistema para hacer proyecciones de crecimiento y diagnóstico de la misma, mediante análisis de las estadísticas generadas por esta aplicación.

La confiabilidad y disponibilidad del sistema son requisitos críticos, al igual que su usabilidad y mantenibilidad del mismo. Se espera que este sistema le permita al PEC profesionalizarse y crecer de manera armónica.

## **1.2. Objetivos de la Memoria**

El objetivo general de este trabajo de memoria fue diseñar e implementar un sistema de información Web, para apoyar la gestión interna de cursos y programas de extensión que el DCC entrega a través del PEC. Dentro de cada curso hay profesores, estudiantes, salas de clases, programas de clases, etc., que deben ser administrados de manera apropiada para evitar eventuales contratiempos. Dado el actual volumen de operaciones del PEC, las operaciones del área ya no son sostenibles con la solución actual.

Este software desarrollado permite también la comunicación entre los distintos actores involucrados en estos programas: el coordinador del PEC, los coordinadores de programas, los profesores, estudiantes y el personal del PEC.

Los objetivos específicos que se desprenden del objetivo general son los siguientes:

- Crear un modelo de datos integrado, que considerase no sólo el problema a abordar en esta memoria, sino que además le permitiese al PEC evolucionar en el tiempo y crear nuevos servicios de apoyo a su gestión interna según lo requiera en el futuro.
- Asegurar la confiabilidad de los datos y servicios mediante el uso de patrones de seguridad.

- Desarrollar un sistema de información Web que permita solucionar los problemas de gestión que actualmente tiene el PEC. Este software deberá ser seguro, robusto, usable y extensible.

### **1.3. Resumen de la Situación Actual del PEC**

A modo ilustrativo, a continuación se presenta un ejemplo de un trozo del calendario de clases (realizado en Excel por el personal del PEC) para el profesor Sergio Ochoa (Figura 1). Como en cualquier proceso que involucra muchas actividades (en este caso, las clases), hay actividades que se cancelan o deben ser re-agendadas por diversos motivos (por ejemplo, enfermedad del profesor, movilizaciones estudiantiles, solicitudes de la empresa cliente, etc.). Ante un cambio de clases el cronograma completo se ve afectado, puesto que los estudiantes involucrados en estos cursos tienen muy poca holgura de tiempo para afrontar estos cambios. Típicamente los estudiantes de estos cursos toman cursos todos los días de lunes a jueves, e inclusive muchas veces tienen que venir a realizar actividades durante sábados completos. Los días viernes en general no se hacen clases, por solicitud de las empresas cliente y de los estudiantes.

El reordenamiento de clases es una actividad frecuente en el PEC, que sus funcionarios realizan a mano. Debido a que esta actividad se realiza en forma manual, ésta requiere un esfuerzo importante del personal que la lleva a cabo. Dada la experiencia que esta gente tiene realizando dicha labor, el número de errores en los que se incurre es relativamente bajo. Sin embargo, el impacto de estos errores es muy alto. Por ejemplo, si un profesor tiene solapado el dictado de 2 clases de distintos cursos, claramente uno de los dos grupos de estudiantes involucrados asistirá a la Universidad pero no podrá tomar su clase regular.

Mes	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Mayo	21 Feriado	22 Feriado	23	24	25 Feriado	26 Feriado	27
Mayo/Junio	28	28 Gestión de Proyectos PGI 18:30 a 21:30	30	3 Arq. de Software PICS 18:30 a 21:30 hrs	1	2	3
Junio	4	5 Gestión de Proyectos PGI 18:30 a 21:30	6	7 Arq. de Software PICS 18:30 a 21:30 hrs	8	9	10
Junio	11	12 Gestión de Proyectos PGI 18:30 a 21:30	13 Gestión de Proyectos PGI 18:30 a 21:30	14 Arq. de Software PICS 18:30 a 21:30 hrs	15	16	17
Junio	18 Arq. de Software PICS 18:30 a 21:30 hrs	19	20	2 Arq. de Software PICS 18:30 a 21:30 hrs Juan Francisco	22	23	24
Junio/Julio	25 Arq. de Software PICS 18:30 a 21:30 hrs	26	27 Plan Diseño en TI Padan 7 18:00 a 21:00	28 Arq. de Software PICS 18:30 a 21:30 hrs	29	30	1
Julio	2 Feriado	3 Plan Diseño en TI Padan 7 18:00 a 21:00	4 Plan Diseño en TI Padan 7 18:00 a 21:00	5	6	7	8

Figura 1. Ejemplo de un Calendario de Clases para un Cierta Profesor

Esto afecta tremendamente la imagen que el DCC deja ante sus actuales clientes, y puede poner en riesgo la continuidad de esto. Es también por esa razón que la operatoria del área debe ser mejorada considerablemente.

El siguiente capítulo muestra los principales requisitos que están involucrados en la solución. En el capítulo 3 se discuten las herramientas, lenguajes y tecnologías consideradas para implementar la solución actual. En el capítulo 4 se presenta el diseño de la solución. En el capítulo 5 se describe la implementación realizada y una evaluación preliminar de la misma. Finalmente, en el capítulo 6 se presentan las conclusiones y el trabajo a futuro.

## 2. Requisitos de la Solución

A continuación se presentan los principales procesos que la aplicación Web debe apoyar, según el tipo de usuario del cual se trate. Los tipos de usuarios soportados son: *funcionario PEC*, *coordinador PEC*, *profesor* y *consultor*.

### 2.1. Principales Procesos a Apoyar

El sistema de administración involucra un número importante de actividades que están inter-relacionadas, las cuales se describen a continuación.

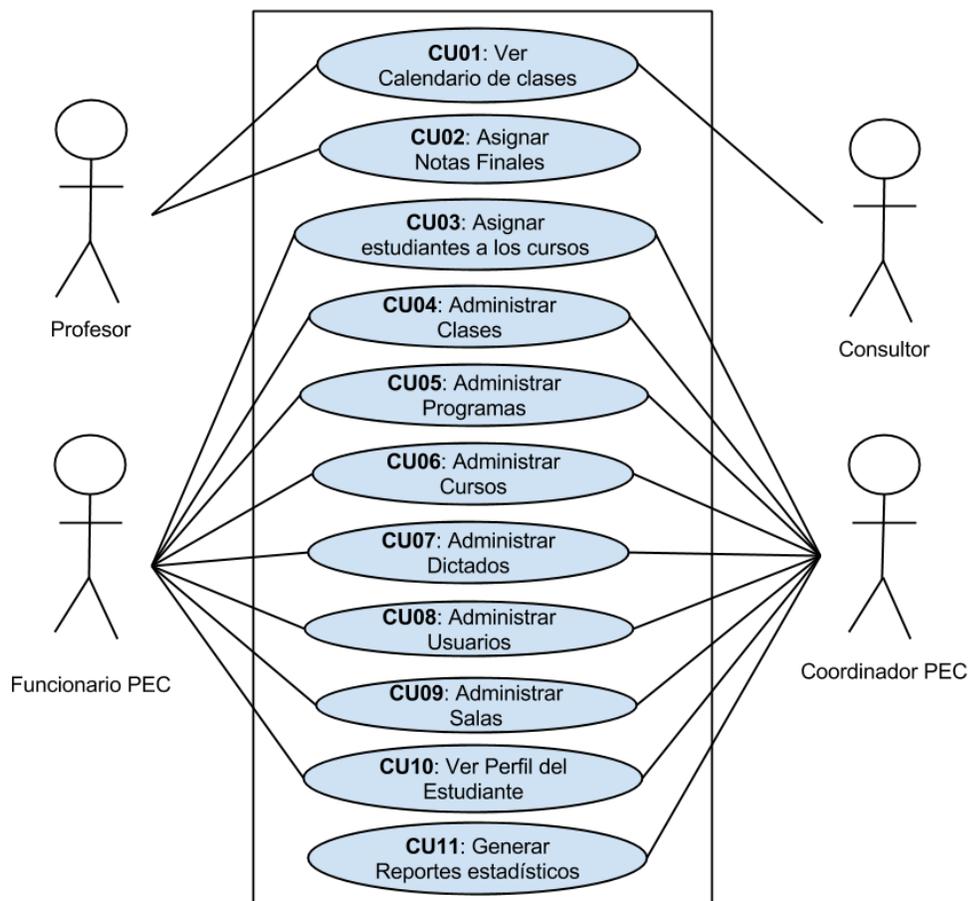


Figura 2. Diagrama de Casos de Uso del Sistema

➤ **CU01: Ver Calendario de Clases.**

**Roles:** Consultor - Profesor

**Caso de Uso:** Una vez logueado en el sistema, el usuario puede ver el cronograma de clases. Si es profesor, entonces verá todos sus cursos y salas designadas; mientras que si es consultor verá los diferentes cursos,

profesores y salas asignadas. El cronograma provisto por el sistema entregará información mínima respecto de la población destino del curso (o sea de los estudiantes involucrados).

➤ **CU02:** *Asignar notas finales de los estudiantes por curso dictado.*

**Roles:** Profesor

**Caso de Uso:** Una vez logueado en el sistema, y accediendo al mantenedor de dictados, el profesor puede ver sus diferentes cursos dictados. En cada curso éste puede asignar una nota final por estudiante.

➤ **CU03:** *Asignar estudiantes a los cursos dictados.*

**Roles:** Funcionario PEC – Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, el usuario puede entrar a la administración de dictados, desde allí debe poder escoger un dictado y poder asignar a los diferentes estudiantes que tomarán el curso.

➤ **CU04:** *Administración del cronograma de clases.*

**Roles:** Funcionario PEC – Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, y accediendo a la vista de calendario, el usuario puede asignar horarios y salas a los diferentes dictados, pudiendo agregar, modificar o eliminar clases.

➤ **CU05/CU06/CU07/CU09:** *Administración de programas, cursos, dictados y salas.*

**Roles:** Funcionario PEC – Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, el usuario debe poder entrar al mantenedor de programas, cursos, dictados y salas, y desde allí puede administrarlos (agregar, modificar o eliminar).

➤ **CU08:** *Administración de Usuarios.*

**Roles:** Funcionario PEC – Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, y accediendo al mantenedor de personas, el usuario puede agregar, modificar y eliminar profesores, consultores y otros funcionarios PEC. Por otra parte, accediendo al mantenedor de dictados, el usuario debe poder asignar los profesores a un dictado en específico.

➤ **CU10:** *Ver Perfil del Estudiante.*

**Roles:** Funcionario PEC – Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, el usuario puede ingresar al perfil del estudiante, donde puede ver su información personal y un resumen de los cursos tomados.

➤ **CU11:** *Generación de reportes estadísticos.*

**Roles:** Coordinador PEC

**Caso de Uso:** Una vez logueado en el sistema, el usuario debe poder acceder a un módulo especial para sacar información estadística directamente desde la base de datos de la aplicación, esto con el fin de apoyar la toma de decisiones en el ámbito del Programa de Educación Continua.

## 2.2. Principales Requisitos de la Solución

A continuación se describe brevemente los principales requisitos que guiaron el diseño de la solución propuesta.

### 2.2.1. Mantenedores

1. Se requiere de un **Mantenedor de Programas**, que permita agregar y editar programas con su nombre y poder asignarle un coordinador.
2. Se requiere de un **Mantenedor de Cursos**, que permita agregar y editar cursos por su código, nombre, número de UD y agregando un programa al cual pertenecen. Además se pueden manejar *Cursos Sinónimos* que posean diferentes códigos, nombres o programas, pero que en realidad se trata del mismo curso. Esto se hace cuando un mismo curso va a ser dictado como parte de diferentes programas cerrados; típicamente para diferentes empresas.
3. Se requiere de un **Mantenedor de Profesores**, que permita agregar o editar profesores con su Rut, nombres, apellidos, correo electrónico, teléfono, dirección y tipo de profesor (part-time o full-time).
4. Se requiere de un **Mantenedor de Estudiantes**, que permita agregar o editar estudiantes con su Rut, nombres, apellidos, correo electrónico, teléfono y (eventualmente) empresa a la cual pertenecen.
5. Al crear un profesor o un estudiante, se debe enviar un correo electrónico al usuario para que éste pueda crear una contraseña para su cuenta.

### 2.2.2. Calendarios

1. Los profesores requieren poder ingresar al sistema e indicar allí los horarios que tienen disponibles para hacer clases.
2. Se debe mantener un pool de profesores, salas de clases disponibles y cursos (asociados a programas).
3. Se debe poder manejar los calendarios para la operación anual del Programa de Educación Continua, con todo lo que ello significa.

4. El sistema debe recomendar calendarios posibles a los usuarios administradores del sistema, según los horarios disponibles de los profesores y la disponibilidad de las salas de clases.
5. El sistema debe por sí mismo mostrar calendarios específicos para profesores y estudiantes, haciendo distinción entre ellos.
6. Al ingresar a la vista del calendario, se deben poder filtrar las actividades ya agendadas, utilizando distintos criterios de filtro. Por ejemplo, se debe poder filtrar por profesor, curso, fechas, programas, etc.

### **2.2.3. Usuarios Soportados**

1. Los perfiles de usuario soportados son los siguientes: *Profesor, Funcionario PEC, Coordinador PEC y Consultores*.
2. El sistema debe ser capaz de manejar un perfil por estudiante, el cual contenga su información personal y sus cursos asociados. Para permitir esto, debe existir una vista específica en el sistema.
3. El sistema debe manejar un histórico de los estudiantes que participan en los programas de diplomado, postítulos, y cursos aislados dictados en el marco del PEC.
4. El sistema debe ser capaz de generar informes de notas por estudiante.

### **2.2.4. Requisitos de Restricción**

1. Aplicación Web que debe quedar instalada en el servidor Anakena, y disponible a través de la Intranet del DCC.
2. El sistema debe soportar múltiples navegadores. Al menos las versiones más recientes de Chrome, FireFox, Opera e IExplorer.

### **2.2.5. Requisitos de Calidad**

1. El sistema debe ser seguro. Para ello debe manejar permisos de ingresos a ciertas áreas de la aplicación, según el perfil que tenga cada usuario.
2. El sistema debe ser flexible permitiendo el uso de diversos tipos de calendarios, e incluso el uso de algunos paquetes de software externos.
3. El sistema debe ser confiable, manejando correctamente la información de los usuarios, clases, horarios y las notas de los estudiantes.
4. Las interfaces del sistema deben ser sencillas e intuitivas para lograr un rápido aprendizaje de la aplicación y alta usabilidad.
5. El sistema debe tener alta disponibilidad.

6. El sistema debe ser extensible, pues la aplicación seguirá creciendo y desarrollándose a futuro.

#### **2.2.6. Estadísticas**

1. Cursos dictados por periodo (es decir, un rango de fechas especificadas por el usuario).
2. Nº de estudiantes por período y curso.
3. Número de cursos abiertos/cerrados dictados en un periodo.
4. Total de horas de clases dictadas por periodo.
5. Número de académicos jornada completa que dictan cursos en el PEC.

### 3. Diseño de la Solución

A continuación se describe el diseño general de la solución. Esto incluye la capa física, la de software y la de datos. Estos tres componentes corresponden a las vistas clásicas involucradas en la especificación de sistemas de información Web.

#### 3.1. Diseño de la Arquitectura Física

A continuación se presenta el diseño arquitectónico de la solución (Figura 3), la cual consta de tres capas: capa de acceso, capa de procesamiento y capa de almacenamiento de datos. Esta separación de "concerns" le infiere flexibilidad al sistema, puesto que cada capa puede evolucionar en forma independiente de las otras. Mientras se mantenga la interfaz de interacción entre las capas, la evolución de una de ellas no debería requerir cambios en el código fuente asociado a las otras capas.

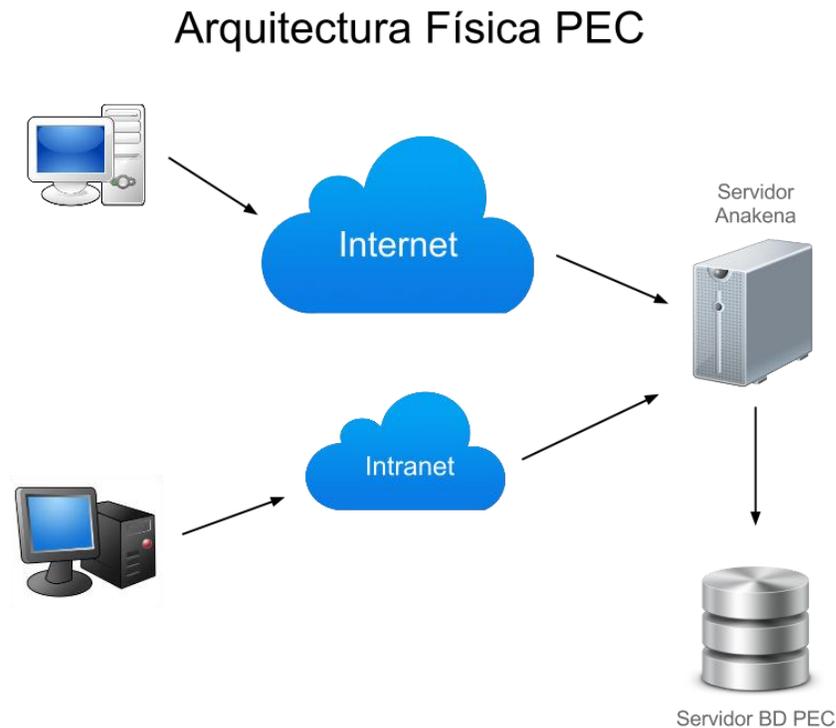


Figura 3. Arquitectura del Ambiente Operacional

Esta arquitectura física (o sea, del ambiente operacional) muestra que existen dos posibles maneras de acceder a la aplicación. La primera forma de acceso es a través de Internet, donde los distintos usuarios podrán acceder siempre que cuenten con un nombre de usuario y contraseña válidos.

La segunda forma de acceso es a través de la Intranet del DCC. A través de esta vía sólo podrán acceder los usuarios administrativos

(Funcionario PEC, Coordinador PEC y Consultores), por lo que el nombre de usuario y clave de éstos, no será requerido a través de la interfaz del sistema, puesto que será consumido directamente desde la sesión de trabajo activa que dicho usuario tiene en la Intranet del DCC. El rol (o perfil de usuario) que el sistema le asignará a cada uno de ellos será el que tienen definido por defecto en el sistema de administración de usuarios de la Intranet. A través de este canal los usuarios administrativos podrán realizar las labores diarias de administración, propias del PEC, así como también obtener informes de estudiantes o estadísticos por demanda.

Se utilizará el servidor Anakena para alojar la aplicación, y esta poseerá su propia base de datos específica, la cual deberá poder satisfacer los requerimientos funcionales antes mencionados.

### 3.2. Diseño de la Arquitectura del Software

La arquitectura de Software (Figura 4) muestra que se utilizará una arquitectura Modelo-Vista-Controlador (MVC); o sea, también se utilizarán tres capas para organizar la funcionalidad implementada en el sistema.

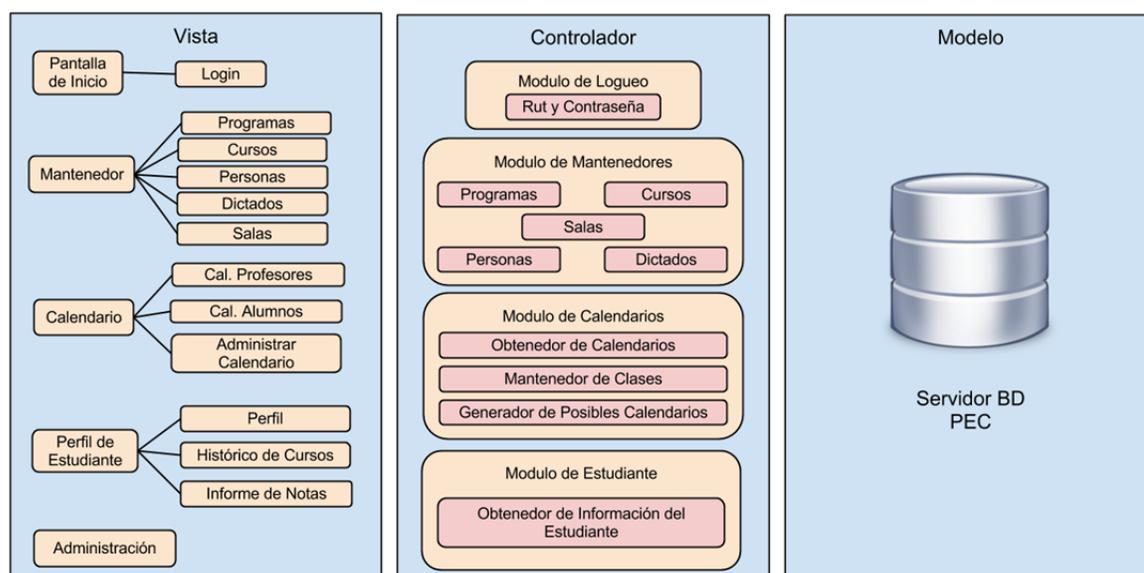


Figura 4. Arquitectura de Software de la Solución

El *Modelo* representa la vista sobre el espacio de datos que brinda la persistencia a la información del sistema. El diseño del espacio de datos se presenta posteriormente en la sección 3.3.

El *Controlador* es típicamente la conexión entre la vista y el modelo. En este caso el Controlador se divide en 4 módulos: *módulo de logueo*, que se encarga de las autenticaciones de los usuarios; *módulo de mantenedores*, que se encarga de las interacciones directas de edición de la base de datos;

*módulo de calendarios*, que se encarga de manipular y hacer la lógica del manejo de calendario; y el *módulo de estudiante*, que se encarga de obtener información de los estudiantes, ya sean informes de notas o de cursos ya cursados por el estudiante. La usualmente llamada "lógica de negocio" se manejará a través del código que vive en esta capa.

Finalmente, tenemos la capa de *Vista*, la cual presenta al usuario las diferentes páginas Web de la aplicación. A través de dichas páginas los usuarios, de acuerdo al rol asignado por el sistema, pueden acceder a la funcionalidad del mismo.

### 3.3. Modelo de Datos de la Solución

La siguiente figura muestra el modelo de datos de la solución. A partir de este modelo de datos se genera la Vista, a través de la cual las capas superiores interactúan con la data.

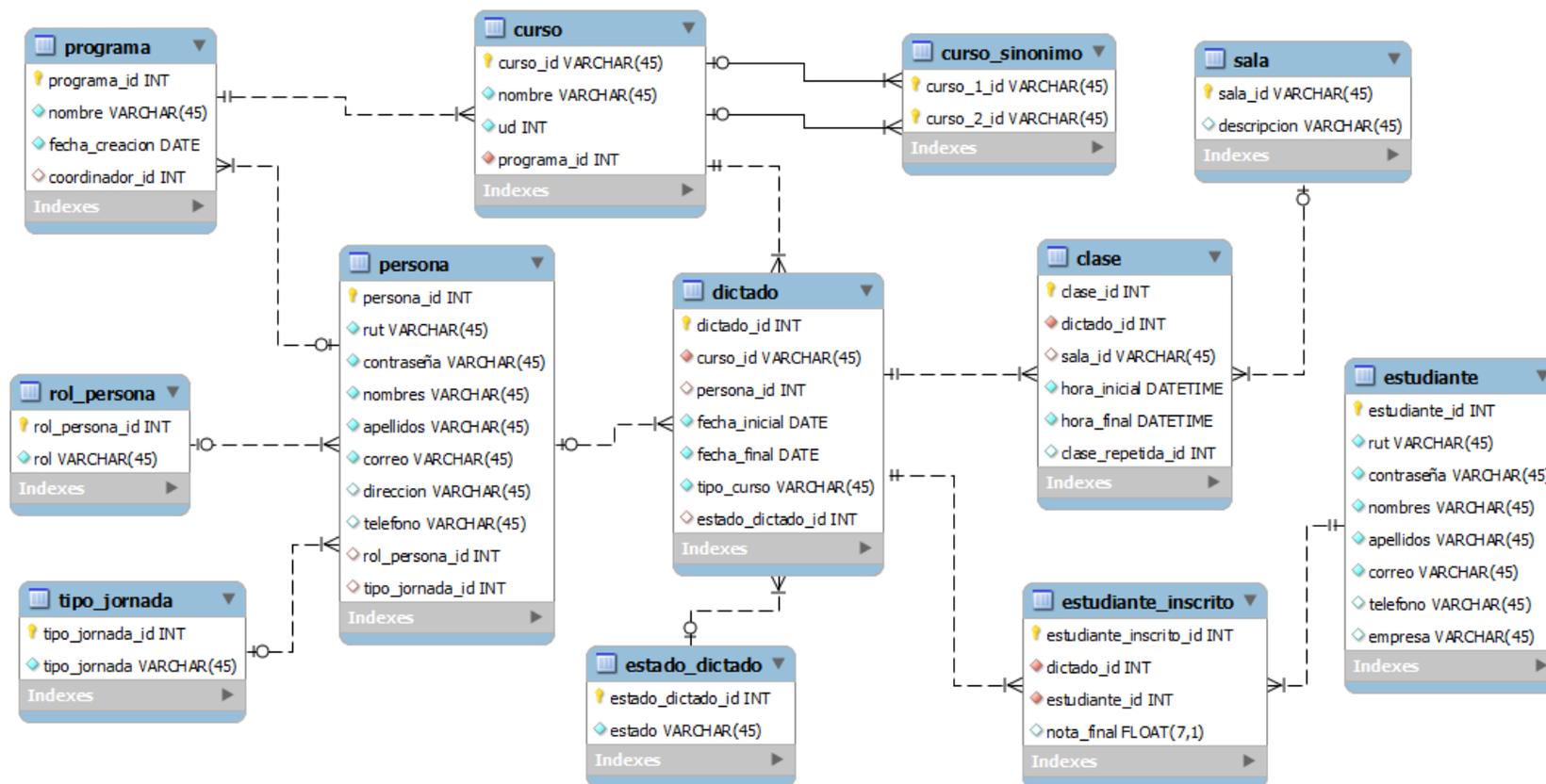


Figura 5. Modelo de Datos de la Solución

A continuación se describen brevemente las tablas que forman parte del modelo de datos presentado en la Figura 5.

- PROGRAMA: Representa a un programa del PEC, incluyendo su nombre, fecha de creación y el profesor que coordina el programa.
- CURSO: Representa a un curso que pertenece a un programa, posee un código (curso\_id), un nombre y el número de unidades docentes (UD).
- CURSO\_SINONIMO: Indica que dos cursos son sinónimos, es decir, cuales cursos son iguales a pesar de tener nombres o códigos diferentes. Esta propiedad es transitiva entre cursos sinónimos.
- PERSONA: Representa a una persona en el sistema, con su información personal.
- ROL\_PERSONA: Representa los roles de las personas en el sistema, que pueden ser: Profesor, Funcionario PEC, Coordinador PEC o Consultores. Estos últimos son los auxiliares del DCC, cuyo rol es organizar los coffee breaks asociados a cada clase.
- TIPO\_JORNADA: Indica si una persona trabaja en jornada part-time o full-time.
- DICTADO: Representa el dictado de un curso, indicando quién es el profesor que dicta el curso, la fecha inicial y final del mismo, el tipo de curso (abierto o cerrado) y el estado del dictado.
- ESTADO\_DICTADO: Indica si un curso aún no comienza, si está en curso, si está finalizado o si está con notas pendientes.
- CLASE: Representa una clase del dictado con su fecha, hora, duración y sala en que se realizará la clase. Si la clase tiene repeticiones se marcan con el mismo id de la primera clase (en clase\_repetida\_id).
- SALA: Representa una sala en el sistema y una descripción de ser necesario.
- ESTUDIANTE: Representa a un estudiante en el sistema con su información personal.
- ESTUDIANTE\_INSCRITO: Representa que un estudiante se inscribió en un dictado de un curso, además de contener la nota final que obtuvo en ese curso.

Se adjunta como anexo (Anexo A) el script generador de la base de datos en MySQL.

### 3.4. Tipos de Usuarios Soportados

A continuación se especifican los tipos de usuario (o roles) que existen en el sistema y las funcionalidades disponibles para cada uno de ellos:

- *Profesor*: el usuario profesor puede acceder al sistema para poder ver su calendario de clases y las salas asignadas para estas. Además tiene permisos de edición sobre las notas finales de cada uno de sus estudiantes.
- *Funcionario PEC*: Este usuario posee permisos para agregar, editar o eliminar personas (usuarios externos del sistema, que acceden a través del canal de Internet), programas, cursos, dictados y salas de clase. Esta funcionalidad está disponible para el *Funcionario del PEC* a través de módulos mantenedores, módulo de administración de clases, y también mediante el calendario de actividades. Por otra parte, este tipo de usuario posee capacidad de ver los perfiles de los estudiantes y entregar reportes de notas de estos.
- *Coordinador PEC*: Este tipo de usuario posee las mismas capacidades que el *Funcionario PEC*, además de poder ver datos estadísticos del área.
- *Consultor*: Son auxiliares, ayudantes, encargados del coffee break (entre otras actividades). Estos usuarios sólo poseen la capacidad de ver el calendario de clases del PEC y sus salas asociadas.

Claramente el tipo de usuario que está más involucrado con el uso del sistema es el Funcionario PEC, ya que su labor diaria va a ser apoyada por este sistema. La Figura 6 muestra los procesos básicos que realiza este usuario. El resto de los tipos de usuario acceder al sistema para realizar consultas o actualizaciones puntuales, pero no procesos formales del PEC.

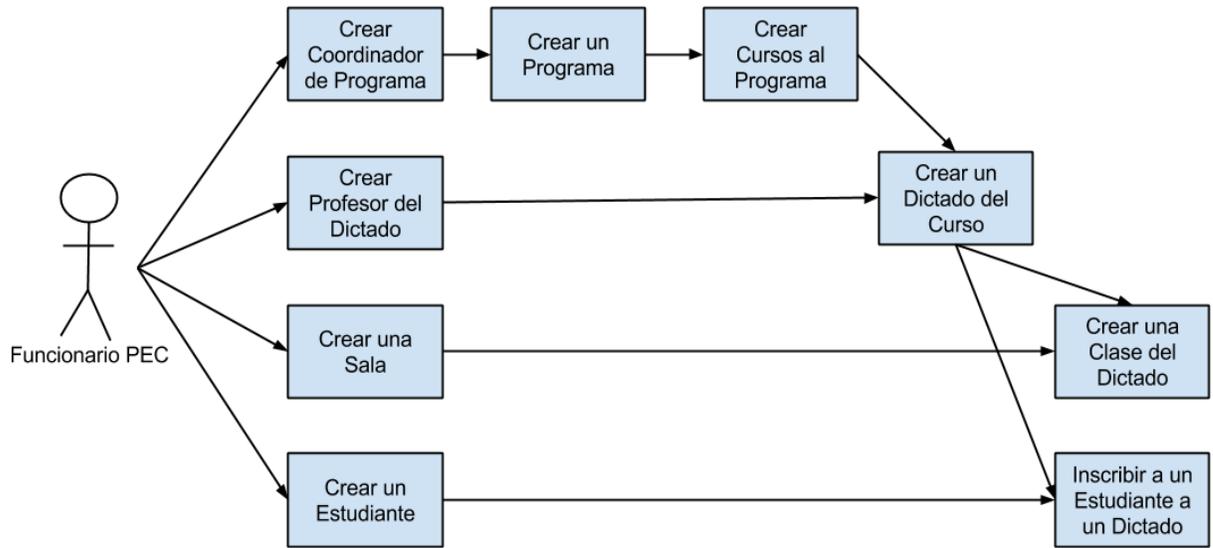


Figura 6. Procesos Básicos realizados por el *Funcionario PEC*

## 4. Análisis de Alternativas para la Implementación

A continuación se especifica brevemente las tecnologías escogidas para implementar el sistema propuesto, y también las opciones de calendario analizadas para visualizar las actividades del PEC.

### 4.1. Tecnologías Básicas

Las tres tecnologías que se mencionan a continuación fueron especialmente escogidas para el desarrollo de la solución desde cero y ad-hoc para el DCC. La principal razón para escoger estas tecnologías fue el dominio que el memorista tiene en el uso de éstas, además de que todas son ampliamente reconocidas y aceptadas en el mercado del desarrollo de software:

**PHP:** Además de las razones previas, se escogió el lenguaje PHP para el desarrollo de la aplicación, por su sencillez y versatilidad; además de ser un lenguaje ampliamente utilizado en aplicaciones Web, por lo que las personas que extiendan la aplicación a futuro no les costará comprender el código fuente de la aplicación. Además, es un lenguaje que posee gran confiabilidad, ya que existe una gran comunidad detrás, que sigue mejorando el motor de PHP, haciéndolo más seguro, robusto y estable.

**CodeIgniter:** Se utilizó el Framework CodeIgniter como complemento a PHP, ya que obliga al desarrollador a organizar el código fuente según el patrón modelo MVC (Modelo-Vista-Controlador). Esto es de gran utilidad para mantener bien separadas la lógica de negocio, de los datos y la presentación. De esa manera, el código queda mucho más limpio y se pueden abordar aspectos de calidad de software de manera mucho más fácil, incluyendo mantenibilidad, flexibilidad y robustez.

**MySQL:** Se utilizó un motor de Base de Datos MySQL, ya que es Open Source, tiene fuerte protección de datos con potentes mecanismos de autenticación. Este motor permite además una buena disponibilidad (uptime) y rendimiento del acceso a los datos. En términos de escalabilidad y flexibilidad, el motor antes mencionado cumple con lo requiere el sistema.

De esa manera están cubiertas todas las capas especificadas en la arquitectura mostrada en la Figura 4.

### 4.2. Opciones de Calendario

En el desarrollo de este sistema se requiere manejar calendarios de actividades, los cuales deben ser muy versátiles, pues gran parte de los servicios que requieren los diversos tipos de usuarios pasan por el uso de estos widgets. Las alternativas para llevar a cabo el uso de calendarios eran: (1) desarrollar una solución desde cero, y (2) utilizar un framework OpenSource que provea ya soluciones básicas para abordar este desafío.

Claramente la segunda opción era la más atractiva, pero requería realizar una búsqueda y evaluación de las opciones potencialmente más útiles.

Luego de un análisis rápido de los pros y contras de cada alternativa, se decidió buscar y evaluar un *software de calendarios* OpenSource, cuyas funciones pudiesen ser incluidas dentro de la aplicación. Esto permitiría que el desarrollo se vuelva mucho más ágil (debido a la reutilización de código) y el producto mucho más confiable, puesto que se estarían reutilizando funciones ya probadas.

Por lo tanto, se procedió a buscar y probar varias alternativas de software de calendarios escritos en PHP. Se evaluó explícitamente la posibilidad de que estos productos ofrecieran servicios para visualizar los eventos en los calendarios, utilizando múltiples colores. Por ejemplo, los colores podrían estar asociados a cursos o programas a los que pertenecen los cursos. El uso de colores también debería poder adaptarse a las necesidades de administración del PEC, y de los profesores. Considerando las restricciones anteriores, las herramientas pre-seleccionadas luego de la etapa de búsqueda, son las que se describen a continuación:

**JQuery Event Calendar Plugin** [5]: Este es un plugin hecho en JQuery, que se integra con la base de datos, y que provee vistas por mes, semana o día. Este plugin permite el uso de varios colores y tiene documentación para la integración de calendarios, por ejemplo para mostrar calendarios agregados por profesor o por programa.

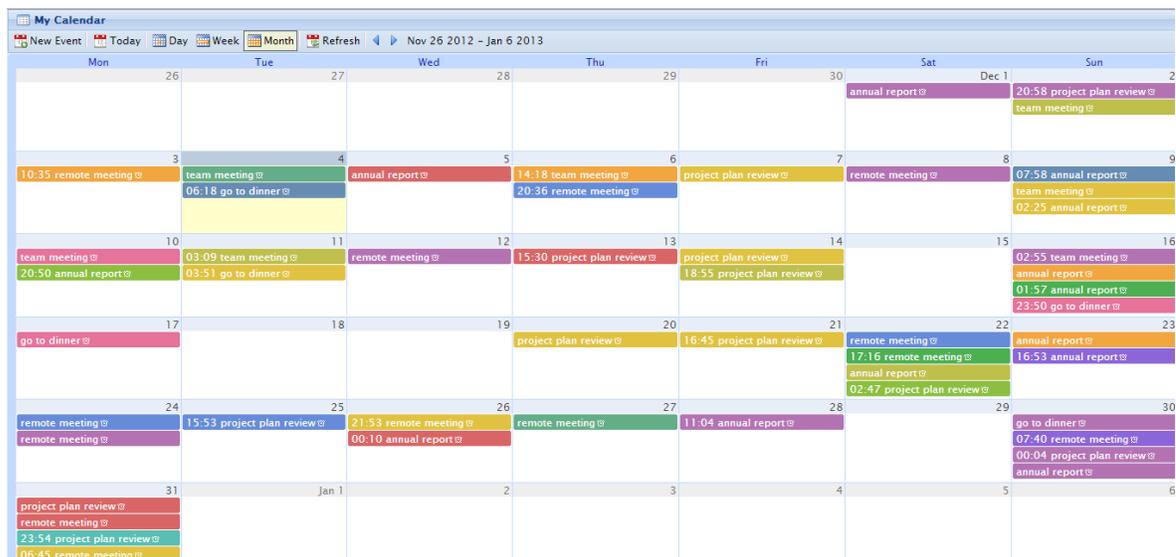


Figura 7. Calendario de Muestra de JQuery Event Calendar

**Calendarix** [6]: Este software de calendario se integra con PHP y MySQL, y posee estilos visuales personalizables. Posee además una versión básica, que se ajusta a lo requerido en estos momentos por el PEC. En la siguiente figura se muestra su interfaz principal, como una alternativa más de las posibles.



Figura 8. Calendario de Muestra Calendarix

**FullCalendar** [8]: Este es un plugin en JQuery, que provee vistas por mes, semana o día, muy similar al *JQueryEventCalendarPlugin*. Posee estilos, capacidades de drag & drop de eventos, e integración de calendarios. Se distribuye además bajo una licencia OpenSource.



Figura 9. Calendario de Muestra de FullCalendar

**Meeting Room Booking System (MRBS)** [7]: Este producto de software es de código abierto y se distribuye bajo una licencia GPL (General Public License), está escrito en PHP y usa un soporte de datos en MySQL/pgsql. Este producto está muy orientado a administrar reservas de salas de reuniones y otros recursos, pero podría adaptarse para ser utilizado con el fin requerido por los usuarios del sistema del PEC.

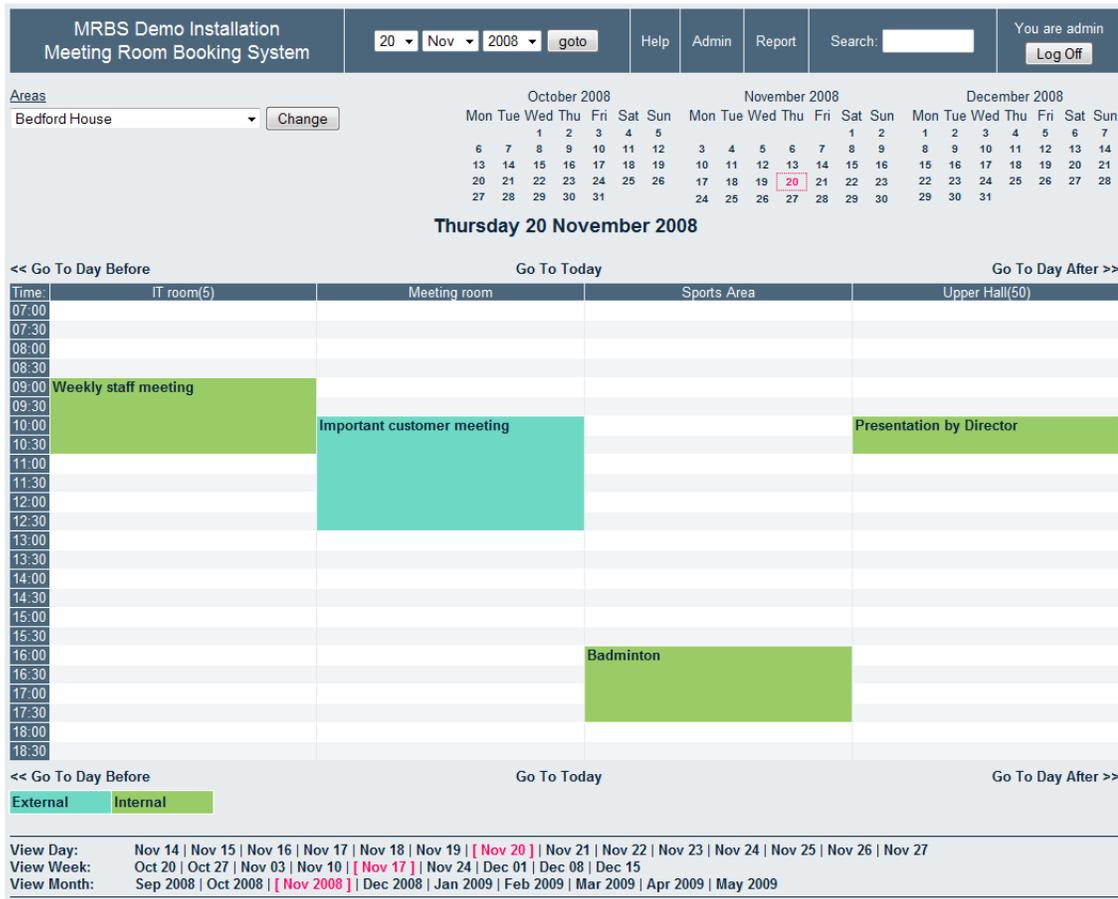


Figura 10. Calendario de Muestra de MRBS

Para evaluar las capacidades de las distintas alternativas presentadas, se crearon eventos sistemáticos (simulando el dictado de clases de un curso), se agruparon estos eventos (simulando el dictado de un programa), se pintaron con diferentes colores y se filtraron para poder generar diversas vistas de las actividades.

Luego de un exhaustivo set de pruebas a los distintos productos de software de calendarios, se decidió utilizar el plugin *JQueryEventCalendar*, pues poseía funciones más avanzadas que *FullCalendar* y *Calendarix*, y la conexión con la base de datos estaba hecha de manera sencilla. Por otro lado, el software MRBS se descartó puesto que las funcionalidades que este ofrecía eran bastante más de lo que el PEC requería para su vista de

calendario. Además, por un tema de tamaño MRBS posee muchas más líneas de código que el resto de las alternativas analizadas, lo cual hace mucho más difícil su adaptación a las necesidades del PEC.

## 5. Implementación de la Solución

A continuación se presentan las principales interfaces de usuario del sistema implementado, las cuales le permiten a los diferentes tipos de usuario poder llevar a cabo sus respectivas actividades. En primer lugar se muestra el mapa de navegación del software, y luego las principales funcionalidades del sistema según el tipo de usuario.

### 5.1. Mapa de Navegación

El modelo general de navegación implementado en la herramienta es el que se muestra en la Figura 11. Allí se puede ver la funcionalidad que está disponible para los distintos tipos de usuario. Particularmente, los módulos en verde son públicos por lo que están disponibles para cualquier tipo de usuario, inclusive para aquellos que acceden en forma externa a través del Internet.

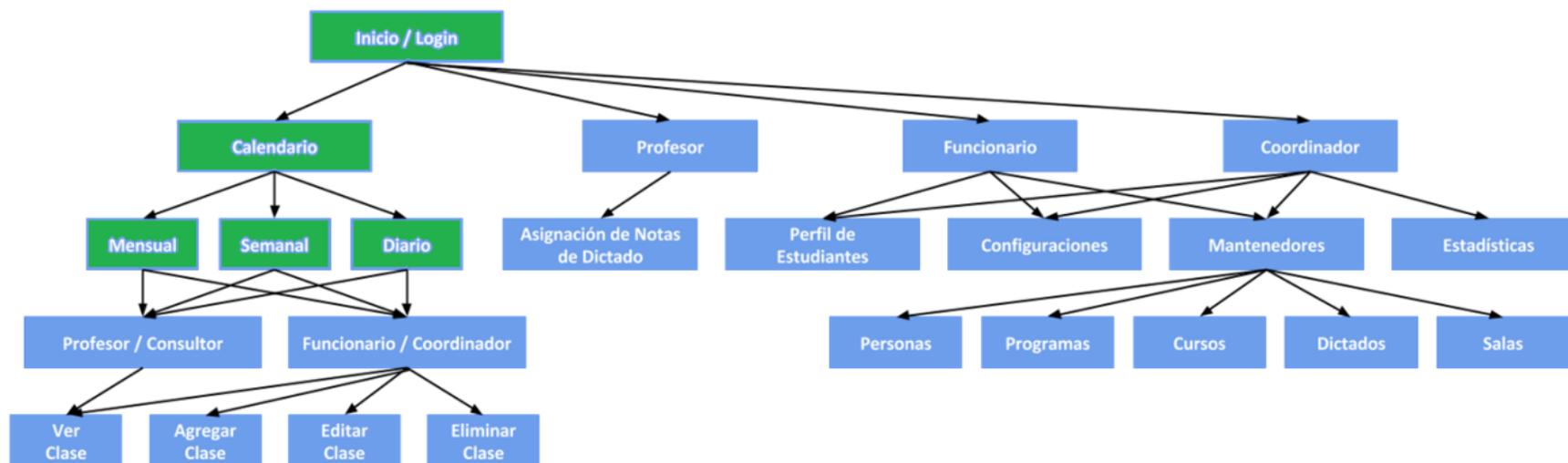


Figura 11. Mapa de Navegación de la Aplicación

## 5.2. Principales Funcionalidades Según Tipo de Usuario

En las siguientes secciones se especifica las principales funcionalidades que el sistema entrega al usuario según su perfil (rol).

### 5.2.1. Funcionalidad Disponible para el Funcionario PEC

En esta sección se explica cómo navegar por las diferentes interfaces del sistema para cumplir el caso de uso **CU4**: "Administración del cronograma de clases" (ver Fig. 2).

En primer lugar el *Funcionario PEC* debe entrar a la página principal de la aplicación, donde debe escribir su e-mail (nombre de usuario) y contraseña para ingresar al sistema (Figura 12). Para poder acceder, el usuario debe previamente haber sido creado por un administrador del sistema (o coordinador PEC). Los datos ingresados por el usuario a través de este formulario, son validados contra la información almacenada en la base de datos del sistema. Este es el principal mecanismo de seguridad de la aplicación. Si bien uno podría pensar que no es gran cosa, hay que considerar que los usuarios actualizadores de datos acceden a través de la Intranet del DCC, por lo tanto ya están claramente identificados, y además pasaron previamente una etapa de autenticación.



Figura 12. Vista de Login de Usuario

La página a la que ingresa el usuario luego de haberse autenticado, es al calendario de clases (Figura 13), donde se puede agregar, modificar o eliminar clases. Sin embargo, para poder llevar a cabo actualizaciones el usuario antes debe ingresar el programa, curso, dictado, profesor y sala correspondiente, según los diferentes mantenedores. A continuación se describe paso a paso el resto de la funcionalidad disponible para este tipo de usuario.

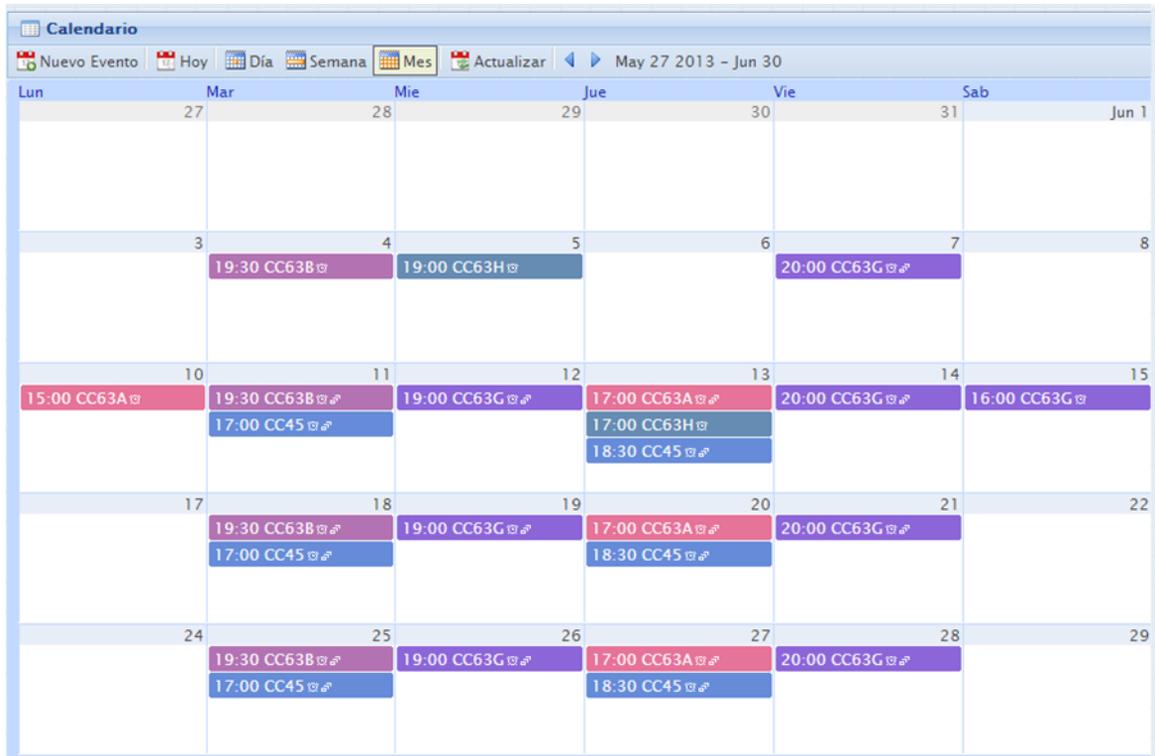


Figura 13. Vista Mensual del Calendario visto por un Funcionario PEC

La aplicación Web cuenta con un menú lateral de la aplicación (Figura 14), que le permite al usuario acceder a diversos módulos del sistema: *calendario*, *mantenedores*, *perfiles de estudiantes*, *estadísticas* y *configuración*. A partir de estas opciones básicas de menú se despliegan varias otras, las cuales permiten acceder al resto de la funcionalidad del sistema. La mayoría de estos módulos está disponible sólo para este tipo de usuario, que claramente es el principal beneficiario de este sistema.

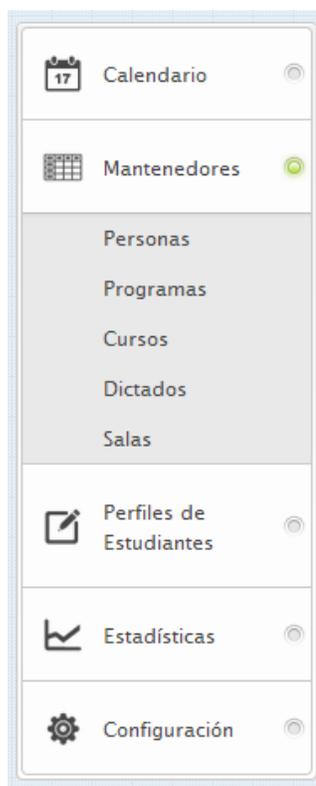


Figura 14. Menú Lateral de la Aplicación

Utilizando el menú lateral de la aplicación (Figura 14), el usuario puede escoger la opción de ir al *Mantenedor de Personas*. En ese caso el sistema despliega un formulario de ingreso de datos (Figura 15), que permite agregar al coordinador del programa (**CU08**). Para ello se debe ingresar el rut, nombres, apellidos, correo, dirección, teléfono, rol de la persona (en este caso, lo usual es que sea "profesor") y su tipo de jornada ("full-time" o "part-time").

La descripción del tipo de jornada del profesor no es demasiado relevante para la operación misma del sistema, pero sí para la obtención de estadísticas por parte de quienes toman decisiones en el PEC. Estos usuarios explícitamente solicitaron poder identificar las actividades de los profesores part-time y full-time, puesto que en el futuro podrían ser manejados, tanto operativamente como presupuestariamente, de manera distinta.

Nueva Persona

Rut \*

Nombres \*

Apellidos \*

Correo \*

Direccion \*

Telefono \*

Rol Persona \*

Tipo Jornada \*

Agregar

Figura 15. Vista del Mantenedor de Personas

Una vez agregado el coordinador del programa, el *Funcionario PEC* debe ingresar los datos básico de dicho Programa desde su mantenedor respectivo (Figura 16). Para ello, el usuario ingresa el nombre del programa, y le asigna un coordinador, de entre la lista de coordinadores existentes en el sistema (**CU05**). Por esa razón es relevante crear al coordinador como primer paso.

Nuevo Programa

Nombre\*

Coordinador\*

Selecciona una opción

Agregar

Figura 16. Vista del Mantenedor de Programas.

Luego de esto, el usuario debe ingresar al mantenedor de Cursos (Figura 17), y debe allí agregar un nuevo curso (**CU06**) al programa recién creado. Cada curso debe tener un nombre, código y número de UD.

Figura 17. Vista del Mantenedor de Cursos

Sólo después de haber creado un programa completo (con todos sus cursos), es posible crear un "dictado". Este dictado corresponde a una instancia específica en la que se dicta cada curso del programa.

Para crear el dictado del curso, el usuario debe ingresar al Mantenedor de Dictados (Figura 18) desde el menú lateral, y agregar un nuevo dictado del curso (**CU07**). Para ello indicará el profesor que estará a cargo de dicho dictado (si el profesor aún no ha sido creado, éste se puede agregar de igual manera que el coordinador), la fecha inicial y la final del mismo, el tipo de curso que será (abierto o cerrado) y su estado (no comenzado, en curso, finalizado o con notas pendientes).

Figura 18. Vista del Mantenedor de Dictados

Por otra parte, es necesario crear salas de clases (**CU09**) que serán utilizadas para el dictado de estos cursos. Para ello el usuario debe ingresar al mantenedor respectivo (Figura 19), donde sólo se debe ingresar el nombre de la sala (por ej.: B101, B212, etc.) y una pequeña descripción si se desea (edificio, ubicación espacial, etc.).

Figura 19. Vista del Mantenedor de Salas de Clase

Finalmente desde la vista de Calendario (Figura 13 y Figura 20), se puede seleccionar un día o un horario donde se creará la nueva clase (**CU04**) que es lo que se deseaba hacer en un comienzo.

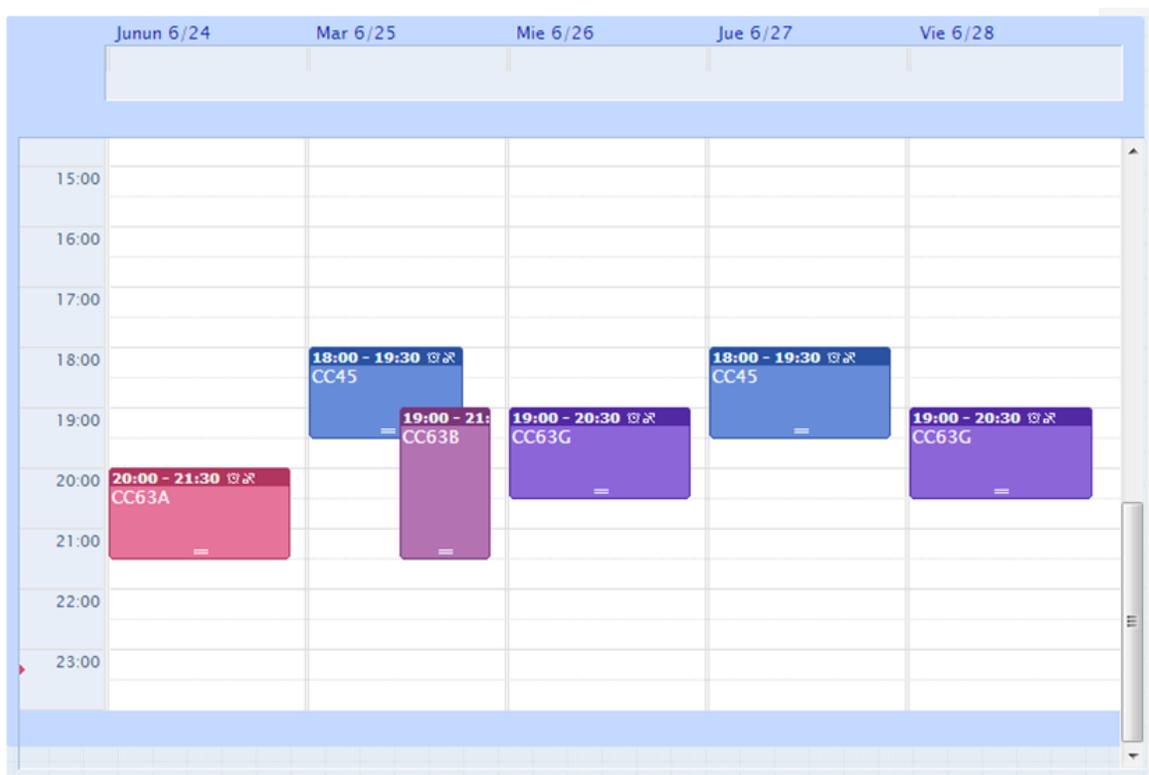


Figura 20. Vista Semanal del Calendario para el Funcionario PEC

Para crear la nueva clase, se abrirá una nueva ventana (Figura 21), donde se deben completar el dictado al cual pertenece la clase, el horario, si se repetirá más veces en ese mismo día de la semana y horario, y la sala respectiva. Si alguna de las clases agregadas tiene tope de horario por profesor o por sala, notificará en la pantalla al usuario. Con esto, finalmente el caso de uso estaría completo.

Figura 21. Vista de Edición de una Clase

Cabe destacar que el calendario posee 3 vistas diferentes, por mes, por día y por año. El acceso a estas vistas dependerá del rol que tenga el usuario. Por ejemplo, los *profesores* sólo ven sus cursos asociados sin permisos de edición, los *coordinadores* y *funcionarios del PEC* ven todos los cursos y poseen permisos de edición, además de poder filtrar los calendarios por programa y profesor (Figura 22), y por último, los *consultores* ven todos los cursos pero no poseen permisos para editar. Por otro lado también se debe destacar que cada mantenedor posee una lista paginada y reordenable, con un buscador integrado para tener fácil acceso a los datos (Figura 23).

Figura 22. Filtros de Calendario por Programa y Profesor

Código Curso	Nombre	Número de UD	Programa	Menú
CC45	Ingeniería de Software II	10	Ingeniería Y Calidad De Software	
CC63A	Liderazgo y Trabajo en Equipo	15	Gestión Informática	
CC63B	Gestión de Proyectos Informáticos	5	Gestión Informática	
CC63G	Introducción a la Ingeniería de Software	5	Ingeniería Y Calidad De Software	
CC63H	Ingeniería de Requisitos	5	Ingeniería De Software	

Mostrando del 1 al 5 de 5 registros

First Previous 1 Next Last

Figura 23. Lista de Cursos

Historial de Eduardo Esteban Escobar Umaña					
Rut	16915398-1				
Nombres	Eduardo Esteban				
Apellidos	Escobar Umaña				
Correo	edescoba@ing.uchile.cl				
Telefono	68746543				
Empresa	Upandoo				

Cursos Dictados Inscritos					
Mostrar	10	registros	Buscar: <input type="text"/>		
Código Curso	Nombre Curso	Fecha Inicial	Fecha Final	Nota Final	
CC45	Ingeniería de Software II	01-06-2013	30-06-2013	5.5	
CC63A	Liderazgo y Trabajo en Equipo	01-05-2013	30-06-2013	4.0	
CC63B	Gestión de Proyectos Informáticos	01-03-2013	14-07-2013	5.0	

Mostrando del 1 al 3 de 3 registros

First Previous 1 Next Last

Figura 24. Vista del Perfil del Estudiante

Los *funcionarios* y *coordinadores* tienen acceso al perfil del estudiante (Figura 24), donde se puede observar la información personal de éstos, además de un resumen de los cursos que ha tomado esa persona (**CU10**).

### 5.2.2. Funcionalidad Disponible para el Coordinador PEC

El *Coordinador PEC* (o simplemente *coordinador*) posee las mismas facultades que el *Funcionario PEC*. Sin embargo este último puede además obtener información estadística, lo cual está especificado en el caso de uso **CU11**: "*Generación de reportes estadísticos*".



Figura 25. Ejemplo de Reporte Estadístico – Número de Estudiantes por Año

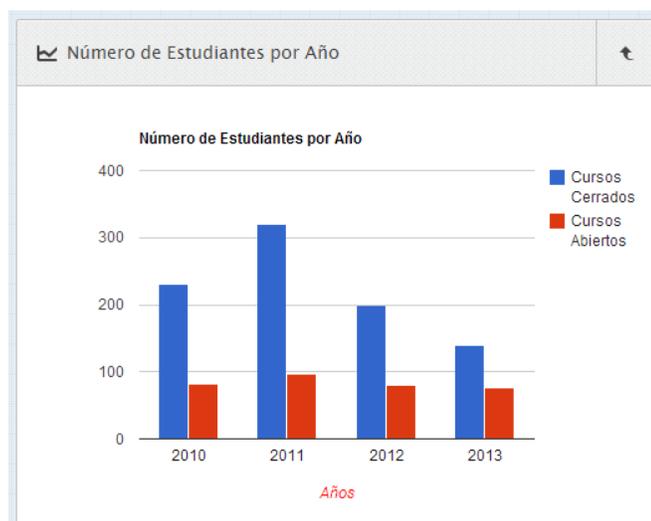


Figura 26. Ejemplo de Reporte Estadístico – Horas de Clase Impartidas por Año

El coordinador PEC, es el único tipo de usuario que tiene acceso a las diferentes estadísticas que se generan con la aplicación, como por ejemplo número de estudiantes por año (Figura 25), cantidad de horas impartidas por año (Figura 26), etc. Es importante destacar que estas estadísticas son parametrizables, pudiéndose solicitar información de (por ejemplo) los programas abiertos, cerrados o de ambos. Hay varios otros tipos de criterios que pueden ser utilizados para filtrar la información estadística a representar en un reporte.

Para poder generar un reporte estadístico el usuario debe ingresar al sistema con su nombre de usuario y clave, y luego dirigirse al área de estadísticas desde el menú lateral. Allí se le despliegan distintas opciones que el usuario podrá escoger para crear el reporte estadístico deseado.

### 5.2.3. Funcionalidad Disponible para los Profesores

A continuación se explica cómo navegar por las diferentes interfaces de usuario para cumplir el caso de uso **CU2**: "Asignar notas finales de los estudiantes por curso dictado."

El Profesor debe ingresar al sistema desde la página principal de la aplicación, y luego dirigirse al mantenedor de dictados ubicado en el menú lateral. Al ingresar verá la lista de dictados en los que él es el profesor (Figura 27) donde al apretar en la lupa podrá ingresar a la lista de estudiantes que tomaron el dictado del curso. Dentro de esa pantalla el profesor es capaz de ingresar las notas finales de cada estudiante al sistema (Figura 28).

ID	Código Curso	Profesor	Fecha Inicial	Fecha Final	Tipo Curso	Estado Dictado	Menú
1	CC63A	Agustín Escobar Arrué	01-05-2013	30-06-2013	Abierto	En curso	
4	CC63H	Agustín Escobar Arrué	24-06-2013	31-08-2013	Abierto	No comenzado	

Figura 27. Lista de Cursos Dictados - Vista Profesor

	Nombres	Apellidos	Correo	Nota Final
1	Eduardo Esteban	Escobar Umaña	edescoba@ing.uchile.cl	5.0
2	Rodrigo Camilo	Marquez Fernandez	rmarquez@dcc.uchile.cl	5.5

Figura 28. Lista de Estudiantes de un Curso – Vista Profesor

### 5.3. Resultados Preliminares

La aplicación Web desarrollada fue evaluada en forma preliminar con dos funcionarias del PEC, el coordinador del área y un Profesor que regularmente dicta clases en el Programa de Educación continua. En primer lugar a cada usuario se le mostró y explicó brevemente las interfaces del sistema que le eran relevante para su labor. Luego se les pidió que indicaran de qué manera realizarían una cierta actividad. Por ejemplo, a las funcionarias del PEC se les asignó como tarea la creación de un curso en el marco de un programa existente, agregándole un nuevo dictado por un profesor, y a ese dictado agregarle clases. Al coordinador del PEC se le pidió la obtención de reportes estadísticos del área, y finalmente, al profesor se le pidió que revisara su calendario de clases para todos los cursos que tenía asignado en los diferentes programas de diplomado y postítulo del DCC.

Los diferentes usuarios al estar familiarizados con la mecánica con la que se maneja el PEC encontraron intuitivo realizar las tareas antes mencionadas con el apoyo del software. Además consideraron que la aplicación es una herramienta que les facilitará enormemente la creación del cronograma de clases, tarea que hoy en día es uno de los trabajos más complejos que deben realizar, debido a la cantidad de restricciones que deben ser consideradas para ejecutar correctamente dicha labor. La creación de estos cronogramas posee además muchas operaciones reiterativas (mucho “copiar y pegar” ya que los calendarios están actualmente implementados en Excel y papel), lo que conlleva a que hayan errores por topes de horarios y de salas. En este sentido, el uso de la herramienta les ayudará a simplificar esta labor y a evitar ese tipo de errores.

Las funcionarias del PEC se mostraron ansiosas por poner en producción la aplicación. Sin embargo nada la trascendencia que tiene el hecho de intervenir las operaciones del área, se ha preferido realizar una marcha blanca antes de poner en producción este sistema. Las funcionarias del PEC entregaron también sugerencias acerca de cómo mejorar el software. Estas observaciones serán evaluadas, y dependiendo de su complejidad serán implementadas antes de la puesta en producción del sistema, o bien quedarán para que a futuro sean implementadas por otras personas. Las observaciones hechas por las funcionarias del PEC fueron las siguientes:

- Poder agregar un campo de comentarios a las clases, para anotar reemplazos por posible ausencia de algún profesor, o bien agregar la empresa para quien se dicta la clase.
- Que al agregar nuevas personas al sistema, la dirección o el teléfono de esa persona no sean campos obligatorios.
- Que el calendario se pudiese conectar con algún sitio oficial (chileno) que publique los feriados de cada año, o bien que ellas puedan marcar los días en que no se dictarán clases, ya sea por feriado o por otras razones.
- Ya que se utiliza U-Cursos para manejar la operatoria de cada curso, ellas desean poder descargar las notas desde ahí, en formato Excel (que actualmente se puede realizar) y subirlas a la aplicación del PEC. Esto les permitiría sacar provecho de los servicios que brinda el nuevo sistema, como por ejemplo el poder sacar certificados de notas en forma rápida, lo cual no es posible en U-Cursos.

Cabe destacar que el coordinador del PEC se mostró sumamente entusiasmado con los servicios provistos por el software, ya que a su juicio simplificarían las actividades del área. Esta persona mencionó que ojalá esta aplicación pueda quedar en producción lo antes posible, y que a futuro pueda seguir creciendo para poder ir entregando más y mejores servicios a los estudiantes y profesores.

Por último, el profesor que participó en dicha revisión también se mostró muy motivado por utilizar el software. Especialmente por la posibilidad de revisar el calendario de clases en línea, sin necesidad tener que solicitarlos por correo, y tener que esperar que dicha solicitud sea satisfecha.

## 6. Conclusiones y Trabajo a Futuro

El Programa de Educación Continua (PEC) del DCC tiene por objeto el dar capacitación a profesionales del área de la computación que desean actualizar sus conocimientos para continuar siendo competitivos en un mercado cada vez más exigente. Para lograr este objetivo el PEC brinda dos tipos de servicios de capacitación. En primer lugar están los cursos aislados, los cuales pueden ser cerrados o abiertos (dependiendo si los participantes pertenecen o no a una misma organización), y dentro de estos últimos se encuentran los fijos y los cursos a pedido. En un segundo grupo se encuentran los Programas de Postítulos y Diplomados, los cuales se comportan como un conjunto de cursos que son tomados por los alumnos. Estos programas pueden también ser tanto abiertos como cerrados.

La logística asociada a la entrega de estos servicios involucra varios procesos que están interrelacionados. La calidad del servicio entregado depende en gran medida de que ninguno de los pasos intermedios falle. En caso de fallas, el impacto puede llegar a ser importante. Por ejemplo, una clase mal agendada podría dar lugar que los alumnos asistan al DCC pero no pueden tomar clases, o que el profesor vaya al DCC pero no los alumnos. Además, varios de los alimentos que se proveen en cada coffee break son perecibles, por lo tanto una clase mal agendada podría dar lugar a un uso ineficiente de los recursos del DCC. Esto es sólo por nombrar a una de las innumerables situaciones a las que se ve expuesto el personal del PEC.

El aumento por la demanda de cursos y de programas de postítulo y diplomados, ha puesto a prueba la logística y las operaciones del PEC. El diagnóstico actual es muy claro: "ya no es posible seguir operando utilizando planillas Excel y reportes impresos". Por esa razón este trabajo de memoria buscó desarrollar una aplicación Web que le facilite la operación diaria al PEC, que automatice parte de sus procesos y le permita al área crecer en forma organizada.

En el informe de memoria antes presentado se describe el trabajo realizado en este sentido. Dicho trabajo abordó todas las etapas del ciclo de vida de un software, menos la implantación y el mantenimiento. Se partió interiorizándose de los procesos con los que trabaja el PEC como una forma de establecer alcances y prioridades en el trabajo a realizar. Luego de eso se procedió a hacer el levantamiento de requisitos entrevistando al jefe del Programa de Educación Continua, al personal de esa área, y al coordinador de un programa que se ofrecen tanto en modalidad cerrada como abierta. Esta última persona es también profesor regular de cursos del PEC.

Con los requisitos ya planteados, se procedió a realizar el diseño preliminar del modelo de datos, donde se tomó en consideración la posible evolución de dicho modelo, dejándolo lo más extensible posible. En forma paralela a dicha actividad se diseñó la arquitectura física y de software del sistema, haciendo énfasis en los requisitos de calidad que una aplicación

como ésta debe poseer, es decir: robustez, seguridad, mantenibilidad y disponibilidad.

El diseño de las interfaces de usuario fue realizado en forma evolutiva, a medida que se realizaba la implementación de los componentes del sistema. En el diseño de estas interfaces se utilizaron plantillas Web y los estilos propios de los diferentes Widgets. En este proceso se pretendió lograr que las diferentes interfaces de usuario fuesen usables. Este requisito de calidad es muy importante dada la variedad de tipos de usuarios que van a acceder al sistema. Más aún, se tiene planeado extender aún más la población de usuarios de este sistema, por lo tanto el requisitos de "usabilidad" toma mayor relevancia.

El sistema implementado puede accederse desde Internet, para usuarios consultores, y desde la Intranet del DCC para el resto de los usuarios. La integración de la aplicación Web a la Intranet del DCC se hace a través de sesiones de trabajo, que son creadas por la Intranet y compartidas con las aplicaciones que forman parte de ésta. La administración de usuarios y la autenticación es manejada también por la intranet. Sin embargo la administración de los roles y los permisos de acceso para los diferentes tipos de usuario es manejada por la aplicación desarrollada en esta memoria.

La aplicación Web desarrollada fue validada en forma preliminar con diferentes tipos usuarios: funcionario PEC, coordinador y profesor. Estas personas debían realizar labores específicas a través de la interfaz Web de la aplicación, según el rol de cada uno de ellos. Los resultados de esta validación preliminar indican que existe un gran interés de parte de los futuros usuarios en la aplicación, pues ayudará enormemente en mantener un mejor orden de la información del PEC, además de poseer información histórica online respecto a los cursos y los estudiantes. Por otro lado, los evaluadores hicieron hincapié en que una de las tareas más complejas del área, que es la creación del cronograma de clases, se podrá simplificar con esta aplicación, además de ayudar a evitar errores y complicaciones por topes de horarios.

El software desarrollado debe pasar primero por una etapa de marcha blanca antes de ser puesto en producción. Se espera que dicho proceso se lleve a cabo al comienzo del semestre otoño 2014, puesto que el actual proceso de dictado de cursos (semestre primavera 2013) está ya en plena ejecución. A partir de Diciembre de 2013 se pretende realizar una marcha blanca, pues ahí finalizan los dictados normales de los cursos del PEC. Además, se realizará la implantación del producto en Enero próximo.

Se espera que este software le permita al PEC agilizar y robustecer las actividades del área, de manera tal que sea factible crecer en forma sostenida y organizada.

Respecto al trabajo futuro sería bueno agregarle al software procesos automáticos, para que el sistema de calendarios propusiese posibles planificaciones en base a los cambios usualmente requeridos a la

planificación inicial. Estos procesos automáticos deberían buscar minimizar los cambios al cronograma de actividades del PEC. De esa manera se podría mejorar aún más el servicio entregado a los estudiantes, y haría a la operación del PEC un poco más independiente del personal que actualmente posee.

Como conclusión final creo significativo el hecho de poder vivir la experiencia de crear una aplicación desde cero, y pasar por todas las etapas de construcción de una solución. Es una experiencia que enriquece a un Ingeniero Civil en Computación, ya que fortalece en forma significativa los conocimientos que fueron adquiridos a lo largo de la carrera.

## 7. Bibliografía y Referencias

1. Fernández, E. B. Security Patterns in Practice: Designing Secure Architectures Using Software Patterns. Wiley Series on Software Design Patterns. 2012.
2. Fowler, M.J. Patterns of Enterprise Application Architecture. Addison-Wesley, 2003.
3. Pressman, R. Ingeniería del Software: Un Enfoque Práctico, (Sexta Edición). McGraw-Hill, 2006.
4. Programa de Educación Continua y Capacitación (PEC). URL: <http://dcc.uchile.cl/pec>. Última visita: Octubre de 2013.
5. JQuery Event Calendar Plugin. URL: <http://www.web-delicious.com/jquery-plugins/#calendar>. Última visita: Octubre de 2013.
6. Calendarix. URL: <http://www.calendarix.com/>. Última visita: Octubre de 2013.
7. FullCalendar. URL: <http://arshaw.com/fullcalendar/>. Última visita: Octubre de 2013.
8. Meeting Room Booking System (MRBS). URL: <http://mrbs.sourceforge.net/>. Última visita Julio de 2013.

## Anexos

### Anexo A – Script Generador de la Base de Datos en MySQL

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,
ALLOW_INVALID_DATES';
```

```
CREATE SCHEMA IF NOT EXISTS `pec_db` DEFAULT CHARACTER SET utf8
COLLATE utf8_unicode_ci ;
USE `pec_db` ;
```

```
-----
-- Table `pec_db`.`rol_persona`
-----
```

```
DROP TABLE IF EXISTS `pec_db`.`rol_persona` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`rol_persona` (
  `rol_persona_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `rol` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`rol_persona_id`) ,
  UNIQUE INDEX `rol_UNIQUE` (`rol` ASC) )
ENGINE = InnoDB;
```

```
-----
-- Table `pec_db`.`tipo_jornada`
-----
```

```
DROP TABLE IF EXISTS `pec_db`.`tipo_jornada` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`tipo_jornada` (
  `tipo_jornada_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `tipo_jornada` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`tipo_jornada_id`) ,
  UNIQUE INDEX `tipo_jornada_UNIQUE` (`tipo_jornada` ASC) )
ENGINE = InnoDB;
```

```
-----
-- Table `pec_db`.`persona`
-----
```

```
DROP TABLE IF EXISTS `pec_db`.`persona` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`persona` (
  `persona_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `rut` VARCHAR(45) NOT NULL ,
  `contraseña` VARCHAR(45) NOT NULL ,
  `nombres` VARCHAR(45) NOT NULL ,
  `apellidos` VARCHAR(45) NOT NULL ,
  `correo` VARCHAR(45) NOT NULL ,
  `direccion` VARCHAR(45) NULL ,
  `telefono` VARCHAR(45) NULL ,
  `rol_persona_id` INT UNSIGNED NULL ,
```

```

`tipo_jornada_id` INT UNSIGNED NULL ,
PRIMARY KEY (`persona_id`),
UNIQUE INDEX `rut_UNIQUE` (`rut` ASC),
UNIQUE INDEX `correo_UNIQUE` (`correo` ASC),
INDEX `fk_Persona_rol_persona1_idx` (`rol_persona_id` ASC),
INDEX `fk_Persona_tipo_jornada1_idx` (`tipo_jornada_id` ASC),
CONSTRAINT `fk_Persona_rol_persona1`
  FOREIGN KEY (`rol_persona_id`)
  REFERENCES `pec_db`.`rol_persona` (`rol_persona_id`)
  ON DELETE SET NULL
  ON UPDATE CASCADE,
CONSTRAINT `fk_Persona_tipo_jornada1`
  FOREIGN KEY (`tipo_jornada_id`)
  REFERENCES `pec_db`.`tipo_jornada` (`tipo_jornada_id`)
  ON DELETE SET NULL
  ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `pec_db`.`programa`

```

```

-----
DROP TABLE IF EXISTS `pec_db`.`programa`;
CREATE TABLE IF NOT EXISTS `pec_db`.`programa` (
  `programa_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(45) NOT NULL ,
  `fecha_creacion` DATE NOT NULL ,
  `coordinador_id` INT UNSIGNED NULL ,
  PRIMARY KEY (`programa_id`),
  INDEX `fk_Programa_Persona_idx` (`coordinador_id` ASC),
  CONSTRAINT `fk_Programa_Persona`
    FOREIGN KEY (`coordinador_id`)
    REFERENCES `pec_db`.`persona` (`persona_id`)
    ON DELETE SET NULL
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `pec_db`.`curso`

```

```

-----
DROP TABLE IF EXISTS `pec_db`.`curso`;
CREATE TABLE IF NOT EXISTS `pec_db`.`curso` (
  `curso_id` VARCHAR(45) NULL ,
  `nombre` VARCHAR(45) NOT NULL ,
  `ud` INT UNSIGNED NOT NULL ,
  `programa_id` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`curso_id`),
  INDEX `fk_curso_programa1_idx` (`programa_id` ASC),
  CONSTRAINT `fk_curso_programa1`

```

```

FOREIGN KEY ( `programa_id` )
REFERENCES `pec_db`.`programa` ( `programa_id` )
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;
-----
-- Table `pec_db`.`curso_sinonimo`
-----
DROP TABLE IF EXISTS `pec_db`.`curso_sinonimo` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`curso_sinonimo` (
  `curso_1_id` VARCHAR(45) NULL ,
  `curso_2_id` VARCHAR(45) NULL ,
  PRIMARY KEY ( `curso_1_id` , `curso_2_id` ) ,
  INDEX `fk_curso_sinonimo_curso2_idx` ( `curso_2_id` ASC) ,
  CONSTRAINT `fk_curso_sinonimo_curso1`
    FOREIGN KEY ( `curso_1_id` )
    REFERENCES `pec_db`.`curso` ( `curso_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_curso_sinonimo_curso2`
    FOREIGN KEY ( `curso_2_id` )
    REFERENCES `pec_db`.`curso` ( `curso_id` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-----
-- Table `pec_db`.`estado_dictado`
-----
DROP TABLE IF EXISTS `pec_db`.`estado_dictado` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`estado_dictado` (
  `estado_dictado_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `estado` VARCHAR(45) NOT NULL ,
  PRIMARY KEY ( `estado_dictado_id` ) ,
  UNIQUE INDEX `estado_UNIQUE` ( `estado` ASC) )
ENGINE = InnoDB;
-----
-- Table `pec_db`.`dictado`
-----
DROP TABLE IF EXISTS `pec_db`.`dictado` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`dictado` (
  `dictado_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `curso_id` VARCHAR(45) NOT NULL ,
  `persona_id` INT UNSIGNED NULL ,
  `fecha_inicial` DATE NOT NULL ,
  `fecha_final` DATE NOT NULL ,
  `tipo_curso` VARCHAR(45) NOT NULL ,

```

```

`estado_dictado_id` INT UNSIGNED NULL ,
PRIMARY KEY (`dictado_id` ) ,
INDEX `fk_dictado_curso1_idx` (`curso_id` ASC) ,
INDEX `fk_dictado_persona1_idx` (`persona_id` ASC) ,
INDEX `fk_dictado_estado_dictado1_idx` (`estado_dictado_id` ASC) ,
CONSTRAINT `fk_dictado_curso1`
  FOREIGN KEY (`curso_id` )
  REFERENCES `pec_db`.`curso` (`curso_id` )
  ON DELETE CASCADE
  ON UPDATE CASCADE,
CONSTRAINT `fk_dictado_persona1`
  FOREIGN KEY (`persona_id` )
  REFERENCES `pec_db`.`persona` (`persona_id` )
  ON DELETE SET NULL
  ON UPDATE CASCADE,
CONSTRAINT `fk_dictado_estado_dictado1`
  FOREIGN KEY (`estado_dictado_id` )
  REFERENCES `pec_db`.`estado_dictado` (`estado_dictado_id` )
  ON DELETE SET NULL
  ON UPDATE CASCADE)
ENGINE = InnoDB;
-----
-- Table `pec_db`.`sala`
-----
DROP TABLE IF EXISTS `pec_db`.`sala` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`sala` (
  `sala_id` VARCHAR(45) NULL ,
  `descripcion` VARCHAR(45) NULL ,
  PRIMARY KEY (`sala_id` ) )
ENGINE = InnoDB;
-----
-- Table `pec_db`.`clase`
-----
DROP TABLE IF EXISTS `pec_db`.`clase` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`clase` (
  `clase_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `dictado_id` INT UNSIGNED NOT NULL ,
  `sala_id` VARCHAR(45) NULL ,
  `hora_inicial` DATETIME NOT NULL ,
  `hora_final` DATETIME NOT NULL ,
  `clase_repetida_id` INT UNSIGNED NULL ,
  PRIMARY KEY (`clase_id` ) ,
  INDEX `fk_clase_dictado1_idx` (`dictado_id` ASC) ,
  INDEX `fk_clase_sala1_idx` (`sala_id` ASC) ,
  CONSTRAINT `fk_clase_dictado1`
    FOREIGN KEY (`dictado_id` )

```

```

REFERENCES `pec_db`.`dictado` (`dictado_id` )
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `fk_clase_sala1`
FOREIGN KEY (`sala_id` )
REFERENCES `pec_db`.`sala` (`sala_id` )
ON DELETE SET NULL
ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `pec_db`.`estudiante`
-----

```

```

DROP TABLE IF EXISTS `pec_db`.`estudiante` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`estudiante` (
  `estudiante_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `rut` VARCHAR(45) NOT NULL ,
  `contraseña` VARCHAR(45) NOT NULL ,
  `nombres` VARCHAR(45) NOT NULL ,
  `apellidos` VARCHAR(45) NOT NULL ,
  `correo` VARCHAR(45) NOT NULL ,
  `telefono` VARCHAR(45) NULL ,
  `empresa` VARCHAR(45) NULL ,
  PRIMARY KEY (`estudiante_id`),
  UNIQUE INDEX `rut_UNIQUE` (`rut` ASC) )
ENGINE = InnoDB;

```

```

-----
-- Table `pec_db`.`estudiante_inscrito`
-----

```

```

DROP TABLE IF EXISTS `pec_db`.`estudiante_inscrito` ;
CREATE TABLE IF NOT EXISTS `pec_db`.`estudiante_inscrito` (
  `estudiante_inscrito_id` INT UNSIGNED NULL AUTO_INCREMENT ,
  `dictado_id` INT UNSIGNED NOT NULL ,
  `estudiante_id` INT UNSIGNED NOT NULL ,
  `nota_final` FLOAT(7,1) UNSIGNED NULL ,
  PRIMARY KEY (`estudiante_inscrito_id`),
  INDEX `fk_estudiante_inscrito_dictado1_idx` (`dictado_id` ASC),
  INDEX `fk_estudiante_inscrito_estudiante1_idx` (`estudiante_id` ASC),
  CONSTRAINT `fk_estudiante_inscrito_dictado1`
  FOREIGN KEY (`dictado_id` )
  REFERENCES `pec_db`.`dictado` (`dictado_id` )
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `fk_estudiante_inscrito_estudiante1`
  FOREIGN KEY (`estudiante_id` )
  REFERENCES `pec_db`.`estudiante` (`estudiante_id` )

```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

```
-----
-- Data for table `pec_db`.`rol_persona`
-----
```

```
START TRANSACTION;
USE `pec_db`;
INSERT INTO `pec_db`.`rol_persona` (`rol_persona_id`, `rol`) VALUES (1,
'Profesor');
INSERT INTO `pec_db`.`rol_persona` (`rol_persona_id`, `rol`) VALUES (2,
'Coordinador PEC');
INSERT INTO `pec_db`.`rol_persona` (`rol_persona_id`, `rol`) VALUES (3,
'Funcionario PEC');
INSERT INTO `pec_db`.`rol_persona` (`rol_persona_id`, `rol`) VALUES (4,
'Consultor');
COMMIT;
```

```
-----
-- Data for table `pec_db`.`tipo_jornada`
-----
```

```
START TRANSACTION;
USE `pec_db`;
INSERT INTO `pec_db`.`tipo_jornada` (`tipo_jornada_id`, `tipo_jornada`)
VALUES (1, 'Full-time');
INSERT INTO `pec_db`.`tipo_jornada` (`tipo_jornada_id`, `tipo_jornada`)
VALUES (2, 'Part-time');
COMMIT;
```

```
-----
-- Data for table `pec_db`.`estado_dictado`
-----
```

```
START TRANSACTION;
USE `pec_db`;
INSERT INTO `pec_db`.`estado_dictado` (`estado_dictado_id`, `estado`)
VALUES (1, 'No comenzado');
INSERT INTO `pec_db`.`estado_dictado` (`estado_dictado_id`, `estado`)
VALUES (2, 'En curso');
INSERT INTO `pec_db`.`estado_dictado` (`estado_dictado_id`, `estado`)
VALUES (3, 'Finalizado');
INSERT INTO `pec_db`.`estado_dictado` (`estado_dictado_id`, `estado`)
VALUES (4, 'Notas Pendientes');
COMMIT;
```