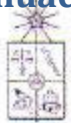


Informe de práctica profesional

CC5901 – Práctica Profesional II

Empresa:	Idea Uno
Alumno:	Matías Pardo Gutiérrez
Rut:	17.670.571-K
Teléfono:	+569 86400863
Email:	mat.pardo@gmail.com

Evaluación por parte de la empresa



Ciencias de la
Computación
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Práctica Profesional

Práctica Profesional (CC4901 – CC5901)

FORMULARIO DE EVALUACIÓN

Datos:

Nombre del Alumno	MATIAS PARDO GUTIERREZ
Empresa	IDEAUNO
Nombre del Supervisor	PABLO RODRIGUEZ NILO
Teléfono del Supervisor	+569 67158126

Fecha de Inicio: 28 / 12 / 2015
Fecha de Término: 29 / 02 / 2016

Evaluación:

Use una nota en la escala del 1 al 7

Satisfacción con el trabajo realizado	7
Calidad técnica	7
Iniciativa e interés	7
Responsabilidad	7
Trato personal y capacidad de adaptación	7
Evaluación del Informe de Práctica Escrito	7

Comentarios Adicionales:

MATIAS muy bien en equipo y es capaz de superar los tareas asignadas.

 **IDEAUNO**
Pablo Rodríguez Niño - Providencia
Idea Uno Ltda. - Julio 2015 2016
www.ideauno.cl
Teléfono: +562 2583 5585
Supervisor

Sumario

Evaluación por parte de la empresa.....	2
1.- Resumen	4
2.- Introducción	5
2.1 .-Lugar de trabajo	5
2.2.- Grupo de trabajo.....	5
2.3.- Visión General del Trabajo.....	5
2.4.- Descripción de equipos y software utilizado	6
2.5.- Situación previa.....	6
2.6.- Descripción general del trabajo realizado	6
3.- Trabajo realizado.....	7
3.1.- Trabajo inicial: Documentos	7
3.2.- Descripción general de programas y archivos del proyecto	7
3.11.- Trabajo de desarrollo	9
3.12.- Descripción del sistema desde el punto de vista del usuario	19
4.- Conclusiones	20
5.- Apéndices.....	21
5.1.- Idea Uno.....	21
5.2.- Vista Principal del Sistema	21
5.3.- Propiedades del proyecto	22
5.4.- Layout de e-mail de Notificaciones.....	22
5.5.- Base de datos	23
5.6.- SmartGit	23
5.7.- Jira	24

1.- Resumen

En este informe se explica el trabajo realizado por el alumno durante su estancia en la empresa IdeaUno.

Se trabajó en el desarrollo de una aplicación Web llamada iScope3, que sería utilizada principalmente en iPhones o Android. Dicha aplicación es para organizar tareas y sub tareas e invitar a otros usuarios a participar de las mismas.

El desarrollo de la aplicación se empieza desde un documento de propuesta hasta llegar a un sistema más o menos finalizado (el proyecto sigue en desarrollo después de que el alumno termina el periodo de práctica).

Durante el desarrollo se refuerza el trabajo en equipo, la utilización de frameworks (cakePHP y framework7) y herramientas de apoyo para el desarrollo (como Jira).

2.- Introducción

2.1 .-Lugar de trabajo

La empresa Idea Uno es una empresa dedicada principalmente a lo que es el desarrollo de sistemas web, ya sea para plataformas de escritorio como móviles, y además de ello cuenta con un área de Streaming. Físicamente cuenta con una gran casona, cuyo primer piso está destinado a una sala comunitaria/reuniones y el área de desarrollo, dónde están todos los desarrolladores y diseñadores gráficos (lo que facilita la comunicación por vías no digitales). El segundo piso son oficinas y área de Streaming.

2.2.- Grupo de trabajo

Los tamaños de grupo de trabajo son diversos, puede ir de una o dos personas a alrededor de cinco, dependiendo de la envergadura del proyecto. Para el Proyecto que trata este informe el grupo de trabajo partió en dos personas, un desarrollador y un jefe de proyecto y el grupo creció hasta tener 5 personas.

2.3.- Visión General del Trabajo

Se desarrolló una aplicación Web que se deseaba utilizar en iOS y que se sintiera como una aplicación nativa (se usaría una aplicación que abriera un navegador con el sistema desarrollado). La aplicación era para crear y organizar tareas (las tareas y sub tareas tenían estructura de árbol) y se podía invitar a usuarios internos y externos a participar de ellas.

2.4.- Descripción de equipos y software utilizado

Acerca de la estación de trabajo:

- Sistema Operativo: Windows 8.1 x64
- Memoria RAM: 8,00 GB
- Procesador: AMD A4-5300 APU with Radeon(tm) HD Graphics 3.40 GHz

Acerca de las aplicaciones:

- Base de datos: MySQL 5.3.6
- Código servidor: PHP 5.3
- Código Navegador: HTML5, CSS3 y javascript
- FrameWork: CakePHP 2.5.3 y Framework7 1.3.5

Herramientas adicionales:

- Editor de Texto: Sublime Text 3
- Gestor Base de Datos: Navicat v11.0.7 - Premium
- Control de Versiones: SmartGit versión 6.5
- Servidor: Vagrant 1.7.2
- Asignación de tareas: Jira

2.5.- Situación previa

La situación previa del proyecto era un documento de propuesta (aún no se había creado el documento de requisitos), y un documento creado por la empresa que contrataba con varias capturas de lo que se deseaba que fueran las pantallas de la aplicación (era una aplicación que sería usada en iPhones, pero no nativa, habría una aplicación que abriría un navegador con la url de la aplicación web).

2.6.- Descripción general del trabajo realizado

Ayudar a generar los documentos de requisitos, diseñar el cómo se construirían varias partes de la aplicación, y realizar el desarrollo de la misma (desarrollo en grupo).

3.- Trabajo realizado

3.1.- Trabajo inicial: Documentos

La aplica a desarrollar es un sistema para gestionar tareas en un celular, con notificaciones, grupos de trabajo, similar a Trello, para utilizar en iPhones (y una versión de escritorio).

Lo primero para comenzar el proyecto se realizaron una serie de reuniones entre el jefe de proyecto y el cliente. Durante esta etapa el alumno leyó y revisó la propuesta inicial del proyecto y las notas de la reunión, y generando un documento de preguntas varias sobre las vistas y flujos. Este documento alcanzó una longitud de cuatro planas el que ayudó bastante durante el desarrollo inicial del proyecto y fue un apoyo durante el resto del desarrollo.

3.2.- Descripción general de programas y archivos del proyecto

El sistema funcionaba sobre una máquina virtual usando Vagrant (facilitando el tener el mismo ambiente de trabajo entre los distintos desarrolladores, e independizándolo de las máquinas dónde se desarrolla).

El jefe de proyecto y el alumno decidieron dividir el trabajo a realizar en módulos.

Los módulos son:

- Sistema Base
- Tareas
- Asignación de usuarios a tareas
- Administración de usuarios (licencia empresa)
- Exportación de tareas y generación de reportes
- Sistema de alertas y notificaciones
- Sistema de pago de la aplicación
- Administración (del sistema)
- Configuración de cuentas de usuario

El alumno trabajó principalmente en el Sistema Base, Tareas, Asignación de usuarios a tareas, Configuración de cuentas de Usuario y Sistema de Alertas y Notificaciones.

Para cuando el alumno terminó su práctica el sistema el sistema poseía 5.962 archivos y 1.577 carpetas (163 MB), que se puede ver en el Apéndice 5.3 Propiedades del proyecto.

3.11.- Trabajo de desarrollo

Para la parte de desarrollo, el alumno ya teniendo los documentos de propuestas y habiéndose aclarado las dudas sobre los flujos, confeccionó la base de datos utilizando conocimientos del ramo de Bases de datos y además conocimientos sobre las convenciones del Framework CakePHP para la creación de tablas (cada tabla tendría correspondencia con un Modelo, CakePHP es un framework que sigue el modelo MVC).

También el alumno buscó Plugins que serían de utilidad durante del desarrollo, como Plugins para manejar imágenes y para generar documentos de Word o PDF.

Paralelamente el jefe de proyecto configuró la máquina Vagrant para su utilización y colocó en ella CakePHP, el framework base de la aplicación, y Framework7, un framework en javascript para crear aplicaciones con animaciones de iOS (para que el usuario use la aplicación como si fuese una aplicación nativa).

A continuación se describirán las tareas principales ejecutadas.

- Creación de Base de Datos

El alumno creó la base de datos como ya se dijo anteriormente, con conocimientos del ramo de Bases de Datos y convenciones del framework.

- Aplicación en Ajax, comenzar con login y creación de usuarios

Para framework 7 todas las páginas debían ser cargadas por Ajax (al presionar un link debía aparecer la página cargada desde la derecha como si fuese una aplicación de iOS). El alumno crea un formulario de registro y un login para la aplicación usando Ajax, utilizando conocimientos de los ramos de Desarrollo de aplicaciones Web (Validaciones de datos) y Seguridad de Software (encriptar contraseñas para guardar en base de datos).

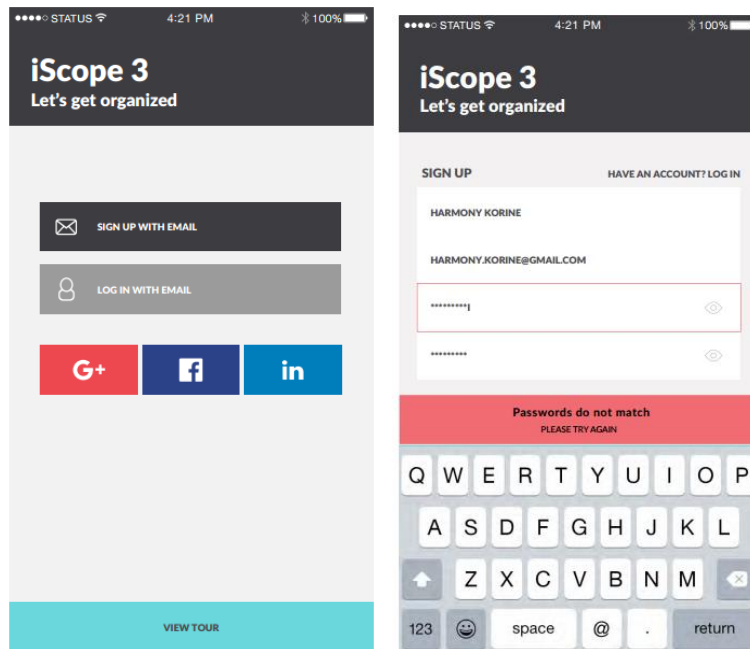


Imagen 1: Formulario de Login y Registro de Usuarios

El formulario de registro además enviaba un mail al usuario que ha creado su cuenta para confirmar la creación y activar la misma.

Finalmente se hace un apartado para configurar la cuenta del usuario con datos y subir un avatar, se crea un sistema para guardar archivos utilizando un Plugin de Cake en que al subir una imagen esta es recortada y guardada en varios tamaños, para usarla según se requiera en la aplicación (y no utilizar imágenes más grandes que consumen recursos al cargar las páginas).

- Creación y diseño de Sistema de Alertas

El alumno diseñó y creó un sistema de Alertas. Varias acciones de la aplicación generarían ciertas alertas para el usuario (como crear una tarea, invitar a una tarea o cambiar porcentaje).

Dichas Acciones generan alertas similares a las notificaciones de Facebook.

La aplicación aún se encontraba en inicio de desarrollo así que las funciones para generar estas Alertas se hicieron de forma genérica, se crearon también tipos de alertas y se crearon entidades llamadas Request, asociadas a las alertas.

Los Request eran para alertas de invitaciones a unirse a una tarea o convertirse en responsable de una tarea, y éstas podían ser respondidas desde la misma vista de Alertas (se podían aceptar, rechazar o delegar a otro usuario, esto último especificando una razón).

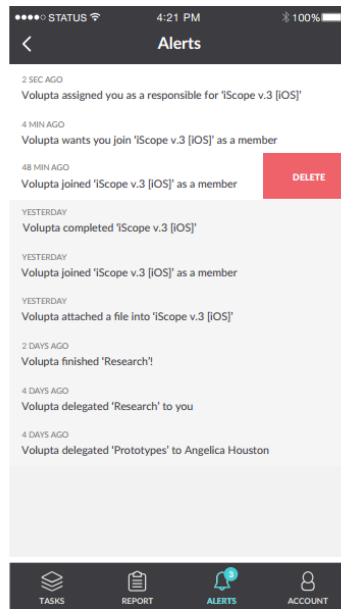


Imagen 2: Pantalla de Alertas

- Invitar usuarios a Tareas

Considerando que el alumno había hecho el sistema de requests y alertas antes descrito le podría facilitar la implementación de la parte de invitación de usuarios a tareas (ya sea para convertirse en responsables de una tarea como miembros de la misma).

Para seleccionar los usuarios a agregar aparecía una pantalla de si se deseaba agregar desde usuarios dentro de la aplicación o invitar a contactos externos, el alumno realizó la invitación interna.

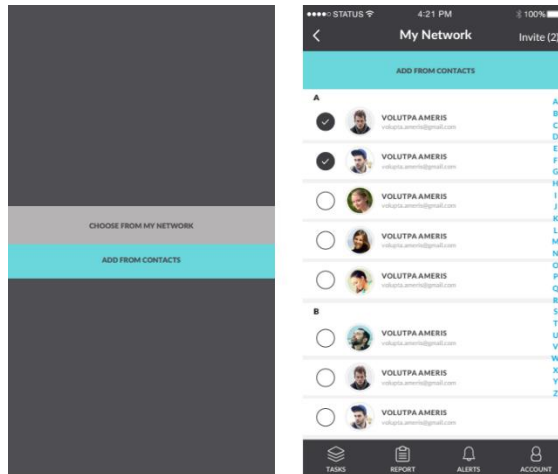


Imagen 3: Selección de contactos de app o externos y lista de selección de usuarios para invitación

- Notificaciones de usuarios (mails)

Asociado a las alertas internas de la aplicación también existían notificaciones por e-mail. Lo interesante aquí para el alumno fue el poner el Layout del e-mail (fue desarrollado por los diseñadores gráficos a pedido del cliente) el cuál utilizaba imágenes. Se debió aprender a enviar las imágenes utilizadas como documentos adjuntos y usadas dentro del e-mail bajo ciertas condiciones (No era solo adjuntar la imagen. En el Apéndice 5.4 Layout de e-mail de notificaciones se puede ver parte de un e-mail.

- Funciones de Tareas

El alumno programa las funciones de las tareas (Crear, Duplicar, Eliminar, Archivar y Cambiar Porcentaje). Utiliza feedback adecuado según lo visto en los Ramos de Taller de Usabilidad y Taller de interacción humano computador.

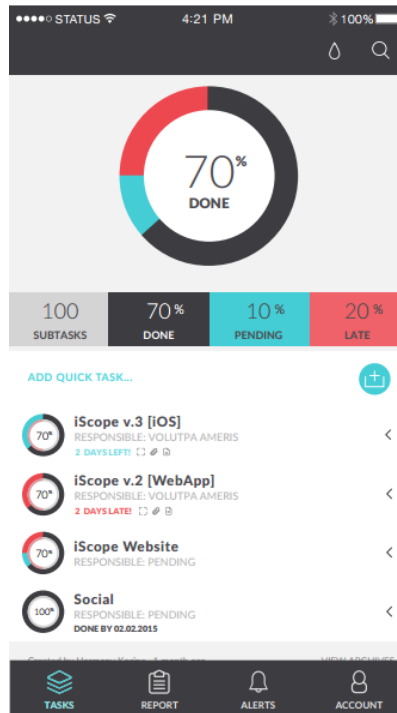


Imagen 4: Vista principal de aplicación, tareas

Lo interesante de este módulo era que acciones cómo cambiar porcentaje. Mirando la imagen cuatro en la parte superior existe un círculo que sirve como resumen de los estados de las tareas de la lista que aparece abajo (la que tiene círculos a su izquierda).

Los colores: negro indica porcentaje de listas, celeste de pendientes y rojas de atrasadas, además en los círculos pequeños hay un círculo interno con gris y rosado, que indican el atraso respecto de la fecha de terminación de una tarea (considerando sus sub tareas, las tareas que aquí aparecen son árboles de sub-tareas).

Como la aplicación debe cargar rápidamente no podían hacerse todos estos cálculos siempre, así que utilizando conocimientos del ramo de Diseño y Análisis de Algoritmos se creó un algoritmo que mantenía calculados correctamente los porcentajes de avance de las tareas en la Base de Datos, y calcular los colores del círculo solo con el número de tareas hijas directas (a un nivel de lejanía) en el árbol.

Así no se desperdiciaba calculando muchas veces esto y se mostraba todo rápido, sólo que al cambiar el porcentaje de una tarea se debía cuidar mantener actualizados los cálculos en la base de datos.

Los círculos fueron adaptados pro el alumno a partir de una librería para graficar en javascript.

- Búsquedas y Filtros

El alumno también diseñó y creó el sistema para buscar y filtrar tareas (el cliente daba cómo quería que se viera, lo que se diseñó fue el cómo).

Como las vistas de resultado eran similares a las listas que aparecen abajo del gran círculo de avance de la tarea (Véase la Imagen 4 ó el apéndice 5.2 de la Vista Principal). Entonces las búsquedas se realizan utilizando css y javascript sobre los datos que ya se encuentran en la vista.

El usuario presiona la lupa, usa los criterios de búsqueda (escribe String que es buscado en nombre de tarea o entre los miembros integrantes de la tarea).

Desaparecen todos los elementos de la vista, se envía un request por Ajax con los datos al servidor y este busca entre las tareas de la vista los elementos que coinciden con la búsqueda y retorna en la respuesta una lista de ids, luego con javascript se muestran entre las tareas visibles los elementos que coinciden y se desaparecen los que no.

Si un elemento que coincidía con la búsqueda pertenece a un árbol en que hay tareas que no coinciden con la búsqueda se cambia la opacidad de los elementos para hacer la diferencia.

Los filtros funcionaban de una forma similar, sólo que previamente se seleccionaban los atributos a filtrar entre las tareas de la vista en una pantalla intermedia.

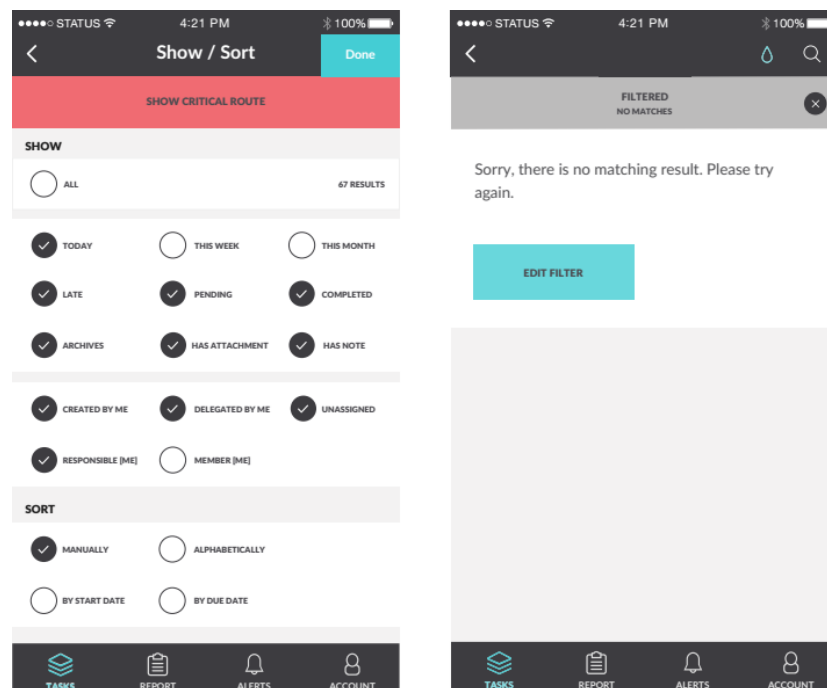


Imagen 5: Atributos de Filtros y Página de resultados (0 resultados)

- Deshacer

El alumno diseña el sistema para deshacer acciones en la aplicación (que como se aprende en los ramos de Usabilidad es una feature importante que da tranquilidad y demuestra robustez en un sistema).

Al realizar una acción en la aplicación como eliminar una tarea o modificar un porcentaje, se guardaba en sesión los elementos que fueron modificados con sus valores originales y se marcaban como inactivos en caso de querer ser eliminados.

Luego se mostraba al usuario la vista y se veía que su acción había sido realizada y aparecía un Pop Up que notificaba al respecto y que se podía tocar para deshacer.

Si el Pop Up se cerraba por tiempo o por presionar la cruz, entonces por Ajax se mandaba un request que confirmaba el cambio, se limpiaba la sesión, si se eliminaban archivos éstos se quitaban del sistema, se borraban alertas y se manejaban los requests asociados.

En caso de presionar deshacer se enviaba otro request Ajax que utilizaba lo guardado en sesión para reestablecer los valores originales y reactivar registros que se volvieron invisibles temporalmente (mientras se esperaba confirmación).

- Otros

Se ha explicado a grandes rasgos lo que hizo el alumno. También durante el desarrollo estaba el trabajo de explicar el sistema a quiénes se iban integrando a trabajar y hablar con ellos para integrar los distintos módulos y funcionalidades que el sistema poseía.

Además cada una de las funcionalidades poseía también detalles que no fueron explicados extensivamente y que se realizaron en su momento (por ejemplo para cambiar porcentaje de una tarea se hicieron reglas y notificaciones, al cambiar el porcentaje de un padre afectaba a los hijos y los listaba en una advertencia, ó cambiar el porcentaje de una tarea desde su View (similar a la pantalla principal del sistema) se hacía girando directamente la zona negra del círculo grande que mostraba el avance.

3.12.- Descripción del sistema desde el punto de vista del usuario

Desde el punto de vista del usuario, el sistema era el login, luego en la pantalla principal divisaba las raíces de sus árboles de tareas, y el progreso de las mismas. Desde aquí podía duplicar, eliminar y crear tareas, y al ingresar a una tarea podía hacer lo mismo con las sub tareas.

También podía invitar a otros usuarios de la aplicación a participar de las tareas o convertirse en responsable (o mandar mails de invitación a personas que no tenían la aplicación).

Las distintas acciones quedaban en los logs de las tareas, y se generaban alertas a los usuarios y notificaciones por mail a quiénes las tenían activadas.

4.- Conclusiones

Durante esta práctica se logró reforzar y afinar lo que es el trabajo en equipo (mediante control de versiones y comunicación entre los integrantes del equipo) respecto a la práctica I.

Se pudo aprovechar también la experiencia previa de la práctica anterior, y los conocimientos adquiridos en los últimos semestres (Diseño y Análisis de Algoritmos, Taller de Usabilidad y Taller de Interfaz Humano computador e Ingeniería de Software, por ejemplo), y practicar aún más conocimientos de cursos más antiguos (Desarrollo de aplicaciones Web y Bases de Datos, por ejemplo).

Finalmente, también fue una instancia de aprendizaje en lo que es el desarrollo de una aplicación destinada a un entorno de producción desde cero, ya que este proyecto no fue tan guiado, aquí el alumno más que sólo recibir tareas para trabajar en un proyecto existente tuvo un rol de diseñador y creador del sistema, ingeniando el cómo desarrollar varios aspectos del software e incluso ayudando a repartir tareas entre los miembros del equipo.

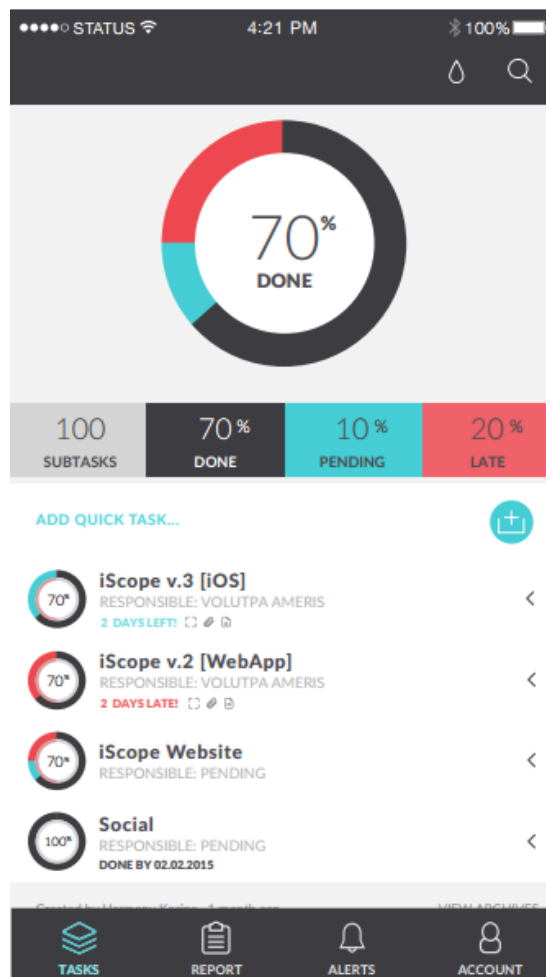
5.- Apéndices

5.1.- Idea Uno

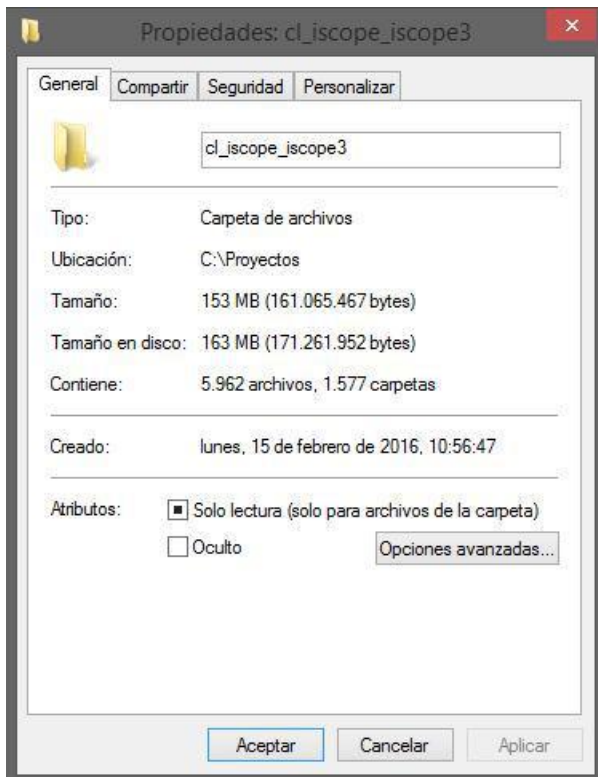
Web de la empresa: **<http://www.ideauno.cl/>**

5.2.- Vista Principal del Sistema

Aquí se muestra la pantalla principal que ve el usuario al loguearse en la aplicación. El círculo más grande en la parte superior representa el avance de todas las tareas de la lista inferior (las raíces). En la lista inferior están las tareas, al hacerles click se pueden ver sus sub tareas.



5.3.- Propiedades del proyecto



5.4.- Layout de e-mail de Notificaciones



Hola Matías:

Te informamos que se ha creado una sub tarea. Para
revisar la tarea presiona el siguiente enlace:

TAREA

Se despide atentamente, el equipo de iScope 3

5.5.- Base de datos

Tablas de la base de datos. El Framework usa modelo MVC y por cada Modelo hay una tabla.

Objects						
Open Table Design Table New Table Delete Table Import Wizard Export Wizard						
Name	Modified Date	Auto In...	Table Type	Data Length	Rows	Comment
alertlogs		1	InnoDB	16 KB	0	
alerts		59	InnoDB	16 KB	52	
alertsettings		37	InnoDB	16 KB	18	
attachments		1	InnoDB	16 KB	0	
avatars		4	InnoDB	16 KB	3	
badges		4	InnoDB	16 KB	3	
companies		2	InnoDB	16 KB	1	
companyrelationships		3	InnoDB	16 KB	2	
contacts		12	InnoDB	16 KB	11	
historictasks		1	InnoDB	16 KB	0	
indexswipeselections		1	InnoDB	16 KB	0	
languages		3	InnoDB	16 KB	2	
licenses		2	InnoDB	16 KB	1	
licensetypes		2	InnoDB	16 KB	1	
memberships		42	InnoDB	16 KB	35	
membershipstatus		3	InnoDB	16 KB	2	
notes		2	InnoDB	16 KB	1	
permissions		1	InnoDB	16 KB	0	
permissiontypes		1	InnoDB	16 KB	0	
proximities		6	InnoDB	16 KB	5	
requestcontact		12	InnoDB	16 KB	6	
requests		17	InnoDB	16 KB	9	
requesttypes		8	InnoDB	16 KB	7	
schema_migrations		42	InnoDB	16 KB	41	
socialconnections		1	InnoDB	16 KB	0	
socialconnectiontypes		1	InnoDB	16 KB	0	
tasklogs		94	InnoDB	16 KB	93	
tasks		22	InnoDB	16 KB	21	
taskselectors		2	InnoDB	16 KB	1	
taskselectortypes		4	InnoDB	16 KB	3	
taskstatus		1	InnoDB	16 KB	0	
urlreports		46	InnoDB	48 KB	55	
userlicenses		2	InnoDB	16 KB	1	
users		38	InnoDB	16 KB	19	

5.6.- SmartGit

The screenshot displays the SmartGit application window. The top menu bar includes options like Repository, Edit, View, Remote, Local, Branch, Query, Changes, Tools, Review, Window, and Help. Below the menu is a toolbar with icons for Pull, Push, Check Out, Add Tag, Add Branch, Merge, Cherry-Pick, Revert, and Reset. The main area is divided into three panes: 1. **Left Pane (Repository Explorer)**: Shows the repository structure with branches (HEAD, develop, master, origin) and tags. 2. **Middle Pane (Commit Log)**: Displays a list of commits with columns for Message, Date, and Author. The selected commit is 'Finish Feature-Click filtro tipo tasks view' by Matias Pardo Gutierrez, dated 02/26/2016 05:46 PM. 3. **Right Pane (File Changes)**: Shows the files modified in the selected commit, including 'views/ctp', 'TaskController.php', 'tasks_view.js', 'tasks_filtro_by_type.js', and 'my-app.js'. The 'Details' tab is active, showing the commit message and a list of files with their modification status and relative paths.

5.7.- Jira

Para repartición de tareas y división de trabajo en Sprints.

The screenshot displays the Jira ISCOPE3 interface. At the top, there's a navigation bar with tabs: Cuadros de mandos, Proyectos, Incidencias, Pizarras, and a prominent orange 'Crear' button. A search bar is on the right. Below the navigation bar, the main header shows 'Pizarra ISCOPE3' and 'Trabajo pendiente'. A sidebar on the left contains icons for various views and a 'FILTROS RÁPIDOS' section with options like 'Open issues', 'Sólo Mis Incidencias', and 'Recientemente Actualizadas'. The main area is a Kanban board titled 'Last week 3' with 25 incidencias. It shows a list of tasks with status icons (green for 'To Do', yellow for 'In Progress', red for 'Done') and priority levels (1, 3, 5). The tasks include: ISCOPE3-133 Vista Ruta Critica para Filtros, ISCOPE3-134 Agregar desde la Rede de Trabajo para Dele, ISCOPE3-135 Vista de Contactos Para agregar desde la Re, ISCOPE3-173 Revisar transiciones y gráficas de recuperar, ISCOPE3-246 Fixear flujo agregar miembros y cuenta> rede, ISCOPE3-210 Notification badge counter, ISCOPE3-228 Alertas - lenguajes y badges, ISCOPE3-229 Inicio de sesión, and ISCOPE3-223 Login - Falta logo - almacenar sesión. On the right, a detailed view for 'cl_iscope_iscope3 / ISCOPE3-134' is shown, including fields for 'Estado' (POR HACER), 'Componente(s)', 'Etiquetas', 'Version(es)', 'Afectadas', 'Versión(es)', 'Correctora(s)', 'Épica', 'Personas' (Informador: Matías Pardo, Responsable: Pablo Rodríguez), and 'Fechas' (Creada: 04/feb/16 4:55 PM, Actualizada: 05/abr/16 2:27 PM).