

Pontificia Universidad Católica de Chile

Escuela de Ingeniería

Departamento de Ciencia de la Computación

|  |  |
| --- | --- |
| **Nombre del proyecto:** | **Administrador de Proyectos de Software** |
| **Organización postulante:** | **Pontificia Universidad Católica de Chile** |
| **Contraparte oficial del proyecto:** | **José Ignacio Benedetto**  **Nicolás Risso** |
| **Dirección:** | **Avenida Libertador Bernardo O’Higgins 340** |
| **Teléfono:** | **2 3542000** |
| **Email:** | **jibenedettoc@gmail.com**  **narisso@uc.cl** |

Versión: 2013.1

Perfil de Proyecto

Formulario de Postulación de Organizaciones Clientes para el Curso IIC2154 Proyecto de Especialidad 2013

Tabla de Contenidos

[1. Información de la Organización Cliente 1](#_Toc308021995)

[1.1 Breve descripción de la organización 1](#_Toc308021996)

[1.2 Productos fabricados o servicios ofrecidos 1](#_Toc308021997)

[1.3 Composición profesional 1](#_Toc308021998)

[2. Información del Problema a Resolver 1](#_Toc308021999)

[3. Información sobre el Contexto del Proyecto 1](#_Toc308022000)

[4. Información de las Contrapartes 2](#_Toc308022001)

[5. Anexos 4](#_Toc308022002)

# Información de la Organización Cliente

## Breve descripción de la organización

La Pontificia Universidad Católica de Chile fue **fundada el 21 de junio de 1888 por iniciativa del Arzobispo de Santiago, Monseñor Mariano Casanova,** con el objetivo de ser una institución que integrara la excelencia académica y una formación inspirada en la doctrina cristiana. Su creación se inserta en las circunstancias históricas universales y locales que afectaron a la Iglesia Católica, instándola a formar profesionales poseedores de un auténtico sentido cristiano de la vida, que sirviera como fermento renovador para las nuevas generaciones.

## Productos fabricados o servicios ofrecidos

n/a

## Composición profesional

n/a

# Información del Problema a Resolver

**2.1. Objetivos**

La gestión de proyectos en la industria tiene la problemática de ser altamente difícil de coordinar debido a la gran gama de aspectos que un administrador debe tener en cuenta durante el desarrollo del proyecto y su posterior liberación. La metodología es diferente para cada tipo de proyecto, usando diferentes herramientas según sus necesidades particulares (proyectos de contrucción suelen usar MS Project, proyectos de computación suelen usar trackers como Redmine, etc.). No obstante es posible identificar puntos de convergencia entre ellos para áreas en particular como trazabilidad y asignación de tareas. Una aplicación basada en la nube puede cubrir a cabalidad estas necesidades. Por otro lado, hay otras áreas en que este enfoque resulta insuficiente. En el caso de reuniones entre cliente y contraparte, lo más importante es facilitar la comunicación y dejar constancia de lo discutido. Para esto una herramienta móvil resulta más apropiada.

La problemática presentada anteriormente abre una oportunidad interesante de explorar: usar una herramienta maestra común para el manejo de estas tareas. La herramienta contará con una serie de módulos que se habilitarán individualmente de acuerdo a las necesidades particulares de cada proyecto, y con un componente móvil y web plenamente integrados.

Podemos conseguir este objetivo extendiendo los conceptos propios de la ingeniería de software, como el desarrollo ágil y disciplinado, a otros tipos de proyectos. A corto plazo, solicitamos un enfoque centrado en las exigencias universitarias pero eventualmente es necesaria la extensión a la industria en general.

Al finalizar, la herramienta deberá facilitar en gran medida la gestión de proyectos permitiendo a un administrador conocer el estado del proyecto y de cada uno de sus integrantes a todo momento, además de una completa trazabilidad de las tareas realizadas. Todo esto debe ser provisto en una interfaz amigable con un alto enfoque en la usabilidad de manera que nuevos grupos de trabajos sean capaces de dominar la herramienta en un tiempo relativamente corto.

**2.2. Requisitos de la solución (web)**

La arquitectura implementada por los alumnos debe estar montada en la web y permitir acceso a los interesados en cada momento. El proyecto combina aspectos de Project Trackers, Task Managers y Kanbanery; no obstante, integra nuevos elementos que permiten un manejo más detallado del proyecto.

El proyecto se divide en dos partes: una plataforma web centrada en la administración de un proyecto de trabajo, y una parte móvil orientada a asistir a analistas en la toma de requisitos, además de proveer un subconjunto limitado de las funcionalidades de la plataforma web. Partiremos describiendo los requisitos de la plataforma web para luego detallar los del componente móvil.

La solución debe permitir inscribir usuarios, los que a su vez podrán crear proyectos. La inscripción debe poder realizarse mediante un formulario personalizado o bien usando datos de redes sociales como una cuenta de Google. Los usuarios además deben poder crear equipos de trabajo donde cada integrante pueda definir un rol con distintos niveles de permisos. Un proyecto solo puede ser accedido por él o los equipos de trabajo que tenga asignados.

El componente web debe definir los siguientes módulos:

1. Task Management
   1. Extensión Software
2. Evaluación
3. Reporting
4. Documentos
5. Calendario

Adicionalmente, debe incluir los siguiente módulos que son específicos para la ingeniería de software:

1. Requisitos
2. QA
3. Issue Tracker

**2.2.1. Task Management**

El módulo de Task Mana Un requisito puede clasificarse como funcional o no funcional. Cada uno de estos debe especificar un template según un subtipo de requisito. Por ejemplo, un requisito funcional puede estar basado en una serie de inputs seguidos de una serie inmediata de outputs esperados; mientras que otro tipo de requisito funcional puede definir un CRON que debe realizar cada cierto intervalo de tiempo.

El módulo de task management debe definir tareas. Cada tarea debe definir un tipo, complejidad y una prioridad. Luego, debe ser posible asignarle esa tarea a uno o más usuarios. Cada tarea debe tener la posibilidad de definir subtareas y dependencias (no se puede hacer esta tarea antes de otra). Cada tarea y subtarea debe dar la posibilidad de hacer comentarios. Asimismo, cada tarea debe poder señalar bloqueos (caso en que una tarea no se puede concretar debido a un imprevisto). Cada tarea debe proveer la capacidad de ingresar estimaciones y de ingresar estado de avance.

Se distinguen los términos de tarea pendiente, tarea en progreso, tarea pausada, tarea completada, tarea en necesidad de revisión y tarea finalizada. Una tarea pendiente corresponde a una tarea que debe ser realizada a futuro. Esta puede ser marcada en progreso por algún integrante del equipo que la tenga asignada cuando empiece a trabajar en ella. El integrante del proyecto debe poder marcar la tarea como completada una vez él estime que la haya realizado a plenitud. Alternativamente, en caso de una redefinición de prioridades o un bloqueo, puede marcarla como pausada. Una vez completada, colegas deben poder criticar la contribución. Una tarea solo puede ser finalizada después de recibir una crítica positiva de parte de otro miembro del equipo. En caso contrario, la tarea pasa a estado en necesidad de revisión donde puede ser retomada eventualmente para volver a tarea en progreso. Eventualmente, una tarea finalizada puede volver a marcarse como pendiente en caso de que esta sea redefinida. Es imperativo que cada vez que haya un cambio de estado vaya quedando un registro de estos en las fechas en que ocurrieron.

Todo lo anterior debe estar disponible en una interfaz lo más fluida posible y amigable para el usuario. Los alumnos deben buscar una solución “drag & drop” o similares y evitar sucesiones de formularios y menús.

**2.2.1.a. Task Management: Extensión a proyectos de software**

Opcionalmente un proyecto puede optar por habilitar una extensión destinada únicamente a proyectos de software. Con esta extensión, los usuarios deben poder asociar código a progresos en el estado de avance. La herramienta de versionamiento solicitada es GIT. El sistema de hosting solicitado es Github. Se les exigirá a los alumnos integrar su solución con el API de Github para encontrar una manera intuitiva de asociar “commits” a actualizaciones de una tarea. Esto permite tener un registro de las contribuciones en líneas de código de cada integrante del equipo.

**2.2.2. Evaluación**

El módulo de evaluación debe definir dos subcomponentes: uno de administración orientada a crear evaluaciones y asignarlas, y otro orientado a los usuarios en general donde podrán rellenarlas. En el componente de administración, el usuario puede definir evaluaciones en base a formularios creados dinámicamente. Luego puede asignárselas a usarios o grupos en particular y definir fechas de entrega. En el otro subcomponente, los usuarios pueden rellenar las evaluaciones con comentarios y/o notas numéricas. Este módulo está orientado a 360 reviewing en la industria y coevaluaciones en el mundo universitario.

**2.2.3. Reporting**

Los usuarios deben tener acceso a una pantalla de vista global que recopile las contribuciones de cada miembro del equipo a un proyecto en particular. Se debe incluir como información al menos: número de tareas realizadas, horas dedicadas y horas estimadas (opcionalmente se pueden definir parámetros adicionales para proyectos de software, como por ejemplo líneas de código contribuidas). Se deberá filtrar esta información por fecha u otros parámetros y deberán proveer medios amigables para presentarla (gráficos, tablas, etc…). Se espera que los alumnos contribuyan con nuevos parámetros que estimen son relevantes para estos efectos.

**2.2.4. Documentos**

La solución debe servir también como un medio de distribución de archivos. Debe ser posible subir archivos y permitir su descarga para el resto del equipo, o bien miembros en específico. Una tarea debe poder definir un enlace a un archivo en particular que pueda ser relevante. Se les pedirá a los alumnos también que busquen alternativas de integración con herramientas como Dropbox para facilitar la usabilidad del sistema.

**2.2.5. Calendario**

Se debe incluir un módulo calendario donde sea visible por día, semana o mes la actividad de cada miembro del equipo, o del grupo en general. En dicho módulo debe poder añadirse fechas de entrega que definan entregables. Cada entregable corresponde a un conjunto de tareas. En el calendario debe salir señalado explícitamente el estado de avance de dicho entregable. Estos entregables pueden ser grupales o específicos para una persona.

El calendario deberá también poder definir eventos de distinta índole (como por ejemplo, reuniones). Los usuarios podrán configurar el recibir una notificación momentos antes del inicio del evento. Los alumnos deberán también evaluar la posibilidad de integrarse con Google Calendar.

**2.2.6. Requisitos**

Específicamente para proyectos de software, se debe poder definir un módulo de requisitos. Un requisito puede clasificarse como funcional o no funcional. Cada uno de estos debe especificar un template según un subtipo de requisito. Por ejemplo, un requisito funcional puede estar basado en una serie de inputs seguidos de una serie inmediata de outputs esperados; mientras que otro tipo de requisito funcional puede definir un CRON que debe realizar cada cierto intervalo de tiempo. Cada requisito funcional debe poder tener asociadas una o más tareas.

**2.2.7. Quality Assurance**

La solución debe incluir un módulo de QA orientado a proyectos de software en donde puedan registrarse los tests que incorpora el proyecto. Estos deben estar clasificados en test unitario, test de integración, test de usabilidad y test de caso de uso. Cada test debe poder asociarse a una tarea en una determinada fecha y debe indicar un resultado (test exitoso o fallado). Esto le permitiría a un administrador tener una evidencia explícita de que cada tarea cumple satisfactoriamente con lo esperado. Cada test debe indicar la fecha en que se realizó y cada tarea debe mostrar en forma explícita si fueron testeadas o no. En caso de redefinición de tarea, debe indicar explícitamente que requiere nuevos tests que respalden la nueva especificación.

**2.2.8. Issue Tracker**

La solución debe incluir un módulo que permita reportar bugs. Los alumnos deberán encontrar la manera más cómoda de que estos se hagan visibles al resto del equipo, que los desarrolladores puedan asignárselos y luego marcarlos como resueltos.

Los usuarios deben tener una pantalla personalizada que resuma toda la información relevante para ellos. En particular, deben tener una bandeja de entrada que reúna mensajes destinados para ellos y que pueda ser redirigida a voluntad a un email, filtrando por diversos aspectos. En la bandeja de entrada deben poder llegar: avisos de tarea asignada, comentarios a tareas que tenga asignadas, mensajes de compañeros de equipo, avisos del administrador. Los usuarios deben poder también suscribirse a una tarea que sea de su interés para recibir avisos de ella también.

Todo lo anterior debe estar cubierto por un sistema de búsqueda.

**2.3. Requisitos de la solución (móvil)**

Pasando a la plataforma móvil, se espera que los alumnos realicen una aplicación en Android de forma nativa. Deben orientar su aplicación a la plataforma Galaxy Note en sus distintas versiones y pueden basarse a lo más en el API Level 17 como versión mínima de Android.

Esta aplicación debe poder integrarse plenamente con la solución web para lo cual debe definir un mecanismo de login que permita identificar al usuario.

El componente móvil debe definir los siguientes módulos:

1. Reunión
2. Tareas

**2.3.1. Reunión**

La aplicación móvil debe definir un entorno de reunión que permita definir salas por un host a la cual interesados o invitados puedan unirse desde sus propios dispositivos. La reunión incluye una herramienta de diagramación que le da a los usuarios la posibilidad de crear distintos tipos de diagramas en un canvas que cubre la mayoría de la pantalla. Estos diagramas deben sincronizarse entre **todos los dispositivos en tiempo real**. Los diagramas pueden pertenecer a las siguientes categorías:

1. Casos de uso UML
2. Mockups de GUI
3. Flowcharts

En cada uno de los casos anteriores se debe presentar un toolbar que le permita a los usuarios arrastrar elementos predefinidos hacia el diagrama (e.g: [www.draw.io](http://www.draw.io)) y ingresar trazos libres y anotaciones.

Los canvas deben poder definir un tamaño arbitrario posiblemente superior al de la pantalla, para lo cual se debe poder hacer scroll y zoom.

Además, los usuarios deben poder enviar mensajes a los demás integrantes de la reunión al estilo de una sala de chat.

Por último, la aplicación debe ser capaz de detectar secondary displays conectados y aprovecharlos de la mejor manera para hacer uso del real state adicional.

~~La aplicación móvil debe permitir a los miembros de un equipo el ingresar a su cuenta y luego a un proyecto. Dentro de un proyecto, deben poder definir nuevos casos de uso o editar los ya existentes. Dado un caso de uso, debe ser capaz de crear esquemas y storyboards de la misma forma que la plataforma web. El conjunto de herramientas provistas debe coincidir en ambas versiones. No obstante, se espera que los alumnos integren la tecnología S Pen para realizar algunas acciones con mayor facilidad.~~

**2.3.2. Tareas**

La aplicación debe permitir al usuario ver sus tareas actuales tal cual están definidas en la plataforma web, y ver el estado de estas. Debe permitir actualizar comentarios, reportar progreso y cambiar su estado (por ejemplo, desde en progreso hasta finalizado). Este módulo debe sincronizarse de la mejor manera con la parte web por lo que debe imperativamente utilizar tecnologías push.

**2.4. Descripción de usuarios**

En primera instancia, los usuarios serán alumnos del curso Proyecto de Especialidad de la Pontificia Universidad Católica de Chile.

Posteriormente se evaluará la extensión de esta herramienta a otros cursos de la universidad, otras universidades y luego su distribución pública open source.

# Información sobre el Contexto del Proyecto

El proyecto debe pensarse como la herramienta idónea para desarrollar software para estudiantes universitarios. Por ende, son los mismos alumnos los más calificados para tomar decisiones macro de diseño pues la calidad del producto es relativa a la satisfacción que ellos mismos tengan al usarlo como herramienta de apoyo al manejo de proyectos.

Se espera que los mismos alumnos contribuyan a seleccionar las funcionalidades que les serían de mayor utilidad y cuál es la mejor manera de presentar la información.

Gran parte de la forma de integrar todas estas funcionalidades dependerán de los mismos alumnos pues ellos como desarrolladores saben cuál es la mejor forma de presentarlas. Por lo tanto, nosotros como clientes no predefiniremos este tipo de requisitos y esperamos que los alumnos colaboren en la toma de decisiones de diseño en función de cuán cómodos se sentirían con ellas.

Se espera al finalizar el semestre que los alumnos presenten una herramienta que integre todas las funcionalidades antes descrita, **con principal énfasis en la plataforma móvil.**

Se exigirá que los alumnos utilicen la plataforma Ruby on Rails para desarrollar la aplicación y Heroku para proveer hosting.

# Información de las Contrapartes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Información de la Contraparte Oficial | | | | |
| Nombre Completo | **José Ignacio Benedetto Causa** | | | |
| RUT | **17.089.269-0** | | | |
| Cargo en la organización | **Estudiante** | | **Antigüedad** | **5 años** |
| Formación profesional | **Estudiante** | | | |
| Teléfono | **2 8355243** | Email | **jibenedettoc@gmail.com** | |
| Dirección Física | **Parcela 15 el peumo marcado la aurora de curacaví** | | | |
| Observaciones | **n/a** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Información de la Contraparte Subrogante | | | | |
| Nombre Completo | **Nicolás Risso** | | | |
| RUT | **17.409.098-K** | | | |
| Cargo en la organización | **Estudiante** | | **Antigüedad** | **5 años** |
| Formación profesional | **Estudiante** | | | |
| Teléfono | **8 8186497** | Email | **narisso@uc.cl** | |
| Dirección Física | **Parque sur 12810 casa 20** | | | |
| Observaciones | **n/a** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Información de la Contraparte Subrogante | | | | |
| Nombre Completo | **…** | | | |
| RUT | **…** | | | |
| Cargo en la organización | **…** | | **Antigüedad** | **…** |
| Formación profesional | **…** | | | |
| Teléfono | **…** | Email | **…** | |
| Dirección Física | **…** | | | |
| Observaciones | **…** | | | |