



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

**USABILIDAD EN UNA HERRAMIENTA DE GESTIÓN DE DESARROLLO DE  
PROYECTOS DE SOFTWARE**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN**

**MATÍAS TORO IPINZA**

PROFESOR GUÍA:

AGUSTIN ANTONIO VILLENA MOYA

MIEMBROS DE LA COMISIÓN:

LUIS GUERRERO BLANCO

GABRIEL IGOR IRIBARREN RIOS

SANTIAGO DE CHILE

ENERO, 2008

## Resumen Ejecutivo

Existen actualmente muchas herramientas de administración de desarrollo de software, las cuales tienen elementos de Planificación y Tracking. Pero no muchas de ellas toman estos dos elementos y los combinan. Una de las herramientas que más hace hincapié en este último tema, es Painless Tracking, la cual es usada actualmente en el curso de CC61A. Esta herramienta está construida sobre una planilla Excel, que la hace no colaborativa y caracterizándola por tener serios problemas de usabilidad.

A raíz de estos problemas y por experiencias personales, nace la idea, y objetivo de esta memoria, de diseñar e implementar la interfaz de usuario de la herramienta *Painless Tracking versión WEB* de manera de hacerla usable: colaborativa, fácil de usar, motivadora en su uso y aportando un mayor valor del que tiene actualmente.

Para el desarrollo de esta aplicación se adaptará una metodología de ingeniería de software existente que consiste en usabilidad ágil. Es decir, utiliza metodologías ágiles de desarrollo pero tomando mucho en consideración la usabilidad del sistema final.

El software obtenido está construido en una plataforma Web, la cual está orientada a una rápida interacción con el usuario, tratando de brindar mayor valor con el menor esfuerzo. Éste, está validado por usuarios finales, los cuales participaron en dos sesiones de pruebas de usuario, los cuales evaluaron positivamente el producto final.

Gracias a las pruebas de usuario y a la metodología de trabajo se pudo obtener un producto que aporta mucho más valor que la versión original, como por ejemplo, mayor visibilidad, mejor interacción, mayor entendimiento de los elementos, mayor rapidez de uso y mayor feedback.

Este producto está listo para ser utilizado por los alumnos del siguiente semestre de CC61A, en donde se espera que siga creciendo según el uso que le den éstos.

## Índice

1. Introducción .....	4
2. Motivación.....	6
3. Objetivos .....	7
3.1.    Objetivo General .....	7
3.2.    Objetivos Específicos.....	7
4. Metodología .....	8
5. Antecedentes .....	9
5.1.    Gestión Ágil.....	9
5.1.    Usabilidad Ágil .....	15
5.1.1.    Metodología ágil centrada en el usuario U-CD .....	15
5.2.    Herramientas.....	17
5.2.1.    Painless Tracking Excel.....	17
5.2.2.    Herramientas Web Similares a la Painless Tracking .....	26
5.2.3.    Iceberg List.....	28
6. Trabajo realizado .....	30
6.1.    Análisis y acciones de usuario de herramientas .....	30
6.1.1.    Painless Tracking Excel.....	30
6.1.2.    Herramientas Web Similares a Painless Tracking .....	30
6.2.    Adaptación de la metodología de usabilidad Ágil .....	37
6.3.    Sesiones de diseño.....	40
6.3.1.    Identificar a los participantes .....	40
6.3.2.    Entrevistas a usuarios.....	41
6.3.3.    Recolección de conceptos previos .....	48
6.3.4.    Definir roles de usuario y modelos de usuario .....	53
6.3.5.    Definir tareas, modelos de tareas y contextos de interacción .....	58
6.3.6.    Definir tareas de usuarios .....	65
6.3.7.    Creación de una interfaz de usuario “wireframe” .....	69
6.3.8.    Testear los contextos de interacción / Evaluación y rediseño del prototipo	75

6.3.9.	Construcción de versión digital.....	76
6.3.10.	Validación con usuarios finales .....	104
6.3.11.	Evaluación de aplicación .....	107
6.3.12.	Últimos cambios.....	108
7.	Conclusiones .....	111
7.1.	Sobre la metodología.....	111
7.2.	Sobre el valor agregado.....	111
7.2.1.	Sobre el Plan.....	112
7.2.2.	Sobre el Track.....	112
7.3.	Experiencia personal .....	113
8.	Trabajo futuro.....	115
9.	Bibliografía .....	116
10.	Anexos .....	118
10.1.	Pauta de evaluación .....	118

## 1. Introducción

Actualmente existen muchas herramientas para administrar y llevar el control y seguimiento de proyectos de software utilizando metodologías ágiles de desarrollo. Algunas herramientas usan tracking o seguimiento, otras usan planificación, otras TODO lists, entre otras.

El problema de las herramientas de seguimiento o timetracker, es que realmente no poseen elementos de planificación, es decir, no incorporan conceptos como velocidad y tiempo de cuando algo se va a hacer o cuanto falta por hacer. De la misma manera ocurre para las herramientas de planificación o herramientas TODO lists, que no incorporan tracking de tareas. ¿Por qué no crear una herramienta que una estos dos conceptos?

Por otro lado el proceso de planificación y seguimiento de tareas, puede ser engorroso, lento y muchas veces difícil de realizar, conduciendo a casos donde debido a la gran cantidad de tiempo necesario para planificar, se retrasan los proyectos o se planifica mal, perjudicando el éxito de éstos. ¿Por qué no facilitar el proceso de planificación?

Por lo tanto se quiere crear una herramienta que mezcle los dos conceptos descritos anteriormente: tracking y planificación, creando una dependencia entre ellos pero sin perder modularidad.

Además, la herramienta debe ser fácil de usar, útil, y que no haga perder el interés a la gente que la está utilizando, teniendo como objetivo brindar la **mayor cantidad de visibilidad y control del proyecto con el mínimo esfuerzo**. La herramienta además introducirá una forma distinta de planificar considerando los tiempos extras de las tareas o acciones que no fueron planificadas de antemano.

Actualmente existe una herramienta que provee con casi todos los puntos descritos anteriormente llamada *Painless Tracking*, la cual está construida sobre una planilla Excel. Esta planilla presenta varios inconvenientes que van en contra de las cualidades

necesarias descritas anteriormente, por ejemplo no soporta el uso simultaneo de dos usuarios de ella, no provee de una interfaz usable (ciertamente es poco intuitiva), dentro de otros problemas.

El objetivo de este proyecto es generar una versión simple y funcional de la planilla original, la cual como al estar construida en Excel, su interfaz y usabilidad no están diseñadas para cumplir con los estándares de las interfaces web actuales. Es decir, no basta con emular una planilla en una página Web, si no que rehacerla de manera de aprovechar las ventajas y desventajas de la Web.

El objetivo principal que tendrá esta memoria será diseñar y proveer una interfaz usable de la versión Web de la planilla, utilizando metodologías ágiles de desarrollo para el diseño que se centren en la usabilidad. Se utilizará una versión adaptada de una metodología de usabilidad ágil propuesta por Jeff Patton. Como el objetivo es la usabilidad, se espera obtener un producto muy bien aceptado por los usuarios finales, agregando nuevas funcionalidades y cambiando la forma de ver y trabajar de la planilla actual.

## 2. Motivación

Existe actualmente una herramienta para gestión de proyectos de software denominada *Painless Tracking*, la cual fue creada por el profesor Agustín Villena para gestionar los proyectos del curso CC61A. Este curso, llamado Proyecto de Software, consiste en la experiencia de ir a trabajar a una empresa real, donde se debe aplicar planificación y trabajo ágil usando la *Painless Tracking* como herramienta de apoyo. Ésta herramienta se basa en metodologías ágiles de desarrollo, y está actualmente construida sobre una planilla de cálculo Excel. Esta implementación, si bien termina cumpliendo su labor, padece de varios problemas, en particular no es multi-usuario, y su uso no es tan simple ni claro como podría serlo, presentando una curva de aprendizaje demasiado elevada.

El tema de memoria consiste en diseñar, e implementar una interfaz de usuario usable para una versión Web de la planilla *Painless Tracking*. Herramienta diseñada para el seguimiento, control y planificación de proyectos que usan metodologías ágiles de desarrollo.

Para ello se planteara el desafío de pasar de un paradigma de tablas (planilla de cálculo) al de una aplicación Web.

### **3. Objetivos**

#### **3.1. Objetivo General**

- Diseñar e implementar la interfaz de usuario de la herramienta *Painless Tracking versión WEB* de manera de hacerla usable: colaborativa, fácil de usar, motivadora en su uso y aportando un mayor valor del que tiene actualmente.

#### **3.2. Objetivos Específicos**

- Analizar los requerimientos de usabilidad de la *Painless Tracking Web* y diseñar soluciones adecuadas usando metodologías de usabilidad ágil.
- Adaptar una metodología ágil de desarrollo Web para usarla en la creación del producto.
- Lograr obtener un producto que sea fácil de usar, que brinde el mayor valor usando el mínimo tiempo de los usuarios.

## 4. Metodología

- Investigar herramientas de planificación similares o que se orienten a desarrollo ágil de software, de manera de rescatar lo mejor de ellas, ya sea buenas interfaces o funcionalidades que ayuden al diseño y construcción de la interfaz de usuario de *Painless Tracking*.
- Evaluar y escoger las tecnologías de desarrollo que se utilizaran en la implementación. (Ajax, u otras). Actualmente se realizará la aplicación en el framework Web Pylons.
- Realizar una análisis de acciones de usuario sobre la herramienta original (basada en Excel) usando una **versión adaptada** de la metodología de usabilidad ágil U-CD propuesta por Jeff Patton en su paper "*Hitting the Target: Adding Interaction Design to Agile Software Development*" [12], y proyectar esas acciones en el nuevo ambiente Web. Ésta metodología fue adaptada por el alumno para cumplir mejor con los objetivos planteados en este proyecto focalizando el desarrollo a la usabilidad.
- Testear el producto con usuarios finales, y obtener una evaluación. A partir de las evaluaciones, modificar el sistema.

## 5. Antecedentes

Conceptos que son fundamentales para el buen uso y las buenas prácticas de la Planilla de Tracking.

### 5.1. Gestión Ágil

Antes se definirán las reglas para gestionar un proyecto usando Gestión ágil [5].

	Cliente	Desarrollador
<b>Desea maximizar</b>	Valor recibido por cada semana de desarrollo	Calidad del trabajo realizado
<b>Puede definir</b>	Qué será implementado, y en qué prioridad, según las necesidades de su negocio	Cuánto se estima que demorará una tarea (idealmente)
<b>Puede cambiar</b>	Funcionalidades solicitadas por otras no implementadas de costo equivalente (canjear)	Sus estimaciones en base a nuevos descubrimientos

Tabla 1 Reglas para gestionar un proyecto [5].

Generalmente cuando se desarrolla software, se efectúa un contrato fijo con el cliente o la contraparte. Esto quiere decir que el precio, las fechas y el alcance del proyecto son fijos, lo que puede ser muy tentativo al momento de dar estimaciones y ganar un proyecto, pero se genera riesgo de que el software puede ser más difícil de implementar. Esto significa trabajar horas extras o trasnocharse trabajando.

**Contratos precio/fecha/alcance fijos** existen porque sirven las necesidades del cliente y del proveedor.

Cliente obtiene	Proveedor obtiene
Costos predecibles	Ingresos predecibles
Entregables predecibles	Demanda predecibles
Planificación predecibles	

**Tabla 2 Lo que el cliente obtiene versus lo que el proveedor obtiene [6]**

Lo malo de este modelo es que no considera la naturaleza incierta de un proyecto de software, por lo que cuando el proyecto resulta ser más difícil de realizar o presenta muchos cambios el cliente no paga más por ello. De hecho si es que el cliente no tiene claro los requerimientos al comienzo, va a pagar por un producto que quizás no es el que quiera y tendrá que renegociar con el proveedor una vez terminado la versión 1 del software.

Otro problema es que este tipo de contratos pone los intereses de los clientes contra los de los proveedores y viceversa.

Cliente	Proveedor
Describe los requerimientos lo más ampliamente posible para obtener lo máximo dado el precio que está pagando	Interpreta los requerimientos lo más fino posible de manera de reducir recursos.
Quiere el trabajo lo antes posible	Quiere terminar el trabajo en la fecha
Quiere la mayor calidad posible	Quiere invertir en la calidad necesaria para que el cliente pague
Programadores exhaustos no es el problema de los clientes	Quiere que el equipo de trabajo sea exitoso y que se quede para el próximo proyecto

### **Tabla 3 Lo que quiere el cliente y lo que quiere el proveedor [6]**

En el manejo de proyectos existen cuatro variables fundamentales:

- **Costos.** Valor que se pagará por el proyecto.
- **Alcance.** El entregable o qué funcionalidades serán implementadas en el producto final.
- **Tiempo.** El tiempo disponible para el desarrollo del producto.
- **Calidad.** Sin errores, bien estructurado, refactorizado, y con set de tests que prueben el 100% de las funcionalidades del sistema.

Con el contrato fijo solo se toman en consideración:

- Costos
- Alcance
- Tiempos

Cuando se llega al deadline del proyecto se sacrifica la calidad de este para obtener el entregable estimado. Esto debido a que como el proyecto es muy susceptible a cambios, y como se dijo anteriormente, la naturaleza del proyecto es incierta, generalmente el proyecto durará más de lo esperado. Aquello generará que cuando se llegue al final del tiempo pactado, no se tendrán todas las funcionalidades prometidas y por lo tanto los productores o desarrolladores tratarán de finalizarlas lo más rápido posible, pudiendo producir que no se testeé bien el producto y por ende obteniendo un software de menor calidad.

Por estas razones se plantea un **Contrato de Alcance variable**, el cual especifica valores para:

- Costos
- Tiempos
- Calidad

Pero no para el Alcance. Es decir que dado un tiempo fijo, se trata de entregar lo mejor posible pero sin especificar en el contrato qué es lo que se obtendrá específicamente.

En resumen:

Variable/Modelo	Tradicional	Alcance variable
<b>Costos</b>	X	X
<b>Alcance</b>	X	
<b>Tiempos</b>	X	X
<b>Calidad</b>		X

**Tabla 4 Variables que toman participación en un contrato Tradicional y otro de Alcance variable.**

Esto introduce el concepto de **Time Boxing [7]**, técnica que dado un tiempo fijo se trata de hacer lo mejor posible. Esto se aplica en lo que es el contrato de alcance variable ya que el proyecto es un time box, que se divide en periodos de tiempo fijo tratando de entregar la mayor calidad posible en cada iteración. Dado que se trabaja con deadlines y el tiempo es limitado, se obliga a *priorizar* las cosas que se tienen que hacer, logrando un mejor producto

Se puede pensar que al plantear un deadline, costo y la calidad esperada en un proyecto, el proveedor puede entregar muy poco, pero de alta calidad. Esto no va a pasar, porque para que esto funcione se tienen que pactar con el cliente timeboxes cortos, es decir varios entregables pequeños o iteraciones. El Cliente va a pagar al momento de recibir cada entregable, y por lo tanto los proveedores van a tratar de dar el mayor esfuerzo para poder terminar el proyecto. En caso de fracaso el Cliente solo va a haber pagado una pequeña parte del proyecto, y no todo el proyecto. En caso de cambios en los requerimientos, este tipo de contrato se adecua muy bien dado las iteraciones cortas.

Al existir disconformidades se negocia con el cliente las funcionalidades que se van a alcanzar en el tiempo fijo y no una nueva fecha de entrega.

Los nuevos intereses bajo un **Contrato de Alcance Variable** son:

Cliente	Proveedor
Describe los requerimientos lo más ampliamente posible para obtener lo máximo dado el precio que está pagando	Feliz de aceptar cambios en los requerimientos
Quiere el trabajo lo antes posible	Quiere terminar a tiempo. Feliz de aceptar la mayor cantidad de funcionalidades posibles sin sacrificar calidad
Quiere la mayor calidad posible	Quiere maximizar la calidad del proyecto
Cliente quiere que los programadores vuelvan a trabajar después de la iteración para terminar el proyecto	Quiere que el equipo de trabajo sea exitoso y que se quede para el próximo proyecto

**Tabla 5 Intereses de Clientes y Proveedores bajo un Contrato de Alcance Variable [6]**

Este contrato mantiene lo que obtiene el cliente y el proveedor con los contratos fijos, salvo por las “entregas predecibles”. Lo que se obtendrá al final del contrato puede ser muy diferente a lo que puede haber pensado el cliente al inicio, pero produce en el cliente una sensación de control por sobre el alcance del proyecto, produciendo al final un producto que es mucho más valioso para él, completando en cada contrato el 100% de las tareas más prioritarias.

Al realizar este tipo de contrato se dejan de lado los requerimientos funcionales pero se toman en consideración lo que son los casos de uso o historias de usuario.

Finalmente lo que hace valioso este tipo de enfoque es:

- Clientes pueden cambiar de idea
- Proveedores no sacrificarán calidad cuando hayan cambios
- Los intereses de los clientes y los proveedores están alineados
- El conocimiento de que las dos partes ganan durante el proyecto puede influir en el producto final

Al usar **Time Boxing e Iteraciones de largo fijo** para abordar la planificación obtiene lo siguiente:

- Se puede mirar hacia iteraciones anteriores y analizar cuanto es lo que se avanzó, aprendiendo a estimar mejor el futuro usando el pasado, respondiendo preguntas como
  - ¿Cuánto trabajo realizamos el mes pasado comparado con lo que estimamos inicialmente?
  - ¿Cuánto trabajo realizamos comparado con el mes anterior?
  - ¿Cuántas funcionalidades van a ser realizadas antes del deadline?
- En cada iteración se obtiene un producto que es mucho más valioso para el cliente dado que se priorizó con antelación las funcionalidades
- Muy susceptible al cambio ya que al trabajar en pequeñas iteraciones se puede cambiar el rumbo del proyecto en cualquier momento
- Aborda mucho mejor tareas que son complejas, debido a que las fragmenta en pequeñas tareas más fáciles de realizar

Este enfoque en la práctica se aplica definiendo una **planificación**, especificando las funcionalidades que se deben realizar y las tareas que las han de componer.

Recapitulando, el proyecto se divide en iteraciones de largo fijo o time boxes, y en cada iteración se deben realizar la mayor cantidad de funcionalidades posibles. Pero para ello se debe definir una planificación, especificando en detalle estas funcionalidades y las tareas que las han de componer.

Es fundamental estimar el tiempo que tardará el desarrollo de cada funcionalidad ya que como el tiempo es limitado, estas deben ser priorizadas. Si se trata de estimar

directamente cada funcionalidad, puede que se produzcan errores de estimación grandes, por lo que se deben estimar las tareas que las componen, ya que son más pequeñas reduciendo la ambigüedad y complejidad de ésta.

## **5.1. Usabilidad Ágil**

La implementación de software usando metodologías ágiles de desarrollo es rápida y entrega software de excelente calidad. El problema es que ese software no necesariamente es amigable para el usuario y muchas veces no es usado ya que no es lo que se quiere o posee funcionalidades que no se necesitan, es decir, mucho tiempo gastado en funcionalidades que no se van a utilizar por los usuarios finales. Esos mismos usuarios, después ocuparán mucho más tiempo en hacer tareas de su negocio, forzándose a trabajar con un software que no es útil para ellos. Por el otro lado, software que se concentra mucho en la usabilidad puede ser muy lento de desarrollar, no cumplir con los plazos o terminar con un software de mala calidad.

Es por esto que se desea utilizar una metodología que una los dos conceptos para obtener un software de calidad y a tiempo, pero que también cumpla con las expectativas y necesidades de los usuarios.

### **5.1.1. Metodología ágil centrada en el usuario U-CD**

Jeff Patton propone en su paper "*Hitting the Target: Adding Interaction Design to Agile Software Development*" [12], una metodología de usabilidad ágil, la cual se explica a continuación.

#### **1. Identificar los participantes.**

Juntar a un diverso grupo de personas en una habitación a colaborar en el diseño. Incluir expertos en el dominio, personas encargadas del negocio, programadores y testers.

#### **2. Recolección de conceptos previos.**

Soltarse. Cada uno debe hacer un *Brain Dump* acerca del software que se va a escribir. Describir funcionalidades innovadoras que se desea que el

producto tenga. Escuchar las dudas y opiniones de cada uno de los integrantes. Escribir esto en papeles y colgarlos en la muralla.

### 3. **Revisar el dominio del problema.**

Expertos en el dominio del problema explican el proceso de negocio que existe actualmente. ¿Quién está involucrado en el proceso? ¿Que combinación de procesos manuales e informáticos deben realizar los participantes para alcanzar sus objetivos?.

### 4. **Definir roles de usuario y modelos de usuario.**

Hacer un *Brainstorm* de los roles de usuario en tarjetas de 3 x 5. ¿Quién usará este software? ¿Cuáles son sus objetivos? Priorizar los roles revolviendo la pila de tarjetas. Escoger lo roles más importantes. Etiquetar estos roles como *focales*. Ponerlos en la mesa de manera de que los que tengan más similitud estén más cerca. Discutir los roles entre cada participante. Este es un modelo de roles.

### 5. **Definir tareas y modelos de tareas.**

Ahora que se sabe quién usará el software, hacer *brainstorm* de tareas que estos roles deberán realizar para cumplir sus objetivos en tarjetas de 3 x 5. Revolver las tarjetas y priorizarlas en orden de importancia y frecuencia. Notar las más importantes y frecuentes. Etiquetar éstas como tareas *focales*. Situar las tarjetas en la mesa. Colocar cerca las tareas que sean similares con otras, y juntas, las que dependan de otras. Colocar tareas que no tengan que ver con otras, bien separadas. Discutir la relación que existe entre cada una de las tareas. Éste es un modelo de tareas.

### 6. **Definir contextos de interacción.**

Para cada grupo de tareas obtenido en el paso anterior definirlo como un contexto de interacción.

## 7. Definir tareas de usuario.

Para cada tarea en el contexto de interacción, escribir un Caso de Tarea directamente en la tarjeta. El Caso de Tarea debe ser escrito de manera conversacional. Usabilidad ágil recomienda simplificar y generalizar estos Casos de Tarea. Usar una forma conversacional los hacen fácil de leer. Limitar el alcance y objetivo, previene de que sean muy amplias o muy detalladas. Generalizarlas produce que queden cortas, disminuyendo el tiempo que se ocupa después para hacer referencias.

## 8. Crear un prototipo abstracto.

Por cada contexto de interacción, usando los casos de tareas que se crearon, crear un prototipo de interfaz abstracta. Al final de este proceso se sabrá, cuales son los componentes que irán en el contexto de interacción.

## 9. Crear una interfaz de usuario “wireframe”.

Con lápiz y papel construir prototipos fieles a una versión digital de cada contexto de interacción.

## 10. Testear los contextos de interacción.

Usar *role-playing*, para iterar por cada uno de los pasos de los casos de tarea en los contextos de interacción. Un participante toma el rol del usuario final, y otro de la interfaz de usuario. Validar que se cumplan los objetivos usando ese contexto de usuario.

## 5.2. Herramientas

### 5.2.1. Painless Tracking Excel

#### ***Painless Schedules***

Antes de hablar de la Painless Tracking, se debe describir acerca de Painless Schedules [15]. Esta herramienta fue propuesta por un ingeniero norteamericano llamado Joel Spolsky en el 2000.

Esta herramienta nació bajo la premisa que los programadores no planificaban porque era “doloroso”, no le veían la utilidad y era demasiado complejo comparado a lo que ganaban con ello. Para que fuera “indolora” se hizo lo siguiente:

1. Usar Excel. Usar un software que fuera flexible y fácil de usar, evitando software de planificación más complejo como Microsoft Project, ya que con él se debían mantener dependencias entre tareas, entre otros problemas.
2. Hacerlo simple. La planificación contenía solo siete ítems:
  - a. Funcionalidad
  - b. Tarea
  - c. Prioridad
  - d. Estimación original
  - e. Estimación actual
  - f. Tiempo ocupado
  - g. Restante
3. Cada funcionalidad estaba compuesta de varias tareas.
4. Las estimaciones debían ser en horas y no en días o semanas. Esto hacía que las estimaciones fueran más realistas
5. Con la estimación original y la actual se podía aprender a estimar
6. Se debía actualizar el tiempo ocupado cada día que se trabajara en una tarea y la estimación actual cada vez que se detectara que la estimación original no se fuera a cumplir
7. Agregar tareas adicionales a la planificación como vacaciones, corrección de errores o bugs, integración con otros programadores, en resumen tareas que no se habían planificado de antemano
8. Pensar la planificación como bloques de madera. Esto quiere decir que si se tenía una cantidad de tiempo limitado y tareas que no fueran reducibles en el tiempo, entonces se debía priorizar y ver cuáles iban a quedar afuera tal cual como un **Time Box**. Por lo tanto esta metodología se adecuó perfectamente con contratos de alcance variables e iteraciones de largo fijo, es decir una prometedora herramienta de gestión ágil.

### ***Historia de la Painless Tracking [22]***

Los proyectos del curso CC61A Proyecto de Software eran planificados usando Microsoft Project y su paradigma de Carta Gantt, bajo la premisa de que “esta es la herramienta que más se usa en la industria”. A pesar de que se planteaba en la teoría a los alumnos un modelo de gestión iterativo e implementado, los resultados obtenidos no eran satisfactorios, y la visibilidad que aportaban las Cartas Gantt obtenidas lo eran menos aún. Al analizar la situación, se pudo encontrar que el tiempo que se requería para actualizar las Cartas Gantt por parte de los administradores de proyecto era simplemente prohibitivo (llegando a ocupar hasta el 50% del tiempo), por lo cual la planificación continúa era abandonada rápidamente, y por ende toda posibilidad de poder observar y enmendar el rumbo.

Fue así que se probó de manera experimental la *Painless Schedules*, por ser una herramienta muy simple de planificación.

Esta herramienta carecía de funcionalidades de gestión administrativa y de lo que es el tracking del proyecto. Por esto nació la herramienta Painless Tracking que introdujo funcionalidades administrativas que dan una visión global del proyecto y que introdujo además el tracking del proyecto que ayuda a una gestión más operacional.

A partir de su aplicación en algunos grupos de CC61A, comenzó a evolucionar la herramienta hasta adquirir la forma actual. En el año 2005, al adoptar CC61A un modelo de gestión ágil para sus proyectos, la ahora denominada *Painless Tracking* adquirió su estatus de herramienta oficial de planificación y seguimiento de proyectos del curso.

### ***Descripción***

La labor de un ingeniero de software se puede resumir en la continua gestión de tiempo y recursos para generar el mayor valor a un cliente, donde el primero, a través de funcionalidades, validará que esta necesidad esté o no cubierta. Las funcionalidades deben, por su naturaleza, estar expresadas en un **lenguaje común** que tenga sentido tanto para el cliente como los desarrolladores. De esta manera si se requiere que el software realice una operación “A”, el software implementará

funcionalidad (F(A)) que efectivamente apuntará a resolver el problema de negocio del cliente.

Puede parecer que con lograr traducir las necesidades a funcionalidades sería suficiente, y que simplemente bastará con agregar las funcionalidades a la planificación del proyecto, para luego comenzar a programar el código. Sin embargo, las funcionalidades a su vez hay que traducirlas al ámbito técnico. Por ejemplo, si una funcionalidad consiste en *“listar los clientes de la base de datos de la empresa, en una página web”*, se requerirá un procedimiento que sea capaz de conectarse a la base de datos de la empresa. Este procedimiento en si no es una funcionalidad, ya que al cliente no le interesa si se puede o no conectar a la base de datos (de hecho, el concepto base de datos puede no significar nada para él); lo que realmente le interesa (por ende, aporta valor al producto), es que se desplieguen los datos de los clientes en la página web. A estos procedimientos que forman parte de una funcionalidad, los denominaremos **tareas**. Las tareas son en sí las subdivisiones de las funcionalidades que permiten llegar a una mejor estimación del tiempo que ocuparán dado que son más específicas. Luego, la división por partes de un proyecto de desarrollo de software sería como lo expresado en la figura que se muestra a continuación:

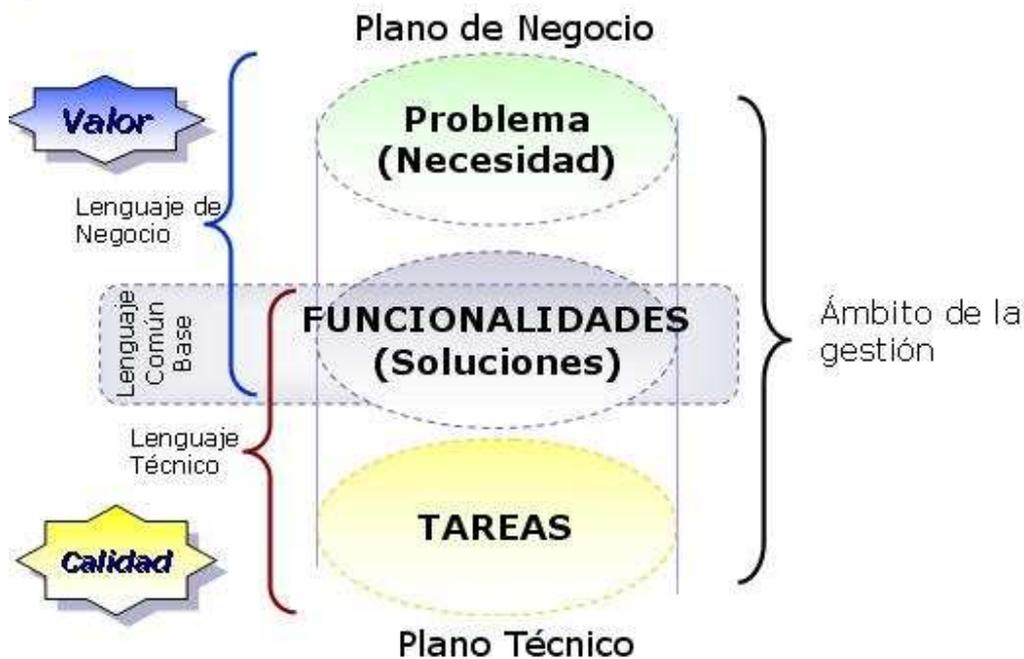


Ilustración 5-1 Vista de un proyecto como un conjunto de necesidades que se traducen en funcionalidades [19]

Esta forma de ver la situación, se orienta al valor dado que cada funcionalidad permitirá al software generarle al cliente herramientas valiosas según la perspectiva de su propio negocio, que es en definitiva por lo que se está pagando.

La naturaleza incierta de los proyectos de software hace que la forma más efectiva de lograr un avance claro es dividir el proyecto en iteraciones de largo fijo, al final de cada cual se debe obtener un software funcional, que permita al cliente validar su aplicabilidad al problema de negocio lo antes posible y así poder retroalimentar y ajustar el proyecto. Para que esto funcione, es necesario definir una **planificación** en la cual se especifica las funcionalidades que se deben realizar y qué tareas las han de componer.

En la realización de una planificación es necesario estimar el tiempo que tardará el desarrollo de cada etapa del proyecto. Si se trata de estimar directamente cuánto demorará en realizarse una funcionalidad en particular, puede que se produzca un

error de estimación significativo. Sin embargo, si se estima cuánto demorará cada tarea, la estimación se vuelve más sencilla de realizar, dado que las tareas debieran ser pequeñas y no debieran demorarse más que un par de horas en realizarse. Si la estimación del tiempo de desarrollo de una tarea es demasiado extensa (48 horas, por ejemplo) quiere decir que falta aún más por desglosar, siendo necesario definir tareas con mayor detalle y por ende de menor tiempo de desarrollo.

Con respecto a las unidades de medición de tiempo en la estimación de tareas, es importante notar que se hará en *horas* dado que si se hace en *días* o *semanas* se estará creando una contradicción con lo recién explicado.

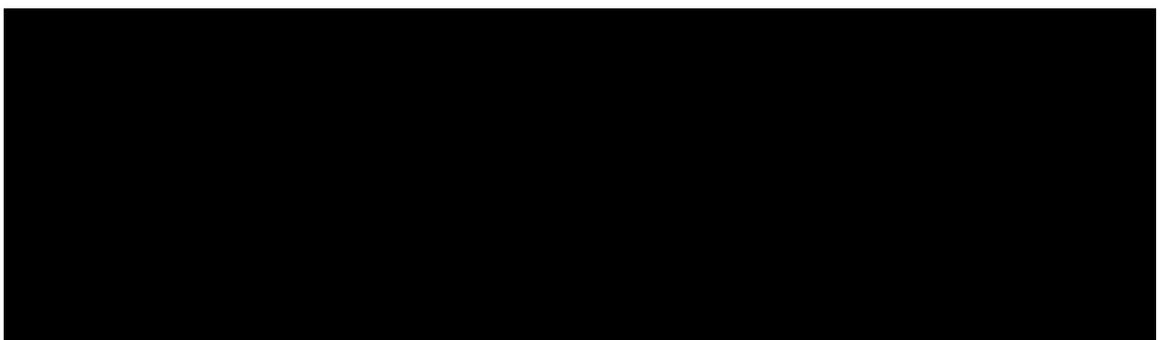
### **Componentes operativas**

#### Planificación

El objetivo de esta interfaz es registrar qué es lo que se planea realizar en el proyecto de desarrollo (funcionalidades) y cómo estos serán construidos (tareas)

Al planificar se definen las Funcionalidades que componen el proyecto o que se necesitan para su desarrollo. Para desarrollar cada Funcionalidad se necesitan ejecutar o realizar una serie de Tareas.

A continuación se muestra una vista del componente Planificación:



**Ilustración 5-2 Ejemplo de la vista del componente de Planificación y las columnas que componen cada tarea**

Cada fila representa una tarea.

El primer campo *Iteración*, indica el número de la iteración a la cual pertenece la tarea.

El segundo campo *Funcionalidad* indica el nombre de la funcionalidad a la cual pertenece la tarea

El tercer, quinto y sexto campo (*Tarea, id tarea, descripción*) representa el nombre de la tarea, el identificador de ésta y su descripción respectivamente

El cuarto campo *Prioridad*, indica que tan prioritaria es la tarea dentro de la iteración, puede tomar tres valores en orden creciente de importancia: Deseable, Importante, Fundamental

El séptimo campo *Responsable* indica quien es el responsable de que se termine la tarea (no necesariamente es la persona que va a realizarla)

El octavo campo *Orig. Est.*, indica la estimación original del tiempo de término de la tarea

#### Tracking

El objetivo de la interfaz de tracking (o seguimiento del proyecto) es llevar un control de lo realizado hasta el momento. Es importante hacer notar que lo efectuado puede o no corresponder a algún ítem de la planificación, lo que es debido a la natural incertidumbre que se tiene al planificar.

A continuación se muestra una vista del componente Tracking:

Funcionalidad	Iteración	Fecha	ID Tarea	Ocupado	Restante Estimado	Desarrollador	Qué se hizo	Qué falta por hacer
Crear ambiente desarrollo	1	23/07/07	ambiente_desarrollo	4	0	Andres Barraza	Se descargó lo necesario y se dejó funcionando ambiente de desarrollo	
NoPlanificado	1	23/07/07		2	0	Andres Barraza	Acuerdo de condiciones de trabajo con el cliente	
NoPlanificado	1	23/07/07		2	0	Carlos Delgado	Acuerdo de condiciones de trabajo con el cliente	
Mostrar la hora en formato digital y	1	24/07/07	clase_pulso	3	3	Andres Barraza	Se aprendió a usar método time de la clase TIME de java. Se comenzó con la programación de los métodos de la clase Pulso	Falta generar el envío del pulso para que pueda recibirlo la clase Reloj
Mostrar la hora en formato digital y	1	24/07/07	clase_reloj	4	3	Bruno Correa	Se aprende a utilizar las clases necesarias para la implementación. Se comienza con el desarrollo para captar el pulso enviado por la clase Pulso.	Falta entregar la señal a las clases TiempoDigital24 y TiempoDigitalAMPM
Mostrar la hora en formato digital 2	1	25/07/07	clase_TD24	4	0	Carlos Delgado	Se implementa la clase y esta es capaz de recibir la información de la clase Reloj, de modo que puede mostrar la hora en formato digital de 24 horas.	
Mostrar la hora en formato digital y	1	25/07/07	clase_pulso	4	0	Bruno Correa	Se termina de implementar el envío del pulso a la clase reloj. Se demora una hora más de lo estimado	
Mostrar la hora en formato digital y	1	26/07/07	clase_reloj	3	0	Bruno Correa	Se termina de implementar la clase siendo esta capaz de enviar las señales de pulso recibido a TiempoDigital24 y TiempoDigitalAMPM	
Mostrar la hora en formato digital A	1	26/07/07	clase_TDAMPM	2	0	Daniel Echaurren	Se implementa la clase y esta es capaz de recibir la información de la clase Reloj, de modo que puede mostrar la hora en formato digital AM/PM. Dado que es similar a la clase TiempoDigital24, el tiempo de desarrollo es menor en un margen de 2 horas menos de lo estimado	
Validar software con gerente	1	28/07/07	validacion_cliente	1	0	Andres Barraza	El cliente queda conforme con esta iteración dejando el camino abierto a la implementación de nuevas funcionalidades para la segunda iteración.	

**Ilustración 5-3 Componente Tracking de la Painless Tracking Excel**

Como se puede ver en la figura, existen varios campos con información relacionada con el avance diario del proyecto.

El primer campo, *funcionalidad o valor*, indica la funcionalidad a la que pertenece la tarea sobre la cual nos encontramos. Hay que recordar que cada línea del tracking representa una tarea.

El segundo y tercer campo (*iteración y fecha*), indican la iteración y fecha en la que se está desarrollando la tarea seleccionada.

El cuarto campo (*ID Tarea*), indica el código que tiene la tarea. El código de la tarea se define en la planificación de la painless tracking. Cuando se ingresa el código de la tarea en este campo, el primer campo se actualiza inmediatamente con la funcionalidad a la que se asocia esta tarea.

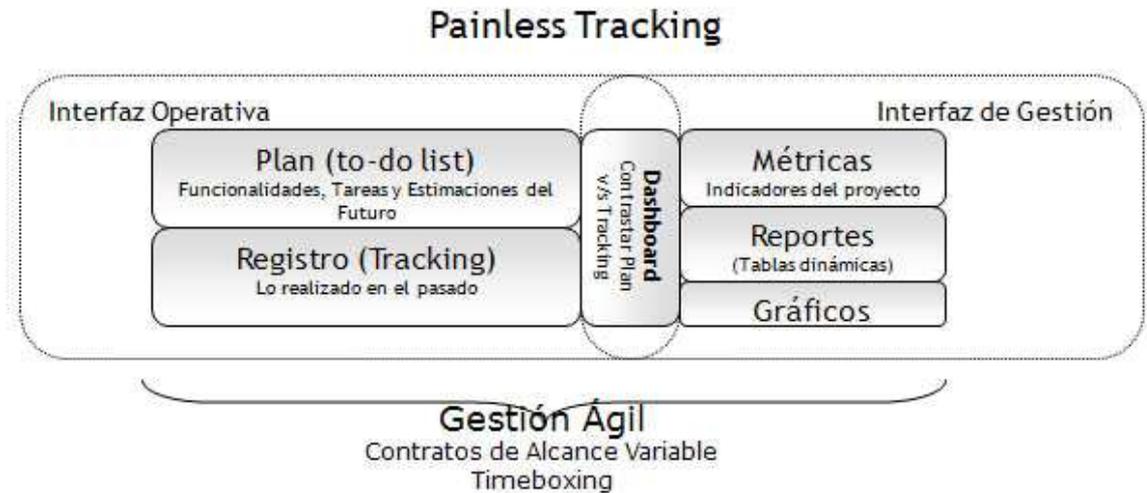
En el quinto y sexto campo (*Ocupado y Restante Estimado*) se anota la cantidad de horas que se ha desarrollado la tarea y cuánto tiempo de desarrollo restante se estima.

Cuando una tarea se ha dado por finalizada, se asume que el tiempo estimado de desarrollo restante es cero.

Los tres últimos campos contienen la información de quién es el implementador, qué es lo que se lleva y que es lo que falta de la tarea seleccionada.

## 5.2.2. Herramientas Web Similares a la Painless Tracking

### Componentes de herramientas de gestión ágil



**Ilustración 5-4 Componentes de una herramienta de gestión ágil [22]**

Como se muestra en la figura ( Ilustración 5-4), existen distintos componentes dentro de una herramienta de planificación ágil. Por un lado está el componente Plan, que está orientado a las personas encargadas de la administración operativa del proyecto, y por otro lado está el Tracking, orientado a personas que trabajan en el actual desarrollo de éste.

El Plan consiste en la herramienta que permite planificar el desarrollo del software, ya sea dividiendo el proyecto en una lista de tareas, en funcionalidades, etc. El Tracking por otro lado, es la herramienta que permite dejar un registro de que es lo que se ha hecho, ya sea dentro de una tarea, dentro del día, etc. Esto no quiere decir que los que trabajan en el Tracking son distintos usuarios a los que trabajan con el Plan, esto no es así y por lo tanto debe de existir un componente que contraste estos dos, el cual es llamado Dashboard. Este Dashboard hace match entre lo que se planificó y lo que se trabajó, mostrando el real avance del proyecto.

Por otro lado también se tienen los componentes de Gestión como son las Métricas, los Reportes y Gráficos, los cuales están orientados a un usuario que necesite una visión más global del proyecto. Este último componente es el que más agrega valor a

las aplicaciones de gestión ágil ya que entrega inteligencia a lo que se realiza día a día en el proyecto.

A continuación se nombrarán herramientas que conforman el estado de arte.

### ***Herramientas tipo Project***

Estas herramientas se mencionarán solo como referencia ya que son basadas en cartas Gantt, la cual sigue un paradigma distinto de gestión de proyectos que sigue la Painless Tracking.

Entre las herramientas que existen:

- Microsoft Project
- DotProject
- Taskjuggler
- GanttPV

### ***Herramientas de planificación o TODO list***

Estas herramientas fundamentalmente se basan en definir un conjunto de tareas y luego chequear si es que se hicieron o no. Entre ellas se encuentra TRAC, que a pesar de llevar el nombre de una herramienta de tracking, se basa en la definición de tickets o tareas, por ende se debe considerar como una herramienta de planificación.

A continuación se presentan herramientas de este tipo.

- TODOIST [18]
- REMEMBER THE MILK [14]

### ***Herramientas de Tracking***

Estas herramientas se basan en anotar registros o lo hecho. La versión más básica se puede pensar como un log de actividades.

A continuación se presentan herramientas de este tipo.

- TIMESHEETS [17]
- TICK [16]

### ***Herramientas de Gestión ágil***

A continuación se presentan herramientas de este tipo.

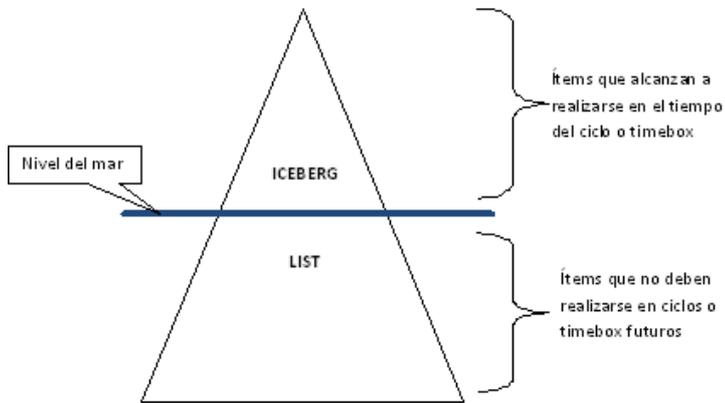
- VERSIONONE [20]
- XPLANNER [21]

#### **5.2.3. Iceberg List**

Entender el concepto de iceberg list [2] es necesario para la realización de este proyecto. El contexto en que se usa esto es en un proyecto donde, se tiene un tiempo fijo de entrega e ítems que componen esta entrega.

El iceberg list es una lista prioritaria de ítems. Lo que está “sobre el agua” indica los ítems que deben ser entregados al finalizar el ciclo de entrega o Timebox. La parte que está “bajo el agua”, lista todo lo que deberá ser entregado en ciclos futuros. El nombre viene de que cuando se inserta un nuevo ítem en la lista sobre el agua, empuja todo lo demás abajo, y algo que estaba sobre el agua cae bajo esta, es decir, no se entregará en este ciclo.

Esta lista es útil cuando se tienen proyectos donde los requerimientos cambian frecuentemente.



**Ilustración 5-5 Iceberg List**

Como se ve en la figura ( Ilustración 5-5) , lo que está sobre el “Nivel del mar” será entregado en el ciclo de entrega. Lo que está bajo el “Nivel del mar”, deberá ser entregado más adelante.

## 6. Trabajo realizado

### 6.1. Análisis y acciones de usuario de herramientas

#### 6.1.1. Painless Tracking Excel

Las acciones más recurrentes dentro de la planilla de tracking son: llenado del Tracking y modificación del Plan. Esto genera dos tipos de perfiles de usuarios: el planificador y el tracker.

El planificador generalmente modifica la planilla una o dos veces a la semana, en cambio el Tracker debe llenar el Track todos los días.

Por lo tanto las acciones más recurrentes de los usuarios son:

- Llenado de Track. Hay dos formas de llenar el Track: buscar el id de la tarea en el Plan y después escribir que se hizo, y la otra es escribir todo lo que se hizo en un día y después buscar en el Plan registro por registro cual es el id de la tarea a la cual pertenece.
- Modificación del Plan. El plan es modificado en pocas oportunidades pero trae consigo varios problemas, ya que no agrupa las tareas por ningún criterio
- Consulta de métricas. La consulta de métricas es usada de manera de obtener valor agregado del software, es decir darme reportes acerca el avance del proyecto.

#### 6.1.2. Herramientas Web Similares a Painless Tracking

A continuación se muestra una serie de herramientas Web que se tienen funciones similares a Painless Tracking. Las tablas descritas a continuación describen una lista de características o funcionalidades que posee la aplicación, y la última columna describe cuales de ellas son de interés para incluir a la interfaz gráfica a construir.

#### ***Herramientas de planificación o TODO list***

- TODOIST [18]

Herramienta orientada a la planificación de proyectos mediante tareas.

Se crean proyectos. En cada proyecto se van creando ítems, los cuales son tareas. A estas tareas se les puede dar prioridad y una “seudo jerarquía” especificando la indentación de estos ítems.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

Característica o funcionalidad	Interés
La edición de los ítems es “in-place”, al momento de hacer clic en el ítem. Esto se da por sorpresa ya que no da la sensación de ser editable.	
Un proyecto puede ser mapeado a un color para hacer un mapping mental de él	
Los ítems se agregan en una posición escogida por el usuario.	
Prioridades pintan el ítem. Pero los colores no son intuitivos. De mayor prioridad a menor prioridad: Rojo -> azul -> verde -> negro.	
Se pueden mover tareas de un proyecto a otro. Se podrían mover las tareas inter productos con drag and drop.	
Las tareas se finalizan con un checkbox al lado izquierdo de éste y pasan a la lista de ítems completados.	
Provee la funcionalidad de obtener una versión imprimible de la lista de tareas a completar.	X
Los ítems completados son escondidos una vez que se reingresa al sistema. De esta manera no se llena la pantalla de tanta información (que se da actualmente en la versión Excel de la Painless Tracking).	X
Para ordenar aprieto en Re-order. Esto me permite hacer drag and drop de los proyectos. Un poco confuso al comienzo.	
Se pueden ver las tareas prioritarias (independiente del proyecto)	
Permite hacer UNDO al borrar una tarea. Muestra las acciones y opciones	X

de deshacer en un globo visible y llamativo.	
No pregunta si se quiere eliminar una tarea, simplemente la borra.	

**Tabla 6 Características o funcionalidades de TODOIST**

- REMEMBER THE MILK [14]

Herramienta orientada a la planificación de proyectos mediante tareas.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

Característica o funcionalidad	Interés
Diagramación: Tareas en un panel principal y un menú a la derecha contextual.	
Las tareas están divididas por categoría: Estudio, Personal y Trabajo.	
Se pueden mover tareas desde una categoría a otra mediante un combo box. Esto limita la cantidad de categorías.	
Las tareas se añaden con la tecla ENTER. Es poco intuitivo.	
Los diálogos de las acciones, por ejemplo, de si uno está seguro de eliminar un elemento, se producen en un área del menú a la derecha, exclusiva para este tipo de cosas. De esta manera no le quita al usuario visibilidad en ningún momento con ventanas emergentes o pop-up.	X
Cuando se selecciona una tarea se muestra la descripción de ésta en el menú contextual de la derecha. Esto permite editar sin cambiar de página.	X
Añadir una tarea se consigue solamente ingresando el nombre de ésta. Detalles más específicos como hora, y recurrencia de esta se ingresan después al ver la información de la tarea como se describió en el punto anterior.	X
Se le pueden poner Tags a las tareas.	
Confuso: El detalle de una tarea se encuentra en el menú contextual de la	

derecha, pero si se quiere darle prioridad o finalizarla se tiene que seleccionar esa acción de un combo box ubicado en otra parte de la página.	
Excesiva utilización de teclado para realizar operaciones. No todas las operaciones se pueden hacer con el mouse.	

**Tabla 7 Características o funcionalidades para REMEMBER THE MILK**

### ***Herramientas de Tracking***

Estas herramientas se basan en anotar registros o lo hecho. La versión más básica se puede pensar como un log de actividades.

- TIMESHEETS [17]

Este programa está diseñado para medir las horas trabajadas de los empleados en proyectos. Dadas estas horas se puede calcular cuánto es lo que se le tiene que pagar. Este proceso otorga más precisión definiendo horas Billable y Non-Billable, donde las primeras se pagan y las segundas no.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

<b>Característica o funcionalidad</b>	<b>Interés</b>
El sistema cuenta con un calendario siempre visible para seleccionar rápidamente la fecha de registros que se muestra actualmente.	X
El sistema provee de un Task Timer que mide en tiempo real cuanto uno tarda en completar la tarea. Basta con seleccionar una tarea y apretar en el botón Play para que calcule realmente cuanto es lo demorado en esa tarea.	
También provee de un sistema de reportes para generar informes de las horas trabajadas en los proyectos asignados. Esto con el fin de saber al empleador cuanto es lo que le tiene que pagar.	
Como el programa se orienta a controlar lo trabajado por los empleados, se convierte en una herramienta muy útil para los empleados que les pagan	

por hora.	
El ingreso de registros funciona asociándolos a proyectos, pero no profundiza más que eso. Tampoco brinda con información acerca el avance de estos proyectos.	
Tiene un sistema de favoritos para clasificar los registros.	X
Se puede especificar cuánto es lo pagado por hora de trabajo en cada proyecto	

**Tabla 8 Características o funcionalidades para TIMESHEETS**

- TICK [16]

Creado para llevar el control de avance de proyectos. Esto gracias a barras de progreso.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

Característica o funcionalidad	Interés
Se crean Clientes. Luego para cada cliente se crean proyectos.	
Para cada proyecto se debe ingresar las horas que se deberían ocupar para completarlo.	
Opcionalmente se pueden crear tareas dentro de cada proyecto. A cada tarea se le asignan horas. No permite que la suma de las horas de las tareas superen a la del proyecto.	
El timetracking se realiza eligiendo proyecto luego la tarea, después las horas gastadas y una descripción.	
Se muestra un menú a la derecha que permite la auto-completación de los campos en el formulario de ingreso de registro. Para llenar un registro de una tarea basta hacer clic en ella en ese menú.	X
Se muestra arriba del formulario de ingreso de registros, paletas con fechas	X

de los días. Esto permite visualizar el tracking hecho en ese día o en anteriores.	
No tiene conceptos de tiempo inicial y tiempo final, elemento fundamental, porque la Painless Tracking permite aprender a estimar. No te permite aprender de tus estimaciones.	

**Tabla 9 Características o funcionalidades para TICK**

**Herramientas de Gestión ágil**

- VERSIONONE [20]

Creado para manejar y planificar proyectos dentro del área de desarrollo ágil de software. Su objetivo es simplificar los procesos de planificación y tracking.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

Característica o funcionalidad	Interés
<p>La planificación se divide en tres módulos:</p> <ul style="list-style-type: none"> <li>• Planificación y manejo de requerimientos: acá se definen Features y Defects.</li> <li>• Planificación de entregas de software. Definir que Features pertenecen a que entrega.</li> <li>• Planificación de iteraciones: crear tareas a partir de Features, se debe estimar el tiempo que demorarán cada una de ellas. Existe además la visión de cuanto ha estimado cada desarrollador.</li> </ul>	
Se usa drag and drop para la priorización de elementos dentro de la planificación. Los elementos de más arriba son los más prioritarios.	X
El tracking presenta la posibilidad de cambiar el estado de las tareas (finalizarlas) y estimar el tiempo restante de ésta.	
El tracking presenta de un gráfico Burndown de la iteración de lo que falta	

por hacer.	
El módulo de reportes presenta una gran variedad de gráficos correspondientes al Dashboard de la aplicación. Presenta una visibilidad global del proyecto, entregas e iteraciones.	
El tracking es distinto al de la Painless Tracking ya que no se especifica qué es lo que se hizo y solo presenta la posibilidad de cambiar el tiempo restante de las tareas. Tampoco existe el concepto de tareas no planificadas.	

**Tabla 10 Características o funcionalidades para VERSIONONE**

- XPLANNER [21]

Herramienta diseñada para planificación y tracking de proyectos que usan eXtreme Programming (XP). Se basa en la definición de features que son descompuestos posteriormente en tareas con estimaciones de término correspondiente. Esto de manera de ver cuáles de las features necesitarán renegociación con el cliente. Esta herramienta fue creada para soportar esta última función.

A continuación se muestra una lista de características o funcionalidades que posee la aplicación.

Característica o funcionalidad	Interés
Para cada feature se crean tareas las cuales se pueden priorizar.	
Cada tarea se puede completar, pero no se puede re-estimar el tiempo restante de ellas.	
Se puede ver el avance de una tarea con una barra de progreso.	X
Las tareas tienen un TimeLog que indica los intervalos de tiempo en los cuales se trabajó, especificando si se trabajó de pares o solo.	
Posee una visión detallada de un desarrollador, especificando las tareas que ha cerrado, las tareas en las que está actualmente trabajando, y las	X

que aún no comienza que son responsabilidad de él.	
Muestra un gráfico de métricas que indica con barras cuanto es lo trabajado por cada desarrollador y cuanto es el total de horas comprometidas por este para trabajar.	
Presenta un gráfico con la velocidad del proyecto que refleja lo que originalmente se estimó, lo que está completo y lo que necesita ser completado.	
Presenta planificación detallada pero el tracking es pobre.	

**Tabla 11 Características o funcionalidades para XPLANNER**

## **6.2. Adaptación de la metodología de usabilidad Ágil**

La adaptación se hizo, debido a que no se contaban con los mismos recursos que se tenían para la metodología original (grupo de trabajo) y porque se encontró que se necesitaba expandir y complementar mucho más esta metodología. A continuación se muestran los pasos que se seguirán en el trabajo.

### **1. Identificar los participantes.**

En este paso se identifica quienes son los usuarios que ocupan la planilla actual, quienes la usarán y quienes la han usado. También se identificará quienes son los desarrolladores o las personas que participarán en el diseño conceptual para definir acciones de usuarios o funcionalidades de la Painless Tracking.

### **2. Entrevistas a usuarios.**

Se pretende integrar a los usuarios lo más posible en el diseño de una interfaz usable. Para ello se contactará con usuarios dispuestos a colaborar en la planilla (usuarios, ex – usuarios y futuros usuarios), y con ello identificar rutinas al momento de planificar o registrar diariamente lo hecho en el día (planificación y tracking). Se espera obtener de estas entrevistas, inquietudes, aspectos que mejorar, aspectos que mantener, de manera de obtener la máxima información

respecto a sus experiencias y expectativas con la planilla. Esto, tal como se describe en [8] y [9].

### 3. **Recolección de conceptos previos.**

Junto a los desarrolladores se aclararán y definirán los conceptos acerca de la planilla que deberían estar claros. Además se plantean ideas innovadoras que se podrían agregar.

### 4. **Revisar el dominio del problema.**

Se evalúan las entrevistas de usuarios, se recopila la información más importante y se redefinen aspectos que no necesiten cambiar o que no estén muy claros.

### 5. **Definir roles de usuario y modelos de usuario.**

Definir cuáles son los distintos tipos de usuario. Definir los objetivos, rutinas de cada tipo de usuario y escribir cada uno de ellos en tarjetas. Revolver estas tarjetas para no perder objetividad y priorizar los roles de usuario más importantes.

Luego se deben agrupar los roles que sean similares, y discutir la relación que existe entre cada grupo. Con esto se obtendrá un modelo de usuarios.

Un rol de usuario es como un perfil que toma un usuario según la tarea que esté realizando. Por ejemplo, el mismo usuario puede tomar el rol de Planificador cuando planifique, y Tracker cuando llene el Track.

### 6. **Definir tareas y modelos de tareas.**

Extraer qué tareas son las que realiza cada tipo de usuario para cumplir sus objetivos. Escribirlas en tarjetas. Revolverlas y luego priorizarlas basados en importancia y frecuencia. Las tareas más importantes serán llamadas *tareas focales*.

Agrupar las tareas que sean similares y discutir la relación que existe entre cada grupo. Con ello se obtiene un modelo de tareas.

En este punto se pueden definir nuevas tareas que actualmente no se efectúan de manera aumentar la funcionalidad y usabilidad de la aplicación.

**7. Definir contextos de interacción.**

Para cada grupo de tareas obtenido en el paso anterior, definirlo como un contexto de interacción.

**8. Definir tareas de usuario.**

Para cada tarea dentro de un contexto de interacción escribir un Caso de Tarea, similares a los casos de uso. Estos casos de tarea describen los pasos a seguir que se efectúan al momento de realizar una tarea.

**9. Crear una interfaz de usuario “wireframe”.**

Con lápiz y papel construir prototipos fieles a una versión digital de cada contexto de interacción.

**10. Testear los contextos de interacción.**

Para cada Caso de Tarea probarlo con usuarios finales y emular una interacción real. Se presenta el prototipo en papel y se van cambiando las vistas a medidas que el usuario va interactuando y navegando por el prototipo ([8] y [9]). El evaluador no debe inferir en las acciones del usuario, solo debe escribir las acciones de usuario y las inquietudes que se le presenten. Se presentarán pautas de usuario final para ser contestadas por los participantes.

**11. Evaluación y rediseño del prototipo.**

Evaluar el testeo de los contextos de interacción y rediseñar según corresponda. Volver al paso 9 de ser necesario una nueva validación de parte de los usuarios.

#### **12. Construcción de versión digital.**

A partir de los prototipos en papel y la interacción de parte de los usuarios, construir una versión digital del producto.

#### **13. Validar con usuarios finales.**

Validar el producto digital con usuarios finales, ojala que sean distintos a los usados con la versión en papel para no perder objetividad. Se presentarán pautas de usuario final para ser contestadas por los participantes.

#### **14. Evaluación de aplicación.**

En este paso se evalúa la usabilidad de la aplicación usando pautas de usuario final.

### **6.3. Sesiones de diseño**

#### **6.3.1. Identificar a los participantes**

Para la realización de este proyecto se contó con la colaboración de personas clasificadas entre Usuarios comunes y Usuarios expertos. Los Usuarios comunes son los que la usaron en el curso de Proyecto de Software y no han seguido con su uso. Los Usuarios expertos son los que la siguen usando hasta el día de hoy, y que además han hecho colaboraciones a ésta. Los usuarios son los siguientes:

Usuarios comunes:

- Cristina Melo
- Francisco Gutiérrez
- Gustavo Garcia
- Ignacio Ortega

- Víctor Ramiro
- Mauricio Farah
- Julio Villane
- Alan Shomaly

Usuarios expertos:

- Agustín Villena
- Rodrigo Ojeda
- Roberto Carrasco

Para el proceso de diseño se necesitó de un grupo de trabajo el cual estuvo compuesto por los usuarios expertos nombrados anteriormente.

### 6.3.2. Entrevistas a usuarios

Se entrevistaron a ocho personas, de las cuales tres eran usuarios expertos y el resto inexpertos. Las entrevistas se realizaron personalmente, anotando lo que decían en una planilla.

A continuación se muestra una tabla con las preguntas que se realizaron y el objetivo que tenían para el entrevistador.

Pregunta	Objetivo
<b>¿Para qué sirve la Painless Tracking?</b>	<ul style="list-style-type: none"> <li>• Hacer que el usuario tratará de recordar su uso en caso de olvido</li> <li>• Saber si es que tienen la idea correcta acerca de lo que es la planilla de Tracking. De esa manera analizar y juzgar si es que la usaron correctamente</li> </ul>
<b>Describa un caso típico de uso. Indique detalladamente como la usa en un día</b>	<ul style="list-style-type: none"> <li>• Que el usuario piense en como la usa y así que piense también en</li> </ul>

normal de trabajo.	<p>las cosas que le molestan y gustan.</p> <ul style="list-style-type: none"> <li>• Obtener detalladamente cómo es que realizaba su acción más recurrente. Importante para generar una buena usabilidad.</li> </ul>
<b>¿Cuáles son las acciones más recurrentes que realizó?</b>	Obtener cuales son las funcionalidades que el usuario más ocupa.
<b>¿Cuáles son las acciones más intuitivas y fáciles de realizar? Indique cosas que no le gustaría cambiar.</b>	Obtener lo bueno que existe actualmente en la planilla Excel
<b>Si es que usó la painless, explique cuáles fueron las funcionalidades que le llamaron la atención. Indique si es que haya algo que le haya llamado la atención la primera vez que usó la planilla</b>	Obtener que cosas fueron las que gatillaron el contenido en el usuario por primera vez. Tratar de no perder eso.
<b>¿Que le gustaría mejorar?</b>	Obtener nuevas funcionalidades que no existen actualmente que le gustaría agregar.
<b>¿Cuáles son las principales críticas que le encuentra a la planilla?</b>	Obtener cosas que deben cambiar para un mejor uso.
<b>¿Cuáles son las acciones más tediosas de la planilla?</b>	Obtener las acciones que tiene la planilla que son más conflictivas con el usuario. Estas deben ser hechas de manera de minimizar esto.
<b>Si es que no uso la painless, diga porque</b>	Si es que es un usuario inexperto que haya tomado el curso CC61A y no haya usado la planilla. Con esto se puede obtener como captar usuarios nuevos que

	no la hayan usado.
<b>¿Cuáles son las dudas que tiene acerca de la planilla?</b>	Obtener cuales son las funcionalidades de la planilla que necesitan de mejor explicación o detalle.
<b>¿Usó el Plan?</b>	Obtener que tipo de usuario es.
<b>¿Usó el Tracking?</b>	Obtener que tipo de usuario es.

**Tabla 2 Preguntas y objetivos de encuestas.**

De los ocho entrevistados se hicieron cuatro encuestas personales y dos grupales, es decir dos entrevistas entrevistando a dos personas al mismo tiempo.

### **Resultados**

Los entrevistados tenían muchos puntos en común y pocos en contra. Sintetizando la información recolectada se sacaron las siguientes conclusiones.

Usuarios comunes

*Uso*

*Plan*

- La planificación la hace un usuario, el resto hace tracking.
- La planificación se realiza definiendo las funcionalidades más grandes y luego subdividiendo cada una de ellas en pequeñas tareas.
- Con respecto a la periodicidad de la planificación se obtuvieron resultados entre una vez por semana, hasta al principio de cada iteración.

*Tracking*

- Las tareas más recurrentes son el tracking y ver cuantas horas a la semana se trabajó.
- El caso de uso típico es, al final del día, buscar el id de la tarea que se trabajo y llenar con ese id el tracking o viceversa: llenar lo que se hizo y buscar el id.

## *Mejoras*

### *Plan*

- Que las tareas se ordenaran por producto al momento de agregar una nueva en el Plan.
- Se deberían priorizar las tareas nuevamente dentro del grupo de trabajo. La prioridad que existe actualmente es la que define el cliente y no la que definen los desarrolladores.

### *Tracking*

- Debería existir que un usuario tenga “tareas favoritas” de manera que al momento de hacer tracking, uno tenga la visibilidad de éstas solamente y elija de cual está haciendo el Track.
- Debería haber un autocompletar para elegir la tarea del tracking
- Un usuario dijo que al hacer el tracking se debería autocompletar con lo que se hizo con la descripción de la tarea, esto debido a que siempre trabajaban con tareas bien granuladas y lo que decía la tarea era exactamente lo que se hizo. También le gustaría solamente elegir que tareas trabajo y escribir solo los tiempos.

### *General*

- La planilla debería ser colaborativa mediante una página Web.
- Debería haber un log de modificaciones.
- Debería haber un lugar donde dejar documentación asociada al proyecto. Por ejemplo cuando alguien tiene que investigar sobre algo, sería bueno dejarla en alguna parte para que todos la vean.
- Deberían haber perfiles de usuario.

## *Críticas*

### *Plan*

- Al re planificar las nuevas tareas quedaban al final y no ordenadas.

- Costaba redefinir tareas, re planificar significaba escribir todo de nuevo.

#### *Tracking*

- Lo más tedioso es recuperar el id de la tarea asociada al tracking que estaban realizando. Para ello se debe mirar en la pestaña de planificación y buscar el id.
- No se tiene visibilidad de cuáles son “mis tareas”.

#### *General*

- La planilla resta mucho tiempo porque debe ser escrita de a uno.
- Las macros se desconfiguraban.
- Las fechas se desordenaban.
- Las métricas no se usaron porque no eran lo suficientemente auto-explicativas. Estaban enfocadas a “Administradores de proyectos”.

#### *Comentarios*

- Lo que más llamo la atención la primera vez que se usó la planilla fueron las métricas. A pesar de ser la funcionalidad menos usada.
- La duda más recurrente fue, ¿que significaban las métricas?. Para los usuarios éstas no aportaban valor.
- Lo que más aportó valor al proyecto fue ver el tracking y las tareas pendientes por hacer.
- La funcionalidad más usada es el tracking.

#### Usuarios Expertos

##### *Uso*

##### *Plan*

- Si es necesario se replanifica después de hacer tracking.
- Revisar en base a lo planificado que es lo que se va a hacer.

### *Tracking*

- Entre dos personas se mira un reporte de avances, se mira el tracking y se va llenando el tracking recuperando el id de la tarea correspondiente al track que se está realizando.
- Un usuario registra su tracking del día anterior, escribe todo lo que hizo y luego mira el plan para tratar de asociar lo que hizo con las tareas existentes.

### *Mejoras*

#### *Plan*

- Ver cuáles son las tareas en las que es responsable el usuario.

#### *Tracking*

#### *General*

- Tener un buen reporte de lo no planificado
- Tener un diagrama de flujo acumulado y la métrica de velocidad del grupo de trabajo.
- Tener una vista multi-proyecto
- Saber cuánto he invertido en generar una funcionalidad cerrada.

### *Críticas*

#### *Plan*

- Tener que duplicar información para agregar cosas, por ejemplo cuando se agregan tareas se duplican funcionalidades, es decir se debe escribir nuevamente la funcionalidad asociada.

#### *Tracking*

- Lo más tedioso fue buscar el id de la tarea en el Plan cuando se está haciendo tracking.
- Al hacer tracking, tener ahí mismo visibilidad de cuales son "mis tareas".

### *General*

- Es mono usuario.
- Están mezclados los datos con el código.
- El modelo de datos son simplificaciones, le falta expresividad, es decir, se podrían guardar muchos más datos que aporten valor.
- Tener las funcionalidades: Plan, Tracking, Métricas, etc., todas juntas. La planilla tiene muchas pestañas y el usuario no sabe donde pinchar.

### *Comentarios*

- Una de las acciones más intuitivas son la reorganización de las tareas y funcionalidades (modelo mental de Excel), y la movilidad.
- Otra de las acciones más fáciles de usar es agregar nuevas métricas y reportes.
- Otra de las acciones más fáciles de usar son los filtros de Excel.
- Lo que más llamó la atención al usarla la primera vez fue que mostraba con colores que era lo que faltaba por hacer, y además saber cuánto se ocupó de más al realizar una tarea. Además decir lo que se va a hacer y cuanto se va a demorar.
- Las acciones más recurrentes son llenar el Tracking y Planificar.
- Las acciones más tediosas son llenar el Tracking por el tema del id de la tarea
- Otra acción tediosa es la re planificación porque hay que estar agregando y quitando filas del Plan.

### **Desafíos**

El mayor desafío fue que los usuarios aceptaran realizar la entrevista, ya que tenían problemas de tiempo. Hubo muchos usuarios que no se pudo entrevistar porque no tenían el tiempo.

Otro desafío fue tratar de obtener de los usuarios nuevas funcionalidades porque al preguntar qué es lo que les gustaría mejorar, se basaban siempre en la planilla Excel y no en un software totalmente nuevo que pudiera brindarles muchas más

funcionalidades. Trataban siempre de cómo mejorar lo actual y no en qué cosas nuevas se podrían agregar.

### **Conclusiones**

En general, al entrevistar a los usuarios inexpertos, se observó que estos tenían una mala imagen de lo que era la Painless Tracking, ya sea porque no les gustaba o porque habían tenido malas experiencias con ella. En general los usuarios lo que más querían declarar eran las críticas que encontraban de la planilla.

Entrevistar a dos personas al mismo tiempo pareció la mejor forma ya que a pesar de que se pueda pensar que uno de ellos puede influir en las respuestas del otro, ocurrió que las respuestas de un entrevistado expandieron las respuestas del otro, por ejemplo, haciendo recordar de otros elementos que le molestaba o que le gustaría mejorar, obteniendo así un mayor feedback.

Con respecto a los resultados, el uso de las métricas se ve fuertemente inclinado a los usuarios expertos. Los usuarios inexpertos no usaron las métricas porque no les aportaba valor, solo debían cumplir con el tracking y ver que tareas faltaban por hacer. La acción fuertemente más recurrente corresponde al Tracking y la cual consiste en buscar el id de la tarea para completar un registro.

#### **6.3.3. Recolección de conceptos previos**

Para este paso se contó con la ayuda del grupo de trabajo o usuarios expertos. En esta sesión se espera analizar las entrevistas de usuario y aclarar conceptos de la Painless Tracking que no estén claros o de los que haya inconsistencias. Además se trata de obtener la mayor cantidad de ideas innovadoras para agregar en la nueva versión de la planilla.



**Ilustración 6-1 Taller de diseño**

Se aclararon casos extremos, como el caso de que es lo que pasaba con los tiempos invertidos en una tarea cancelada. La tarea cancelada desaparece del plan pero no se elimina y todo el tracking realizado hasta el momento hacia esa tarea queda intacto. Las horas que fueron destinadas a esas tareas quedan asociadas a esa tarea cancelada y no se distribuyen a otras tareas como se podría haber pensado.

Se leyeron rápidamente las entrevistas y se sacaron ideas generales que podrían no haber estado en consideración. No se terminaron de analizar todas las entrevistas porque quitaban mucho tiempo y no aportaban mucho a los pasos que se iban a tratar en el “taller”.

Se explicó al grupo la metodología de trabajo que se iba a seguir en ese “taller”. Primero se definirán roles y grupos de usuario, luego las tareas y los contextos de interacción.

A continuación se abrió la posibilidad de que cada uno de los integrantes aportara con ideas o funcionalidades novedosas o útiles, que podría adquirir la aplicación a construir. Nacieron muchas ideas las cuales iban siendo anotadas en PostIts y pegadas en un paleógrafo. Se acogió cada una de las ideas y no se discriminó ninguna, de hecho si se proponían ideas que eran parecidas unas a las otras se escribían igual. Este proceso fue rápido y muy dinámico.

Una vez anotadas las funcionalidades novedosas, estas fueron priorizadas en orden del valor que podrían aportar a los usuarios, y en la dificultad y tiempo requerido para implementar cada una de estas. Se dejaron las más importantes.

Ideas que se dejaron por ejemplo fueron:

- Comentarios sobre todo. Poder comentar sobre cualquier elemento del sistema.
- Avances de conocimientos o WIKI.
- Sistema de mensajería privada
- Undo / Redo de acciones
- Versión imprimible de reporte de avances
- Interfaz de avance que sea la interfaz operativa
- Historia de una tarea.

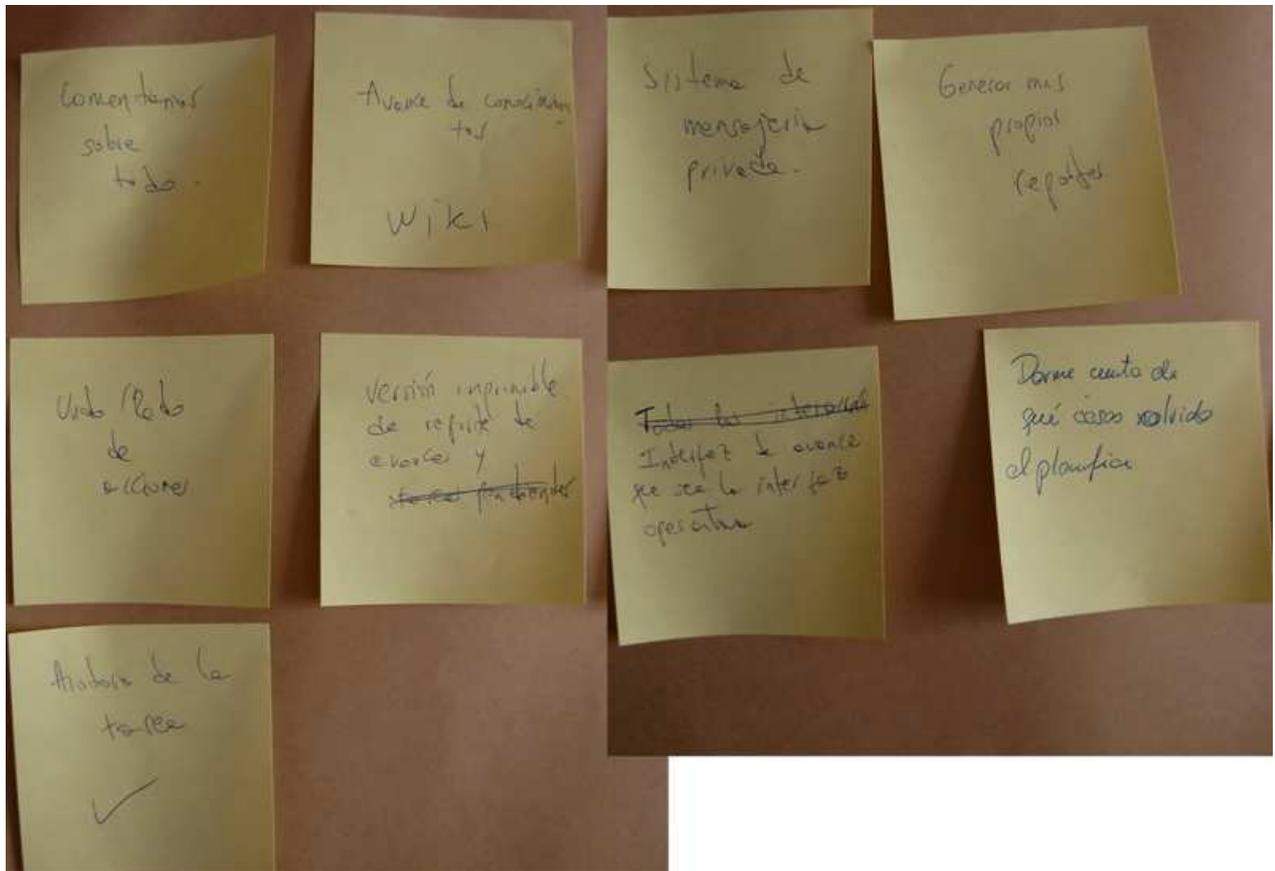
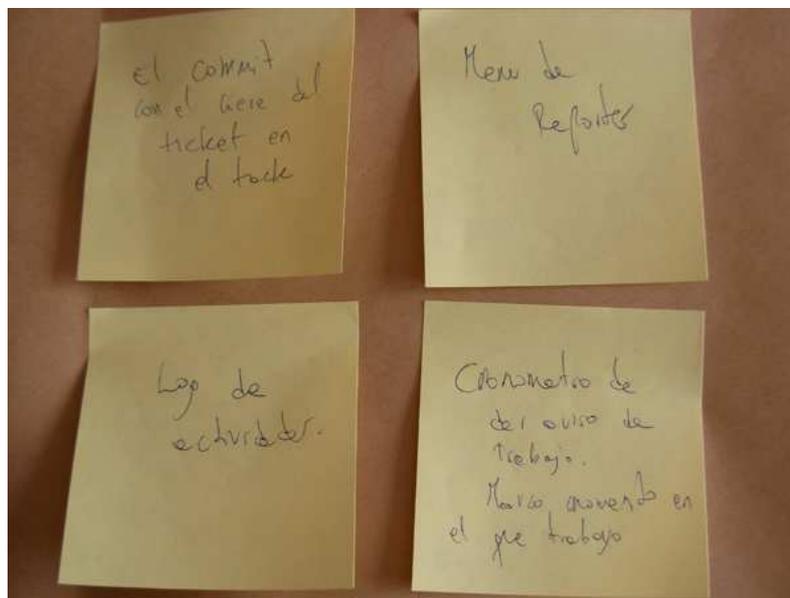


Ilustración 6-2 Ideas que fueron aceptadas

Ideas que fueron rechazadas:

- El comit con el cierre del ticket en el Track.
- Menu de reportes.
- Log de actividades.
- Marcar con un cronómetro en la tarea que se trabaja para calcular las horas exactas trabajadas.



**Ilustración 6-3 Ideas rechazadas**

### ***Dificultades***

Una vez aclarados los conceptos, se continuó analizando las entrevistas pero el grupo no estaba lo suficientemente cohesionado. Cada uno de los integrantes del grupo de trabajo estaba en una posición alejada del resto. Esto produjo que no existiera la química y confianza dentro del grupo para que cada uno de los integrantes se pudiera expresar a plenitud. Este “hielo” existente entre los integrantes fue roto cuando se decidió que cada uno expresara ideas innovadoras para el software, es decir, que funcionalidades les gustaría agregar a cada uno sin discriminar sus ideas. Esto produjo que cada integrante se entusiasmará en el proyecto, ya que iban a ser consideradas sus ideas dentro de este, lo que produjo una mayor participación y dinamismo que hacía falta.

### ***Conclusiones***

Lo que hizo falta en un comienzo fue romper el hielo para tener trabajo mucho más dinámico y participativo. Este paso fue un éxito y se obtuvieron muchas ideas. Hay que recalcar que hubo ideas que iban surgiendo en los siguientes pasos y por ende se debió volver a este paso para agregarlas.

#### 6.3.4. Definir roles de usuario y modelos de usuario

Para este paso se contó con la ayuda del grupo de trabajo o usuarios expertos. En este paso se iban escribiendo en PostIts los tipos de usuarios que existían en el software. Es importante recalcar que no es la definición de Planificador, Tracker, etc. Sino que son intenciones de usuarios, es decir, un tipo de usuario puede planificar y además usar el Tracker, un usuario puede tomar distintos roles de usuario según lo que se esté haciendo.

Una vez obtenidos los roles se revolvieron y agruparon según asociación formando grupos y luego se ordenaron los grupos de la misma forma. Cada grupo forma un modelo de usuario.

Los roles de usuarios más importantes se marcaron con asterisco y se llamaron *roles focales*.

#### ***Roles y modelos de usuario***

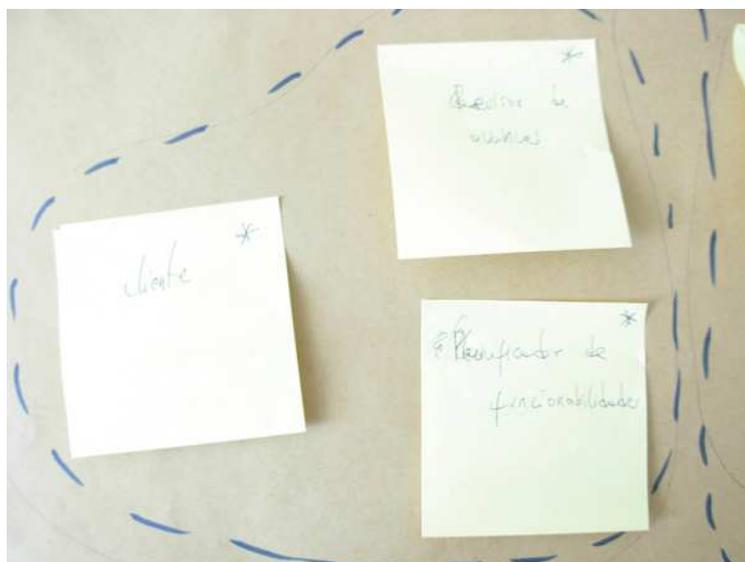
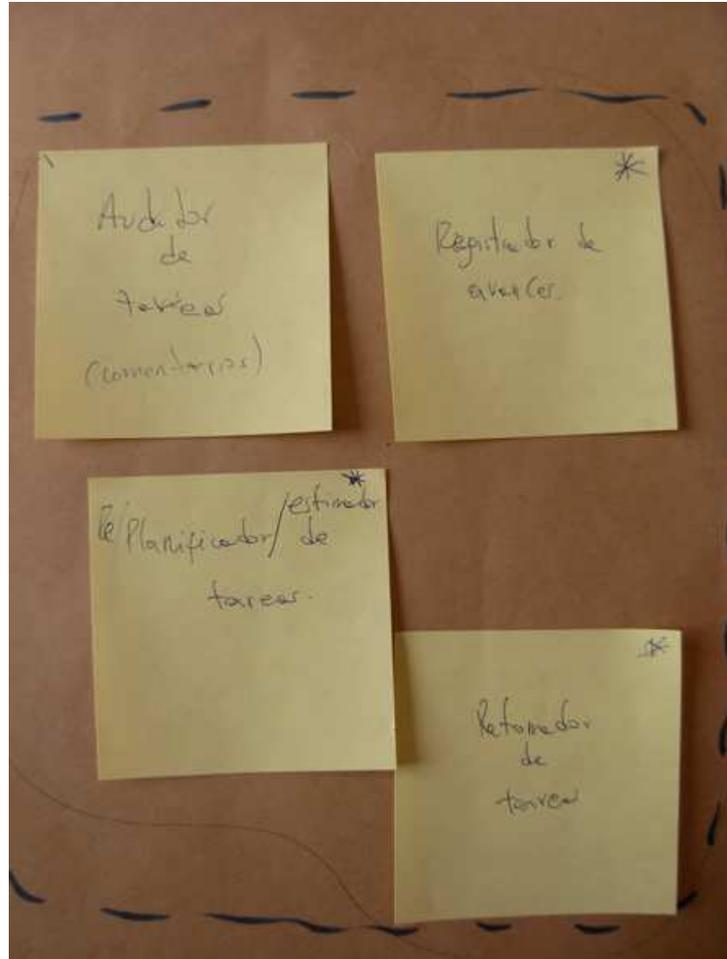


Ilustración 6-4 Modelo de usuario orientado a la gestión administrativa del proyecto

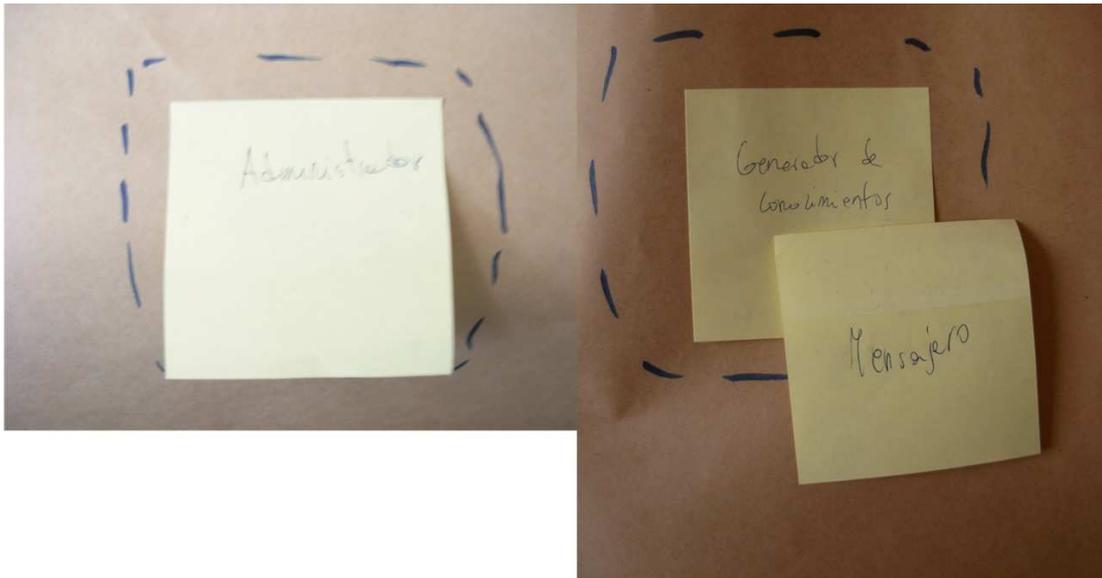
- Modelo de usuario orientado a la gestión administrativa del proyecto ( Ilustración 6-4):
  - **Ciente.** Este rol es el encargado de planificar y re planificar funcionalidades y tareas con el cliente. Surgió la idea de que el cliente pudiera tener control de las prioridades del proyecto
  - **Revisor de avances.** Este usuario es el encargado de ver cuánto le falta a una funcionalidad para estar completa, revisa las métricas para ver el avance del proyecto.
  - **Planificador de funcionalidades.** Usuario encargado de definir y re planificar funcionalidades del proyecto.



**Ilustración 6-5 Modelo de usuario orientado a la gestión operativa del proyecto**

- Modelo de usuario orientado a la gestión operativa del proyecto ( Ilustración 6-5):
  - **Auditor de tareas.** Encargado de realizar comentarios sobre tareas, funcionalidades y tracking. Además revisa estos comentarios y lo que se ha hecho en una tarea.
  - **Planificador de tareas.** Usuario encargado de definir y re planificar tareas que componen una funcionalidad.
  - **Registrador de avances.** Usuario encargado de registrar en el Tracking lo realizado durante el día.

- **Retomador de tareas.** Usuario que revisa que es lo se ha hecho últimamente dentro de una tarea para poder retomarla y trabajar el resto del día



**Ilustración 6-6** A la izquierda, modelo de usuario orientado a la administración. A la derecha, modelo de usuario orientado a la comunicación.

- Modelo de usuario orientado a la administración ( Ilustración 6-6)
  - **Administrador.** Administrador del software a realizar.
- Modelo de usuario orientado a la comunicación ( Ilustración 6-6)
  - **Generador de conocimientos.** Usuario que ve y crea documentos de información para el desarrollo del proyecto (Al estilo wiki).
  - **Mensajero.** Encargado de ver y mandar mensajes privados a otros usuarios.

### **Tabla resumen**

<b>Rol de usuario</b>	<b>Rol focal</b>
<b>Cliente</b>	<b>X</b>
<b>Revisor de avances</b>	<b>X</b>
<b>Auditor de tareas</b>	
<b>Planificador de funcionalidades</b>	<b>X</b>
<b>Planificador de tareas</b>	<b>X</b>
<b>Registrador de avances</b>	<b>X</b>
<b>Retomador de tareas</b>	<b>X</b>
<b>Administrador</b>	
<b>Generador de conocimientos</b>	
<b>Mensajero</b>	

**Tabla 3** Tabla con los roles de usuarios y con X los que son focales.

### ***Dificultades***

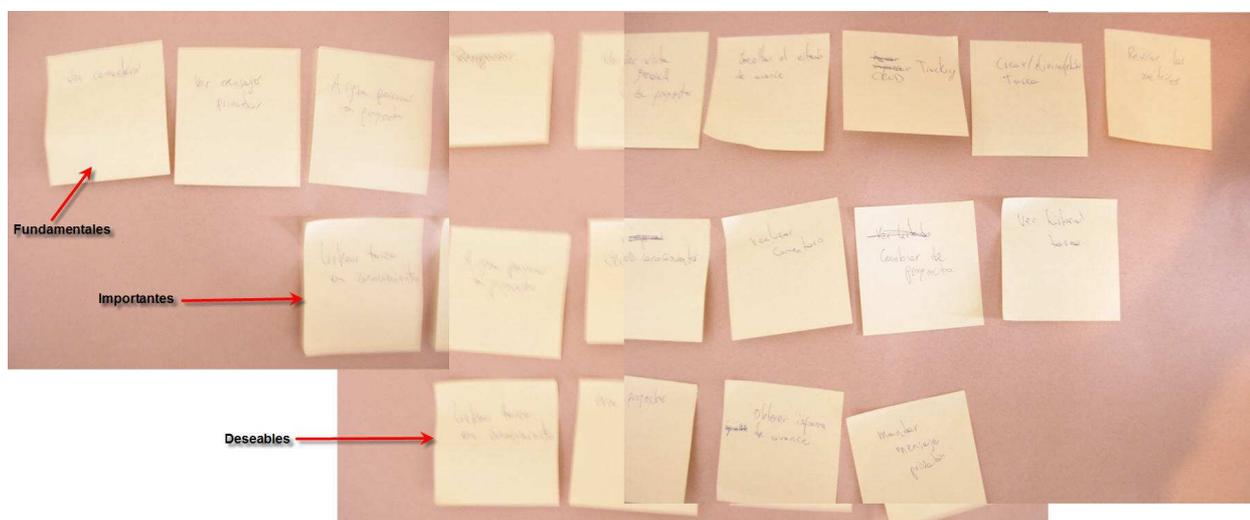
Al comienzo se tuvieron muchas dificultades al definir que eran los roles de usuarios. El grupo partió muy perdido al comienzo, pero de a poco fue concretando más las ideas. Para ello se tuvieron que repasar muchas veces los conceptos para definir bien los roles. Cada vez que surgían intenciones de usuarios se chequeaba si es que se había definido o no y muchas veces se definieron roles dos veces según el punto de vista que se miraba.

Se tuvieron que agregar nuevos roles de usuario a medida que iban surgiendo nuevas ideas o funcionalidades nuevas que aportar al sistema del paso anterior.

### 6.3.5. Definir tareas, modelos de tareas y contextos de interacción

Para este paso se contó con la ayuda del grupo de trabajo o usuarios expertos. Para cada rol de usuario se definieron tareas que realizaban, si había otras tareas que faltaba agregar, se agregaban.

Una vez obtenidas las tareas, se revolvieron y luego ordenaron por orden de importancia. Esto generó tres filas, que se podrían llamar fundamentales, importantes y deseables como se muestra en la figura ( Ilustración 6-7).



**Ilustración 6-7 Vista general las tareas ordenadas en fundamentales (arriba), importantes (al medio) y deseables (abajo)**

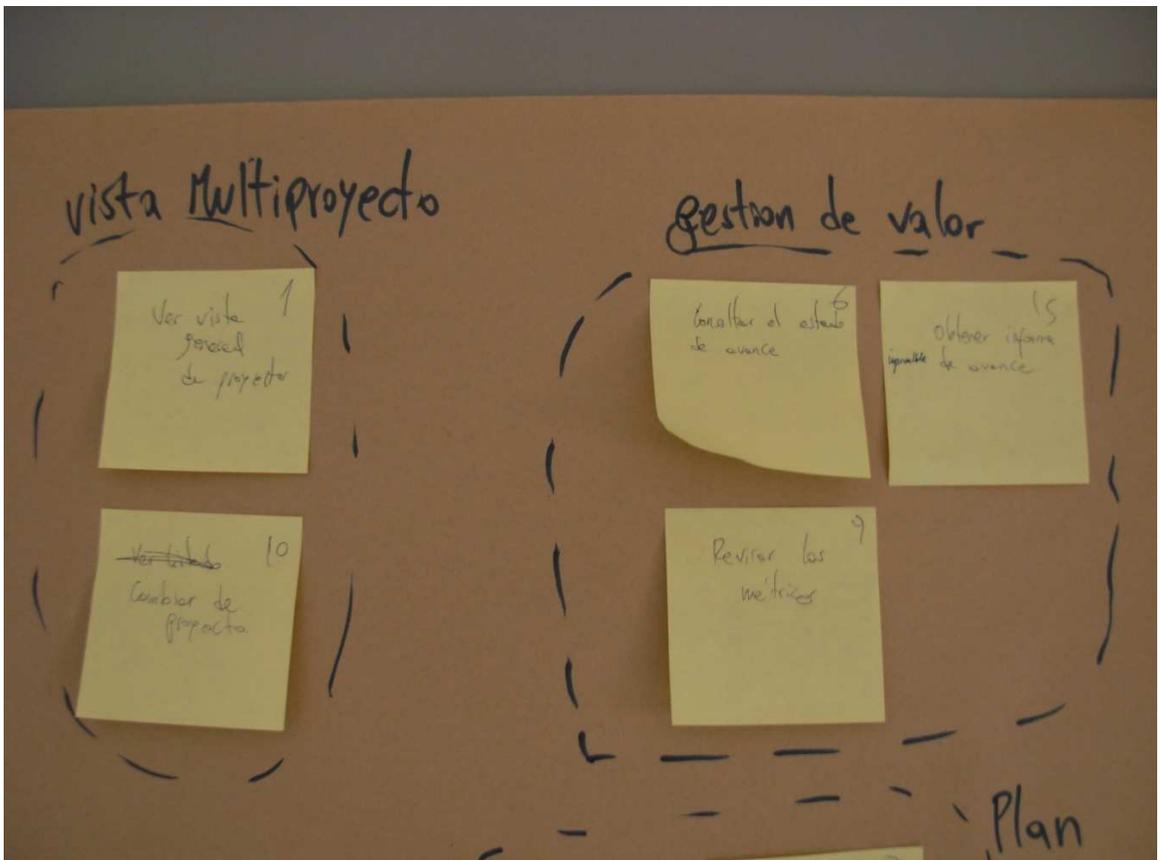
Luego cada fila de importancia se ordenó según el orden de frecuencia que eran o podrían ser ejecutadas por los usuarios. Esto generó una sola columna de tareas ordenadas por importancia y luego por frecuencia. Se decidió enumerar cada una de las tareas con el número uno para la más importante y frecuente, y N para la N-ésima menos importante y menos frecuente, de manera que en el paso siguiente no se perdiera este orden.

Luego, las tareas fueron organizadas en grupos según similitud y cada grupo se posicionó cerca de los grupos que tenían más asociación. Con esto se obtuvieron los modelos de tareas.

Solo bastaba ponerle nombre a cada grupo para obtener contextos de interacción.

### **Contextos de interacción**

A continuación se detallan los contextos de interacción obtenidos en este paso y las tareas que los componen. Entre paréntesis indica la posición de la tarea dentro del orden por importancia y frecuencia. Las tareas que no salen en las imágenes son las que se agregaron después del taller.



**Ilustración 6-8 Contextos de interacción para Vista Multiproyecto y Gestión de Valor**

- **Vista Multiproyecto (Ilustración 6-8)**
  - **Ver vista general del proyecto (1).** Esta tarea me permite ver en todos los proyectos en los que pertenezco mostrando una visión general de ellos.

- **Cambiar de proyecto (10).** El usuario puede pertenecer a varios proyectos, con esta tarea el usuario se cambia de un proyecto a otro.
- **Gestión de Valor (Ilustración 6-8)**
  - **Consultar el estado de avance (6).** Tarea que se efectúa para ver cuánto se lleva completado, cuanto falta por hacer, etc.
  - **Obtener informe imprimible de avance (15).** Tarea que busca una vista imprimible de los avances del proyecto. Orientado para mostrar a los clientes.
  - **Revisar las métricas (9).** Revisar las métricas del proyecto.
  - **Exportar datos a csv ( ).** Exportar los datos a csv. Esto permitirá generar nuevos reportes y métricas.

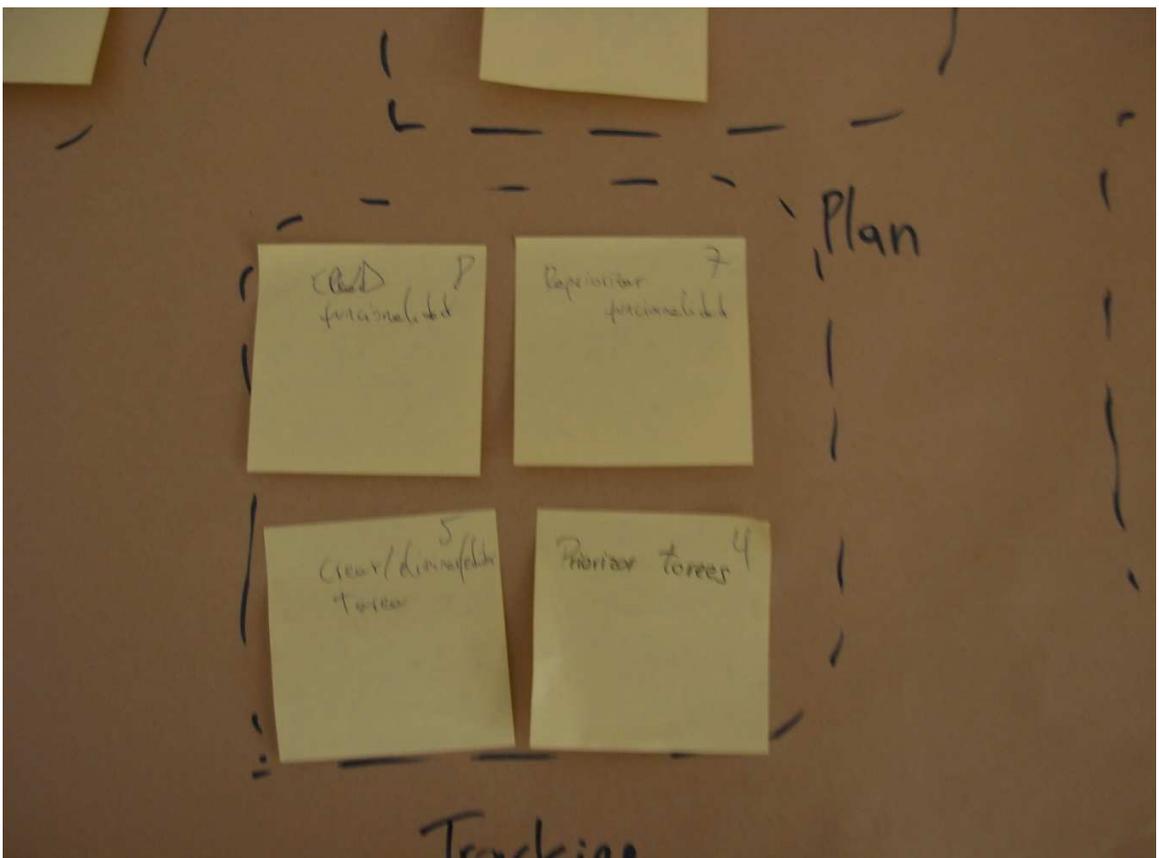


Ilustración 6-9 Contextos de interacción para el Plan

- **Plan (Ilustración 6-9)**
  - **CRUD funcionalidad (8).** Crear, leer, modificar y eliminar funcionalidades.
  - **CRUD tareas (5).** Crear, leer, modificar y eliminar tareas.
  - **Priorizar funcionalidad (7).** Darle otra importancia o prioridad a una funcionalidad.
  - **Priorizar tarea (4).** Darle otra importancia o prioridad a una tarea.

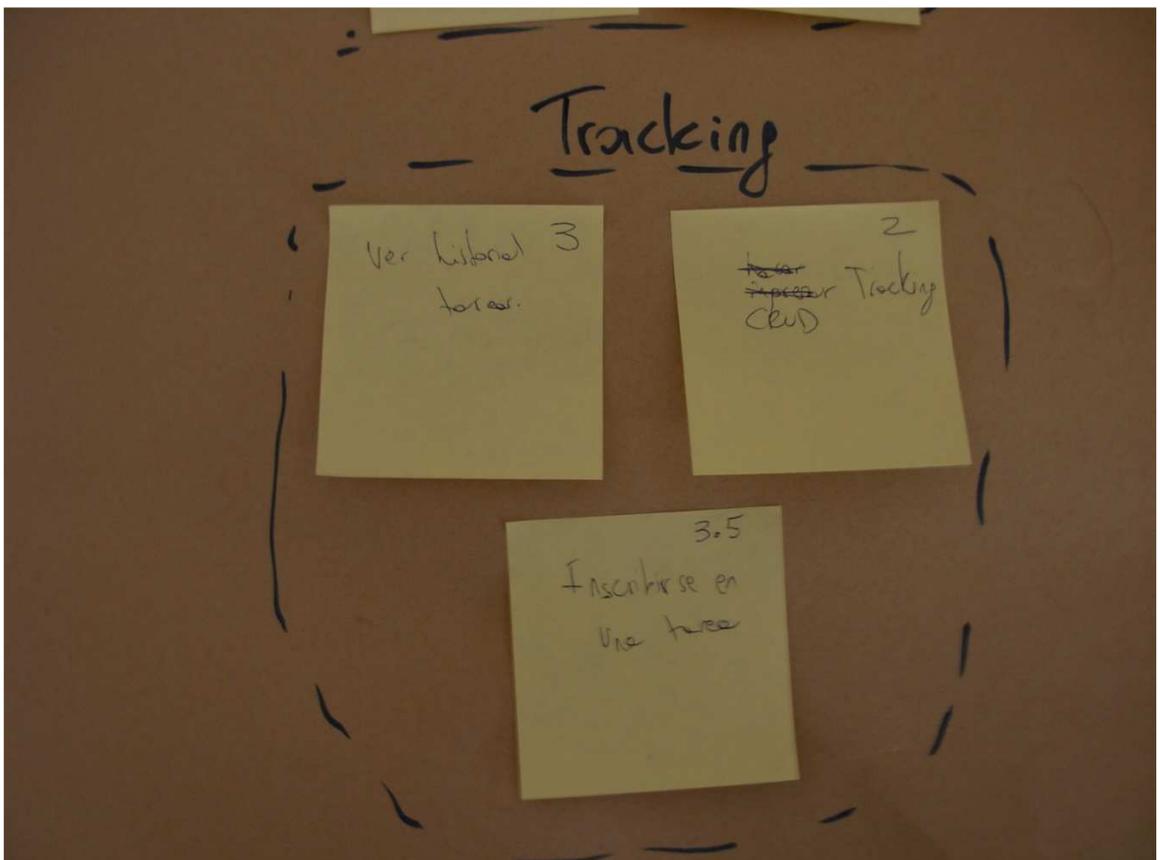


Ilustración 6-10 Contextos de interacción para el Tracking

- **Tracking (¡Error! No se encuentra el origen de la referencia.)**
  - **Ver historial tareas (3).** Ver lo que se ha hecho históricamente dentro de una tarea.
  - **Buscar tarea (3.2).** Buscar una tarea.

- **Inscribirse en una tarea (3.5).** Agregar tarea a “mis tareas” de manera de que al hacer tracking tener visibilidad de esas tareas primero.
- **CRUD Tracking (2).** Crear, leer, modificar y eliminar registro de lo hecho.

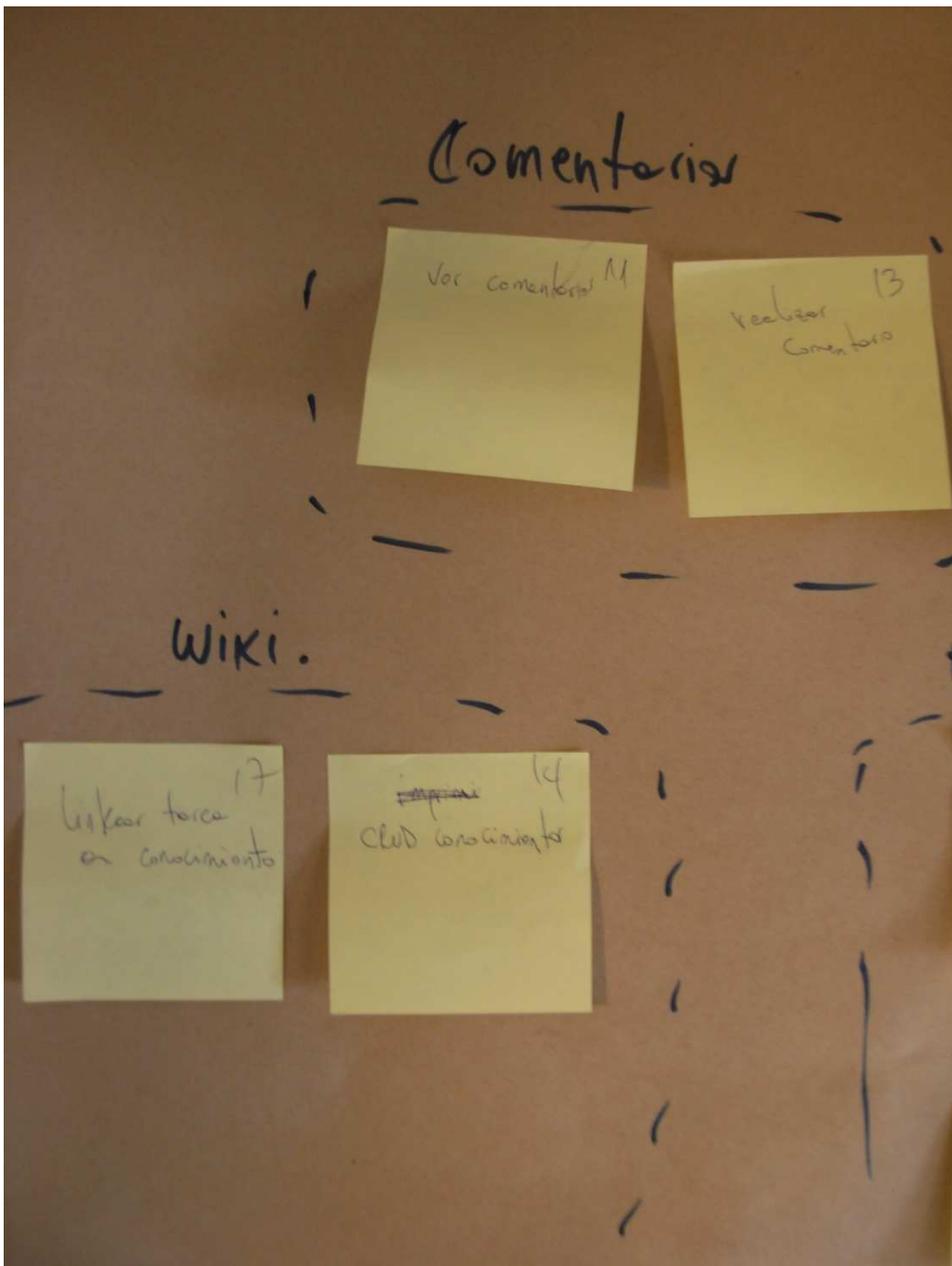


Ilustración 6-11 Contextos de interacción para los Comentarios y el Wiki

- **Comentarios** ( Ilustración 6-11)
  - **Ver comentarios (11).** Ver los comentarios hechos sobre algún elemento del software.
  - **Realizar comentario (13).** Realizar comentario sobre algún registro o tracking o sobre alguna tarea o funcionalidad.
- **Wiki** ( Ilustración 6-11)
  - **CRUD conocimiento (14).** Crear, leer, modificar y eliminar documentos de información.
  - **Enlazar tarea a conocimiento (17).** Enlazar una tarea con un documento. Por ejemplo si existe tarea “Crear usuarios en base de datos”, enlazarla con documento tutorial “Como crear usuarios en base de datos”.

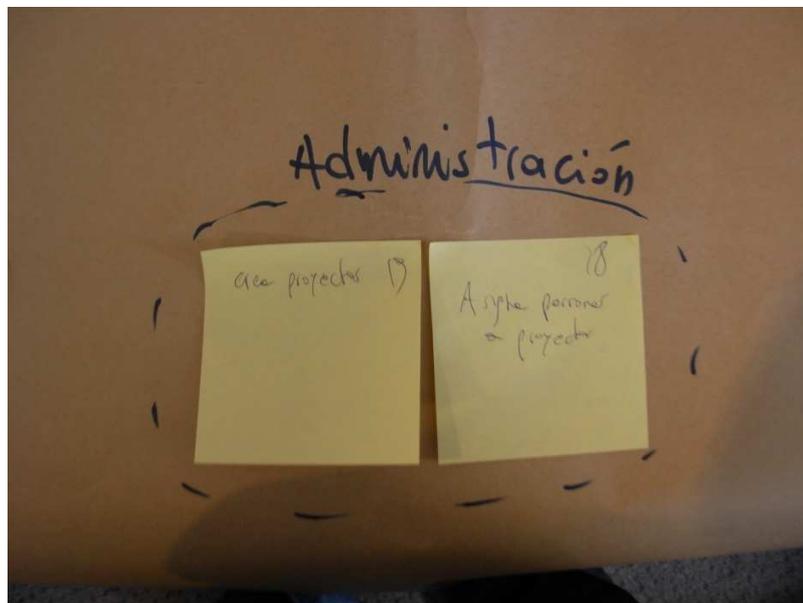
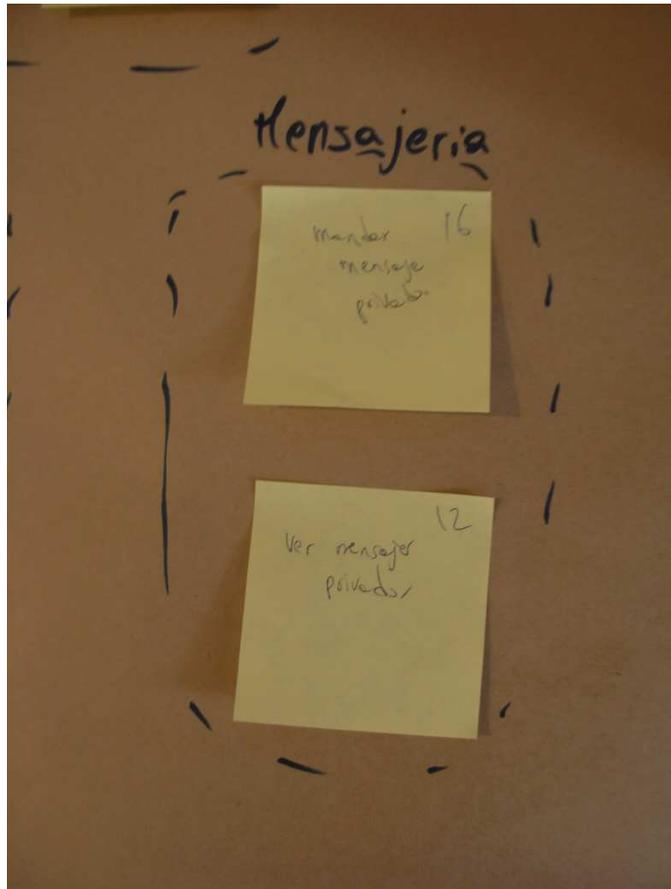


Ilustración 6-12 Contextos de interacción para la Administración

- **Administración (Ilustración 6-12)**

- **Crear proyecto (19).**
- **Asignar usuarios a proyectos (18).**



**Ilustración 6-13 Contextos de interacción para la Mensajería**

- **Mensajería (Ilustración 6-13)**
  - **Mandar mensaje privado (16).**
  - **Ver mensajes privados (12).**

### ***Dificultades***

Generalmente las tareas se repetían, ya que los roles de usuarios no eran excluyentes, por lo tanto se eliminaron varias tareas. Debido al poco tiempo que quedaba, se trató de que todos escribieran las tareas en los PostIts, lo que produjo también que se

repitieran muchas tareas al comienzo, así que finalmente se optó por que uno solo escribiera.

Se tuvieron muchas discusiones y problemas para priorizar las tareas y para ordenarlas en orden de importancia, porque cada uno de los integrantes tenía distintas visiones o predicciones del uso de la Planilla, pero esto fue conversado y se llegó a un orden que fue aceptado por todo el grupo.

También ocurrió que se definió una tarea una vez que ya se había hecho la ordenación y se tuvo que poner una numeración con decimales para posicionarla entre dos existentes.

#### **6.3.6. Definir tareas de usuarios**

En este paso se debe definir para cada tarea dentro de los contextos de interacción un caso de tarea. Este debe ser creado usando un lenguaje conversacional para que sea fácil de leer y no debe ser ni muy ambiguo ni muy detallado. A continuación se muestran los casos de tareas en orden de prioridad:

##### ***Plan***

##### **CT1.1 CRUD funcionalidad**

En la misma página de estado de avance o página principal el usuario crea una funcionalidad llenando un formulario de ingreso. Luego edita la funcionalidad haciendo clic en editar, y la elimina haciendo clic en eliminar

##### **CT1.2 CRUD tareas**

El usuario debe llenar un formulario para la creación de una tarea, y debe seleccionar una funcionalidad a la cual va a pertenecer esta tarea.

##### **CT1.3 Priorizar funcionalidad**

El usuario quiere priorizar una funcionalidad y para ello arrastra la funcionalidad que aparece en la página de estado de avance, hacia otra posición. Entre más abajo que la arrastre menos funcionalidad tiene.

#### CT1.4 Priorizar tarea

El usuario quiere priorizar una tarea y para ello arrastra la tarea y la desplaza en relación a las mismas tareas dentro de la funcionalidad.

#### CT1.5 Inscribirse en una tarea

El usuario hace clic en "inscribir tarea" para agregarla en favoritos.

### ***Tracking***

#### CT2.1 Ver historial tareas

El usuario pincha en historial y el sistema muestra los datos de la tarea, seguidos de todo el tracking que se ha hecho con esa tarea.

#### CT2.2 Buscar una tarea.

El usuario escribe lo que busca en el cuadro de búsqueda para encontrar tareas que está buscando. El resultado se despliega en la página principal

#### CT2.3 CRUD Tracking

Usuario hace clic en "Tracking" y se dirige a hacer tracking. Se le presenta un formulario a llenar para hacer el tracking. A la derecha se le presentan "mis tareas" o las tareas en las que el usuario está inscrito. El usuario hace clic en una de las tareas y se autocompleta el formulario con la información de ella.

### ***Administración***

#### CT3.1 Crear proyecto

El administrador va a la vista de administración y hace clic en 'Crear proyecto' donde debe llenar un formulario para la creación de este.

#### CT3.2 Asignar usuarios a proyectos

Una vez que el administrador selecciona un proyecto dentro de la vista de administración, agrega y modifica usuarios que pertenezcan a él agregándolos o sacándolos de una lista.

### ***Vista Multiproyecto***

#### CT4.1 Ver vista general del proyecto

El usuario abre el sistema por primera vez y ve todos los proyectos en los que está participando. Aparece un avance general de cada uno de ellos representado por un porcentaje de lo completado

#### CT4.2 Cambiar de proyecto

El usuario abre el sistema por primera vez y como pertenece a más de un proyecto, se muestra la vista general de este proyecto. El entonces hace clic en el nombre de uno de los proyectos para seleccionarlo y cambiar a la vista del proyecto. Una vez dentro de un proyecto se cambia a otro haciendo clic en "Proyectos" que lo lleva a la página anterior.

### ***Gestión de valor***

#### CT5.1 Consultar estado de avance

Una vez que el usuario eligió un proyecto, o en el caso de que tenga solamente uno, se le muestra el estado de avance del proyecto, es decir, se muestra cada proyecto con sus tareas y el estado de ellas: si se está trabajando en ella, si aún no se empieza a trabajar en ella, o si es que se terminó.

#### CT5.2 Obtener informe imprimible de avance

Una vez que el usuario ve el estado de avance, este hace clic en el botón con forma de impresora para ver e imprimir una versión "imprimible" del avance.

#### CT5.3 Revisar las métricas

El usuario hace clic en "Métricas" y lleva a la página de métricas. Ahí elige la que desee visualizar para que el sistema lo muestre.

#### CT5.4 Exportar datos a csv

El usuario en la página de métricas hace clic en "exportar datos a csv" para obtener un archivo con toda la información del proyecto en un formato .csv

## **Comentarios**

### CT6.1 Ver comentarios

El usuario hace clic en cualquier tarea o funcionalidad y ve su descripción seguida de toda la lista de comentarios que se ha hecho, al estilo Blog.

### CT6.2 Realizar comentario

El usuario hace clic en una tarea o una funcionalidad y llena el formulario de ingreso de comentario.

## **Mensajería**

### CT7.1 Mandar mensaje privado

El usuario va a la sección mensajería y selecciona “mandar mensaje”, llena un formulario con el destinatario y el mensaje para mandar uno.

### CT7.2 Ver mensajes privados

El usuario va a la sección mensajería y ve todos los mensajes que le han mandado. Si hace clic en uno de ellos ve el detalle de este.

## **Wiki**

### CT8.1 CRUD conocimiento

El usuario entra a la sección “Conocimiento” y ve todos los documentos que hay, hace clic en crear conocimiento y llena los formularios para crear conocimiento.

### CT8.2 Enlazar tarea a conocimiento

El usuario va a editar un conocimiento y selecciona las tareas a las que el conocimiento va a estar enlazado.

También el usuario enlaza tareas con conocimiento llenando al detalle de una tarea y haciendo clic en “enlazar conocimiento”.

### ***Dificultades***

La mayor dificultad fue que al hacer cada uno de los casos de tarea, se debía pensar en cómo el usuario iba a realizar la actividad, y por ende se pensó en la interfaz final y la organización de secciones, etc. Pero el objetivo de este paso fue escribir “que” es lo que se hacía más que el “como” en detalle, siendo general en la forma en que se describe la tarea. Por otro lado, una vez entendido esto, fue bastante fácil seguir con los casos de tarea, porque las tareas eran bien granulares, es decir, bien específicas por lo que no se necesitó entrar en mucho detalle.

Al ir escribiendo los casos de tarea, se pensó en cómo iba a ser la interfaz y realmente ayudó todo este proceso a ello. Los procesos realizados hasta el momento permitieron que las interfaces fueran mucho más fáciles de diseñar, guiando indirectamente el proceso.

#### **6.3.7. Creación de una interfaz de usuario “wireframe”**

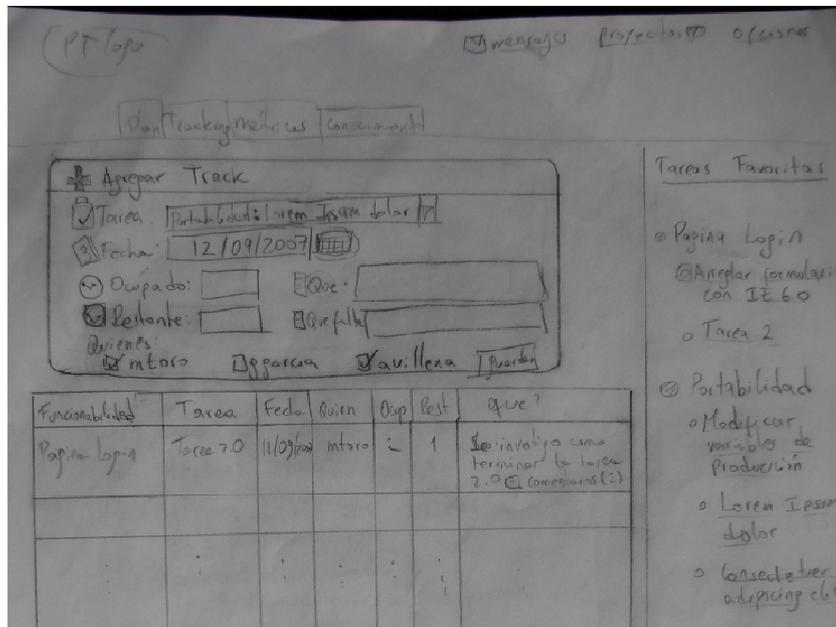
Se analizaron las entrevistas, y la agrupación obtenida al momento de la definición de contextos de interacción y tareas de usuario para obtener una primera versión del prototipo en papel del sistema. Se trabajó en más detalle en los módulos de Track y Plan ya que son fundamentales y tienen prioridad máxima.

Una vez obtenido el prototipo básico se acordó un nuevo taller de trabajo con Agustín Villena y Rodrigo Ojeda donde se debía discutir y refinar el prototipo obtenido.

En aquel taller, además de tratar con el prototipo, se propuso definir la base de datos que se utilizará en conjunto con el proyecto de Rodrigo Ojeda que trata de las métricas de la Painless Tracking.

A continuación se presentan las pantallas más importantes

## Track

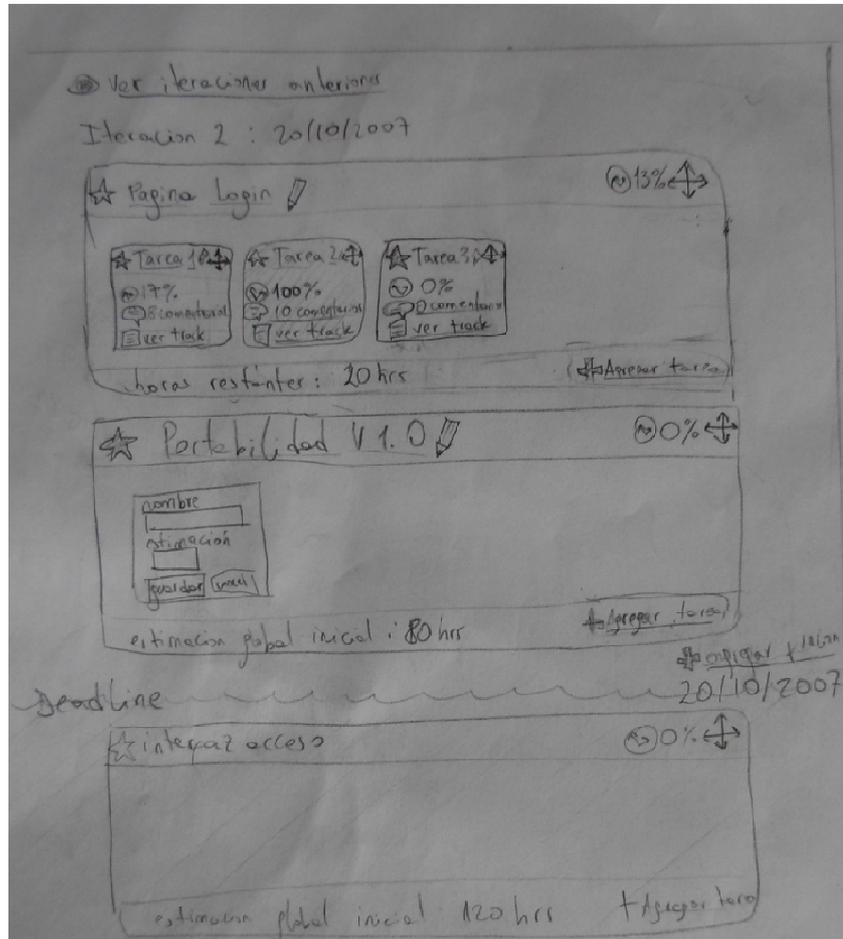


**Ilustración 6-14 Vista del componente Track del prototipo en papel**

Como muestra la figura (Ilustración 6-15), el formulario de inserción de registro en el Track está siempre visible. A la derecha va un menú contextual con las tareas favoritas, de manera de autocompletar el formulario de inserción.

Después de mostrar el prototipo básico se acordó agregar el campo "Que falta" en el Track.

## Plan



**Ilustración 6-15 Vista del componente Plan del prototipo en papel**

Como se muestra en la figura (Ilustración 6-15), cada iteración está compuesta de funcionalidades que a su vez se componen de tareas. Las tareas se representan en cajas de manera de ocupar menos espacio y no saturar tanto la vista. La estrella representa si es que la tarea es favorita o no, similar a Gmail. La flecha indica que se puede mover haciendo clic en ella. El nivel del mar que se ve, es la línea que representa hasta donde llega la punta del iceberg, o el deadline del timebox actual. Las ediciones se hacen in-place. Se propone además estimar globalmente las funcionalidades que no se tiene mucha información aún.

Después del taller se acordó cambiar los relojes que indicaban las horas ocupadas por una barra de progreso, además de poder expandir y contraer las funcionalidades.

### Detalle de tarea

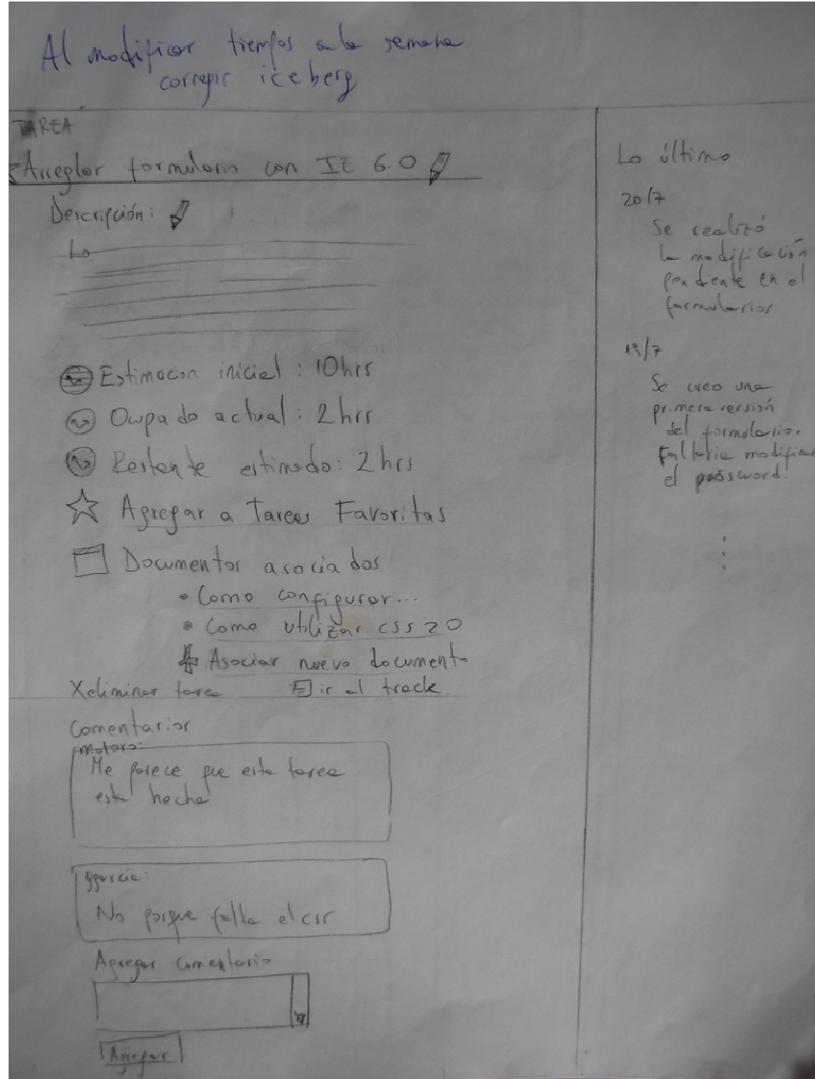


Ilustración 6-16 Vista del componente Detalle de Tarea del prototipo en papel

Como se ve en la figura (Ilustración 6-16), este módulo muestra más información acerca de una tarea, en el menú contextual de la derecha se muestra lo último que se ha registrado en el Track.

Se agrega un campo descripción que se puede editar in-place, se indica la estimación inicial, lo ocupado actualmente, el restante estimado, y las acciones a realizar: agregar la tarea a favoritos, asociar un nuevo documento, eliminar la tarea, y filtrar el Track con la información de ésta.

También está la sección de comentarios al pie de la página donde se pueden agregar tal como lo hacen los blogs.

#### Detalle de funcionalidad

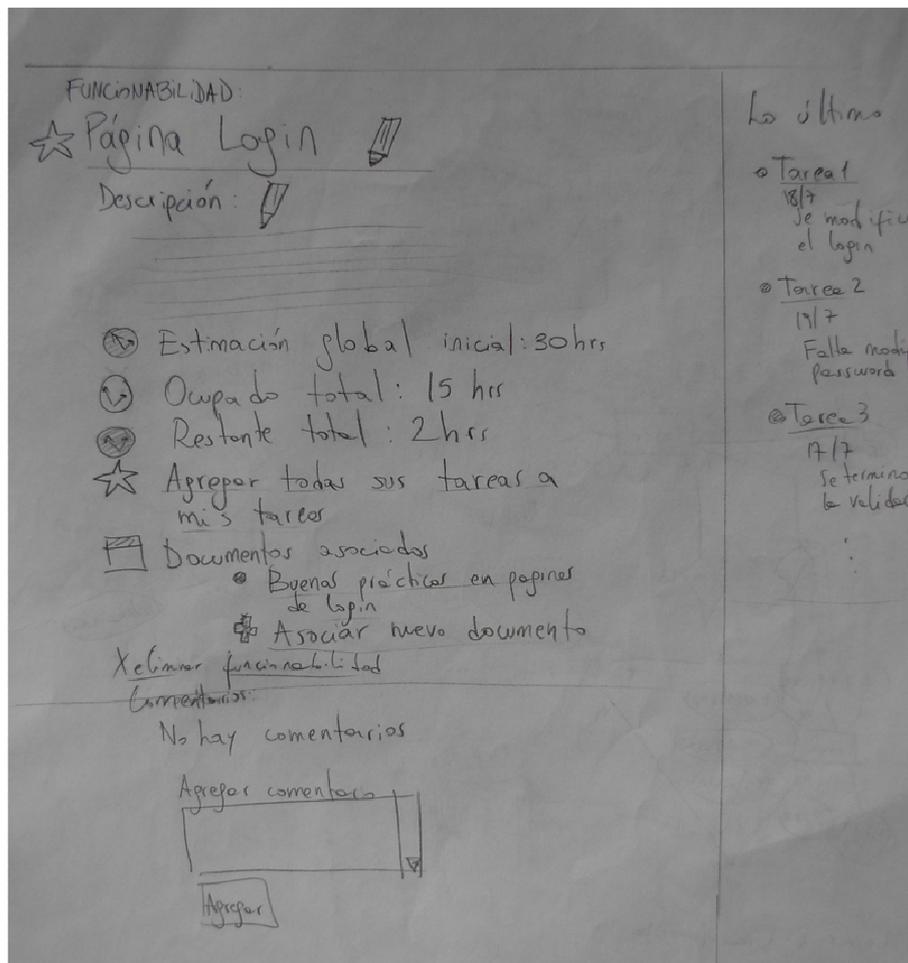


Ilustración 6-17 Vista del componente Detalle de Funcionalidad del prototipo en papel

Como se ve en la figura (Ilustración 6-17), este módulo muestra más información acerca de una funcionalidad, en el menú contextual de la derecha se muestra lo último que se ha registrado para cada tarea que la compone.

Además de la información mostrada en el Plan, se agrega un campo descripción que se puede editar in-place. Se indica la estimación inicial, lo ocupado actualmente, el restante estimado, y las acciones a realizar: agregar todas sus sub tareas a favoritos, asociar un nuevo documento, eliminar la funcionalidad, y filtrar el Track con la información de las sub tareas de ésta.

También está la sección de comentarios al pie de la página donde se pueden agregar tal como lo hacen los blogs.

Se acordó también crear una página de inicio, que muestre los proyectos a los cuales pertenece el usuario y una barra de progreso que indique cual es el avance global de éste y una página de configuración que indique cuanto trabaja cada desarrollador a la semana.

## Base de datos inicial

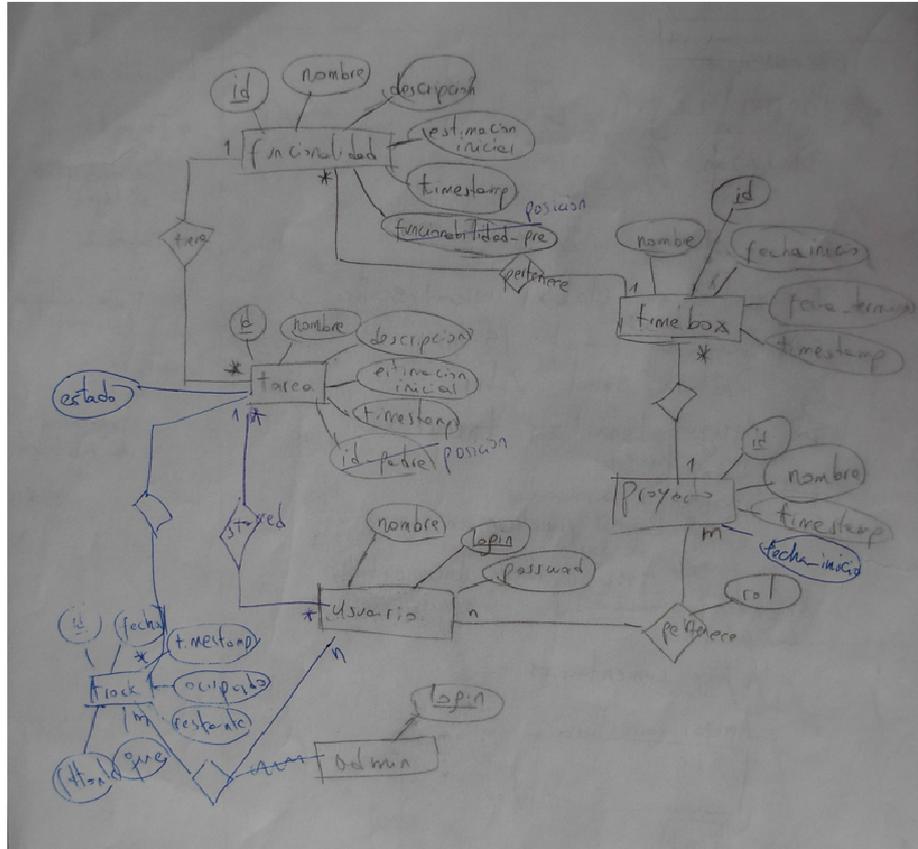


Ilustración 6-18 Base de datos inicial

### **Dificultades**

Hubo muchas discusiones respecto al prototipo debido a que cada integrante del taller esperaba que fuera de una manera distinta a la del resto, y por eso se tuvo que poner en lugar del usuario muchas veces para poder dirimir con cual opción quedarse. Otras veces se tuvo que optar por re-estructurar el prototipo completamente para que se pudieran adoptar todas las ideas propuestas. Todo esto siempre en busca de que la aplicación quedara simple y fácil de usar para los usuarios finales.

### **6.3.8. Testear los contextos de interacción / Evaluación y rediseño del prototipo**

Se probó el prototipo obtenido con usuarios, y muchos no entendieron algunos componentes, porque habían funcionalidades, como el drag-and-drop, que no podían

ser reflejados en papel fácilmente. Se tuvo dificultad con usuarios que no se acordaban cómo funcionaba la Painless Tracking por lo que se tuvo que dar un breve repaso a los conceptos y módulos del sistema. Se tomaron los comentarios y se modificó en el prototipo directamente una vez terminado con el test.

### **6.3.9. Construcción de versión digital**

#### ***Software usado***

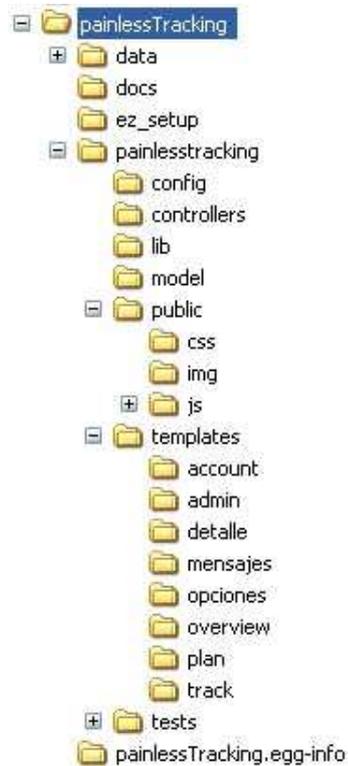
Para la implementación del proyecto se usaron las siguientes herramientas:

- **Pylons 0.93.** Framework para Python, se escogió porque es rápido el desarrollo y porque dentro de los que usan Python es el de mejor control (otros como Django son de muy alto nivel)
- **Sqlalchemy 0.35.** Librería Python para manipular tablas de una base datos orientado a objetos.
- **Mysql 4.** Solo por comodidad ya que Pylons + Sqlalchemy trabajan de la misma manera ya sea con Mysql, Postgres, Sqlite, etc.
- **Jquery.** Librería javascript para manipular documentos Html, manejar eventos, realizar animaciones y agregar interacciones AJAX. Se escogió esta librería porque es más liviana que su competencia (Prototype). Además porque utiliza una sentencia de XPATH para seleccionar objetos que es mucho más potente que el resto.
- **Interface.** Plugin de Jquery para realizar animaciones y la funcionalidad drag-and-drop. Se probaron los drag-and-drop con otros plugins y librerías y esta fue la que mejor funcionó, debido a rapidez y fácil manejo.

#### ***Arquitectura de la aplicación***

Se utilizó una arquitectura que es estándar de pylons que sigue el concepto MVC (model-view-controller). Esta arquitectura separa lo que es el modelo de la base de datos, con las vistas o interfaces de la aplicación y con la lógica o controlador de esta.

La estructura general de la aplicación es la siguiente:



**Ilustración 6-19 Arquitectura de archivos de la aplicación**

#### Modelo de datos

Este se encuentra definido en la carpeta *model/\_\_init\_\_.py*, acá se encuentra la definición de foreign keys y el mapping entre tablas y clases. Además se definen funciones especiales para cada clase.

#### Controladores

En la carpeta *controllers* se encuentra toda la lógica de la aplicación, donde cada archivo representa una página web o módulo del sistema. A grandes rasgos los controladores existentes son los siguientes:

- **Plan.** Controlador encargado de la lógica que tiene que ver con la página donde se planifica.
- **Track.** Controlador encargado de la lógica del track.

- **Account.** Controlador encargado del manejo de autenticación de usuarios.
- **Admin.** Controlador encargado de administración de proyectos y usuarios.
- **Opciones.** Controlador encargado de la lógica de la página para modificar los tiempos que trabaja cada desarrollador.
- **Detalle.** Encargado de las páginas de detalles y comentarios para funcionalidades y tareas.
- **Overview.** Encargado de mostrar la lista de proyectos a los que pertenece el usuario.
- **Mensajes.** Encargado de la lógica de los mensajes inter usuarios.

#### Templates o vistas

Acá se encuentran todos los templates que usan los controladores para desplegar información. Los templates o vistas se organizan en carpetas, donde cada carpeta almacena todos los que están asociados a un controlador.

#### Carpeta public

Acá van lo que son hojas de estilo css, archivos javascript o imágenes.

### Mapa de navegación

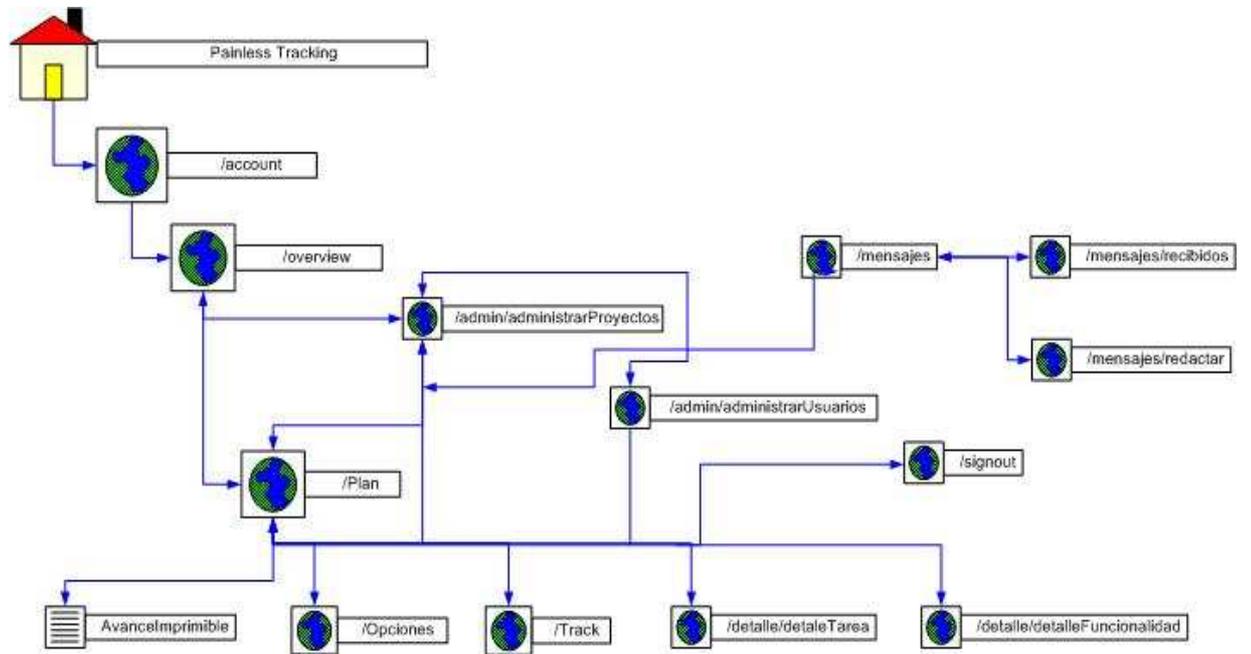


Ilustración 6-20 Mapa de navegación

Login



**Ilustración 6-21** Página de Login

Menús

Existen dos menús, uno que va en todas las páginas:



**Ilustración 6-22** Menú superior

Y otro que es visible cuando ya se escogió un proyecto para trabajar.



**Ilustración 6-23** Menú de segundo nivel

## Proyectos

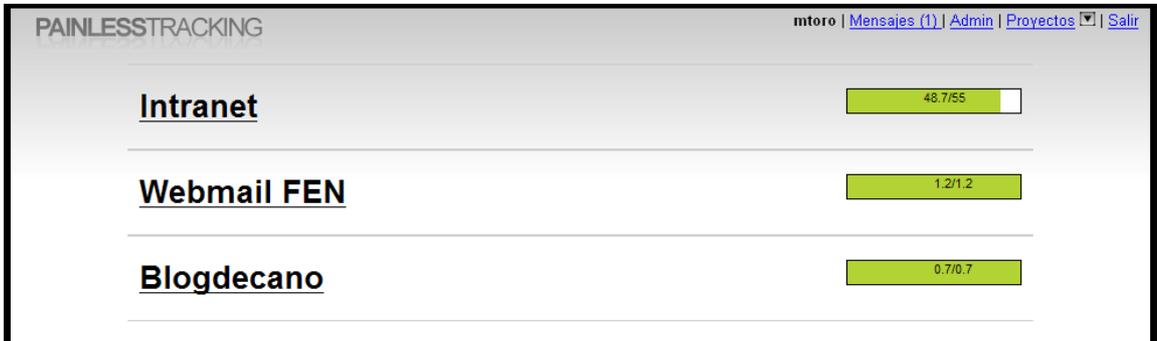


Ilustración 6-24 Vista general de proyectos a los cuales pertenece el usuario

Esta es la primera página que se muestra una vez que un usuario se ha autenticado. Acá se muestran todos los proyectos en los que pertenece y un gráfico de avance de estos. Si es que el usuario tiene un solo proyecto, se redirige automáticamente a la página del Plan de este.



Ilustración 6-25 Vista que muestra como cambiar de proyecto rápidamente

Un usuario puede cambiar de un proyecto a otro apretando en el icono al lado de "Proyecto" en el menú superior, y elegir el nombre de otro proyecto. Se hizo así porque cambiar de proyecto requería ir siempre a la página de Proyectos lo cual era bastante tedioso.

Plan

**PAINLESSTRACKING** mtoro | Mensajes (1) | Admin | Proyectos | Salir

Plan Track Opciones Proyecto: Intranet

[Obtener avance imprimible](#)

**Iteración 1** ➔ **Iteración pasada**  
 22-10-2007 -> 31-10-2007  
 Deadline: 31-10-2007 | Horas restantes: 0.0/0.0 [Agregar funcionalidad](#) | [Eliminar Timebox](#)

---

**Iteración 2** ➔ **Funcionalidad comprimida (completa)**  
 31-10-2007 -> 14-11-2007

**Tutoriales** ➔ **Tarea comenzada**  
 3/3 [Hacer favorita](#) | [Agregar Tarea](#) | [Comentarios \(0\)](#) | [Ver Track](#) | [Eliminar funcionalidad](#)

**Modulo Login** ➔

<b>Interfaz base</b> 4.5/5.5	<b>Diseñar flujo del proceso</b> 3.0/4.0	<b>Implementar flujo</b> 4.0/4.0	<b>Estudiar vulnerabilidad</b> 0/2.0	<b>Interfaz: Adaptación de CSS exter...</b> 0/1.0
---------------------------------	---	-------------------------------------	---	--

**Tarea terminada** **Tarea no comenzada**

[Hacer favorita](#) | [Agregar Tarea](#) | [Comentarios \(0\)](#) | [Ver Track](#) | [Eliminar funcionalidad](#)

---

**modulo de notificaciones** ➔ **mover tarea desde aquí** **mover funcionalidad desde aquí**

<b>Mandar un mail al crear petición</b> 1.0/1.0	<b>mandar un mail al crear wifi</b> 0/0.3
--	--

**Fin de la iteración (fondo del mar)**

[Hacer favorita](#) | [Agregar Tarea](#) | [Comentarios \(0\)](#) | [Ver Track](#) | [Eliminar funcionalidad](#)

---

Deadline: 14-11-2007 | Horas restantes: 6.3/13.71 [Agregar funcionalidad](#) | [Eliminar Timebox](#)

---

**Iteración 3** ➔  
 15-11-2007 -> 30-11-2007  
 Deadline: 30-11-2007 | Horas restantes: 0/34.28 [Agregar funcionalidad](#) | [Eliminar Timebox](#)

---

**Iteración 4** ➔  
 03-12-2007 -> 09-11-2007  
 Deadline: 09-11-2007 | Horas restantes: 0/0.0 [Agregar funcionalidad](#) | [Eliminar Timebox](#)

[Agregar Timebox](#)

Ilustración 6-26 Vista del componente de Plan

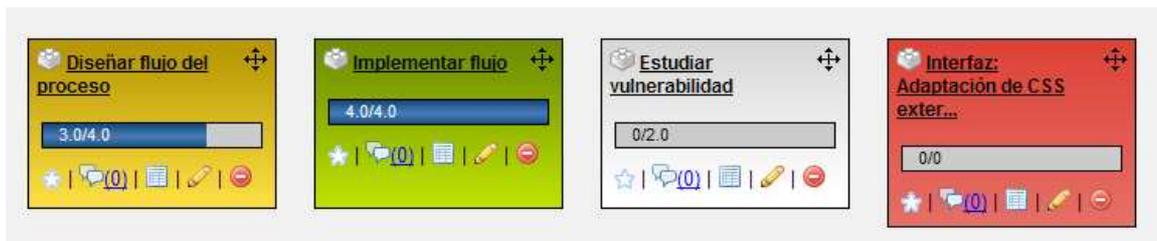
El elegir un proyecto se muestra el plan. El Plan está compuesto de timeboxes o iteraciones los cuales están compuestos de funcionalidades que a su vez se componen de tareas. Las iteraciones se ordenan por fecha, y una iteración pasada se contrae para dar más visibilidad a la iteración actual, y una iteración futura se muestra con un fondo azul para representar que la iteración actual es la “punta del iceberg” y las que vienen “el mar”.

Los timeboxes son representados por una caja (caja de tiempo), las funcionalidades por piezas de Lego armadas y las tareas por una pieza de Lego individual. Es decir, piezas de lego conforman estructuras que van dentro de una caja.

-  : Timebox
-  : Funcionalidad
-  : Tarea

Se escogió representar las tareas como cajas porque de esa manera saturan menos la vista y aprovechan mucho mejor el espacio. Esta técnica es muy usada ahora por muchos sitios, sobre todo los de noticias.

Las cajas tienen tres colores, que representan el estado de cada una:



**Ilustración 6-27** Los cuatro estados de las tareas amarillo (comenzada), verde (terminada), blanco (no comenzada) y rojo (sin estimación inicial).

- Blanco: La tarea aún no se comienza
- Amarillo: Se ha trabajado en la tarea pero aún no se termina
- Verde: La tarea ha sido terminada

- Rojo: No se ha ingresado la estimación inicial de la tarea. Esto debe ser corregido cuando antes, para que las métricas y reportes, den valores correctos.

Estos colores son los mismos que se usaban en la versión Excel de la planilla, por lo que se justifica el uso de estos mismos de manera de no hacer perder el modelo mental de los usuarios antiguos.

Además de los colores se quiso representar el estado de la tarea en términos cuantitativos y por ello se incorporó una barra de progreso que indica las horas trabajadas versus el total de horas necesarias para terminarla. La barra se escogió porque es mucho más fácil y rápida de leer que los números.

La mayoría de las acciones sobre un elemento, ya sea un timebox, funcionalidad o tarea se encuentra en la parte inferior o pie del elemento. Se optó porque la opción de mover el elemento quedara arriba a la derecha de modo de poder seleccionar el texto si es necesario. **Se usaron imágenes y texto donde era posible para un mapping mental más rápido.**

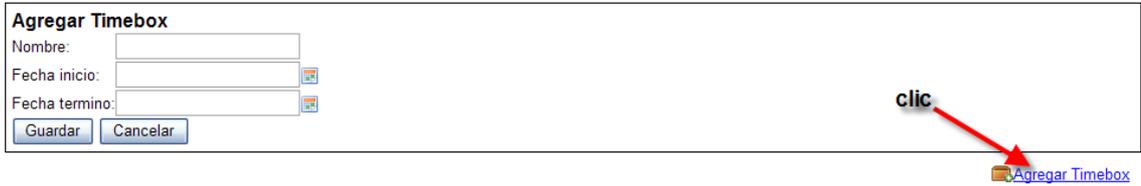
Las acciones son:

- Sobre un Timebox
  - Editar nombre, fechas de inicio y termino
  - Eliminar timebox
  - Agregar funcionalidad. Las funcionalidades siempre se van agregando al final, al igual que las tareas dentro de una funcionalidad.
- Sobre una Funcionalidad
  - Editar nombre
  - Hacer favorita. Hace o deshace que todas las tareas que componen esa funcionalidad se hagan favoritas
  - Agregar tarea. Se agrega siempre al final de la lista de tareas
  - Comentarios. Al pinchar en el nombre de la funcionalidad o en la acción Comentarios, se abre la página de detalle de ésta.

- Ver Track. Abre la página del Track mostrando solo los registros que estén asociados a tareas que compongan tal funcionalidad.
- Eliminar funcionalidad.
- Mover funcionalidad. Al hacer clic en la flecha se puede mover la funcionalidad de manera de priorizarla o cambiarla a otra iteración. Mover una funcionalidad puede hacer que varias funcionalidades se cambien automáticamente a iteraciones posteriores debido al concepto de *iceberg list*. Es decir, un timebox tiene horas limitadas y si hay funcionalidades que ocupan más de este límite, entonces se deben mover las de menos prioridad a la iteración siguiente.
- Sobre una Tarea
  - Hacer favorita. Al igual que en la funcionalidad, la estrella indica si es que la tarea es favorita o no. Basta hacer clic en ella para cambiar entre favorita o no.
  - Comentarios. Se muestra solo el icono de los comentarios.
  - Ver Track. Se muestra el icono del Track.
  - Edición rápida. Permite editar rápidamente el nombre y la estimación inicial de la tarea.
  - Eliminar tarea. Solo se muestra el icono de eliminación.
  - Mover tarea. Al hacer clic en la flecha se puede mover la tarea de manera de priorizarla o cambiarla de funcionalidad. Mover una tarea puede producir que funcionalidades cambien de timebox.

Además el Plan permite obtener una versión imprimible del estado de avance del proyecto.

### Creación de Timeboxes



**Ilustración 6-28 Vista de creación de un Timebox o Iteración**

La creación es in-place, es decir, no abre una página nueva, si no que muestra el formulario donde va a ir la iteración. Este link se encuentra al final de las iteraciones o timeboxes.

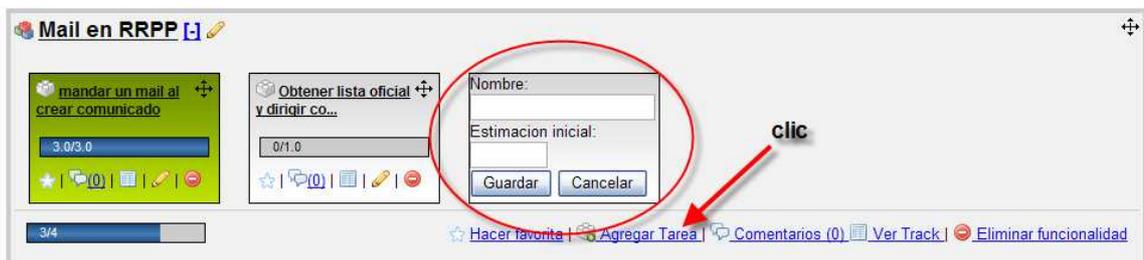
### Creación de funcionalidades



**Ilustración 6-29 Vista de creación de funcionalidades**

La creación es in-place, es decir, no abre una página nueva, si no que muestra el formulario donde va a ir la iteración. Este link se encuentra en el pie de la iteración o timebox.

### Creación de Tareas



**Ilustración 6-30 Vista de creación de Tareas**

La creación es in-place, es decir, no abre una página nueva, si no que muestra el formulario donde va a ir la iteración. Este link se encuentra al pie de la funcionalidad

### Edición rápida

La edición de cualquier elemento es in-place, es decir, el formulario se muestra en el mismo sector.

Edición de un Timebox:

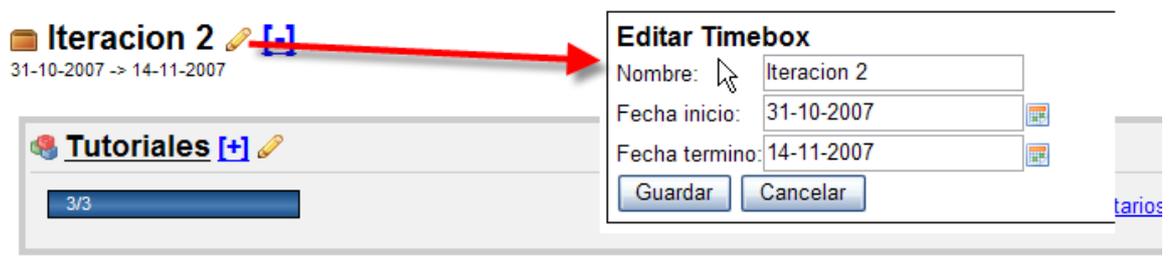


Ilustración 6-31 Vista de edición de un Timebox o Iteración

Edición de una funcionalidad:

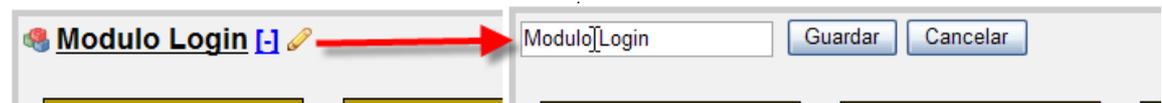


Ilustración 6-32 Vista de edición del nombre de una funcionalidad

Edición de una tarea:



Ilustración 6-33 Vista de la edición de una tarea. La tarea se convierte en un formulario.

## Track

PAINLESSTRACKING mtoro | Mensajes (1) | Admin | Proyectos | Salir

Plan **Track** Opciones Proyecto: Intranet

### Agregar Track

Tarea:

Fecha:

Ocupado: Horas:  Detalle:

Restante: Horas:  Detalle:

Quienes:  alvise  avillena  ggarcia  gnzurita  mtoro

### Track

verde indica tarea completada

Tarea	Fecha	Quien	Ocupado	Restante	Que	Faltante
<b>Modulo Login:</b> Implementar programa de aplicación que recibe datos del módulo	7-11-2007	mtoro	1.5	0.0	Ahora la intranet carga la pagina de donde se hizo login	
<b>modulo de notificaciones:</b> Mandar un mail al crear peticion	7-11-2007	mtoro	1.0	0.0	Mandar un mail al crear peticion	
<b>Mail en RRPP:</b> mandar un mail al crear comunicado	7-11-2007	mtoro	1.0	0.0	Se termino de adjuntar esto	
<b>Menu top:</b> Al apretar en Mi Cuenta sacar menus	7-11-2007	mtoro	1.0	0.0	Al apretar en MI Cuenta sacar menus	
<b>Modulo Login:</b> Diseñar flujo del proceso	6-11-2007	ggarcia, mtoro	2.0	1.0	Diseñar flujo del proceso	
<b>Modulo Login:</b> Interfaz base	6-11-2007	ggarcia	4.0	1.0	Interfaz base, hice 2 propuestas	decidir cual usar, o hacer otra
<b>Modulo Login:</b> Implementar programa de aplicación que recibe datos del módulo	5-11-2007	mtoro	4.0	1.0	Implementar programa de aplicación que recibe datos del módulo	
<b>Modulo Login:</b> Implementar flujo	5-11-2007	ggarcia	3.0	0.0	Implementar flujo	
<b>Modulo Login:</b> Implementar flujo	2-11-2007	ggarcia, mtoro	1.0	4.0	Se implementó flujo en primera instancia	Estudiar falencias y mejorar
<b>Modulo Login:</b> Diseñar flujo del proceso	2-11-2007	ggarcia, mtoro	1.0	3.0	Se diseño en primera instancia	Estudiar falencias y mejorar diseño
<b>Modulo Login:</b> Interfaz base	2-11-2007	ggarcia	0.5	4.5	Interfaz base muy básica	HAcerla de verdad
<b>MiCuenta cambiar pw v2.0:</b> Hacer funcionalidad para cambiar pwd dentro de webmail	31-10-2007	ggarcia, mtoro	3.0	0.0	Hacer funcionalidad para cambiar pwd dentro de webmail	
<b>Tutoriales:</b> Poder escribir imagenes y texto en dos columnas	31-10-2007	mtoro	3.0	0.0	Poder escribir imagenes y texto en dos columnas	
<b>Mail en RRPP:</b> mandar un mail al crear comunicado	30-10-2007	mtoro	1.0	0.5	Se averiguo como adjuntar archivo y se hicieron pruebas exitosas	Implementarlo en la funcion
<b>No Planificado</b>	29-10-2007	mtoro	0.5	0.0	Se agrego una barra para ver el strenght del pwd	

[1] 2 .. 4

### Buscar

### Tareas Favoritas

Haga clic en una tarea para autocompletar el formulario de creación de un track

### Modulo Login

- Estudiar vulnerabilidad [Comentarios \(0\)](#) [Ver Track](#)
- Diseñar flujo del proceso [Comentarios \(0\)](#) [Ver Track](#)
- Implementar flujo [Comentarios \(0\)](#) [Ver Track](#)

### Tutoriales

- Poder escribir imagenes y texto en dos columnas [Comentarios \(0\)](#) [Ver Track](#)

clic para autocompletar

Ilustración 6-34 Vista de la pantalla de Track

Según las entrevistas de usuario, la actividad más frecuente es el llenado del Track, por lo tanto se quiso tener siempre presente el formulario de inserción de registros, y

no hacer que el usuario tenga que hacer más clics para llegar a él. Se incorporó también la posibilidad de hacer Track de parte de más de un usuario a la vez, para ello basta con seleccionar los desarrolladores en la sección “Quienes”. Esto último nació de un requerimiento de un usuario que se quejaba de que tenía que llenar dos veces el Track cuando utilizaba programación de a pares XP.

#### *Los dos modelos mentales de llenado del tracking*

Las entrevistas además entregaron información acerca de como los usuarios llenan el tracking produciendo dos tipos de modelos mentales: el online y el dump mental.

El online consiste en seleccionar la tarea y luego escribir que es lo que se hizo en ella.

El dump mental consiste en escribir todo lo que se hizo en el día como varios registros y luego para cada uno de estos registros asociarlos con una tarea.

La versión en Excel permite estos dos tipos de llenados y se quiso conservarlo de las formas descritas a continuación.

#### *Llenado online*

El llenado online es el llenado más intuitivo y fácil de implementar, basta con seleccionar una tarea del cuadro de selección o de las tareas favoritas en el menú de la derecha y completar los campos restantes.

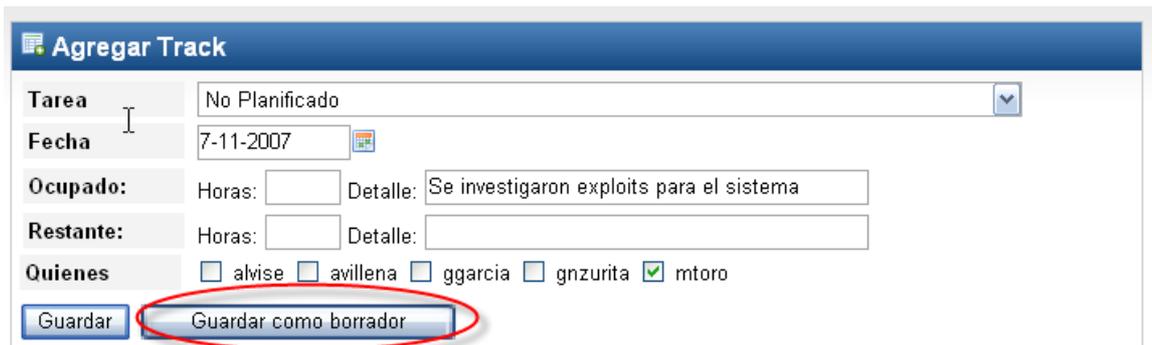
The screenshot shows a web form titled "Agregar Track". It contains the following fields and controls:

- Tarea:** A dropdown menu with the selected value "Modulo Login: Estudiar vulnerabilidad".
- Fecha:** A date input field containing "7-11-2007".
- Ocupado:** A section with "Horas: 3" and "Detalle: Se investigaron exploits para el sistema".
- Restante:** A section with "Horas: 2" and "Detalle: Falta analizar los datos con los oficiales".
- Quienes:** A section with checkboxes for "alvise", "avillena", "ggarcia", "gnzurita", and "mtoro". The "mtoro" checkbox is checked.
- Buttons:** "Guardar" (circled in red) and "Guardar como borrador".

**Ilustración 6-35 Vista de inserción de un nuevo registro en el Track**

### Llenado dump mental

El dump mental consiste en llenar los campos que uno quiera, por lo general va a ser el campo *Detalle de lo ocupado*, y luego hacer clic donde dice “Guardar como borrador”.



**Ilustración 6-36 Al hacer clic en Guardar como borrador no valida los campos**

Esto genera un nuevo registro en la tabla borrador, la cual no valida nada. Una vez hecho esto, se aprieta en “recuperar” del registro borrador, y se autocompleta nuevamente el formulario de inserción del track con los datos para ser completada con los datos que falten.

**Recuperar Track**

Tarea: No Planificado

Fecha: 7-11-2007

Ocupado: Horas:  Detalle: Se investigaron exploits para el sistema

Restante: Horas:  Detalle:

Quienes:  alvise  avillena  ggarcia  gnzurita  mtoro

### Borrador

Tarea	Fecha	Quien	Ocupado	Restante	Que	Faltante
No Planificado	7-11-2007	mtoro			Se refactorizo el código de la página de login	<a href="#">recuperar</a>
No Planificado	7-11-2007	mtoro			Se investigaron exploits para el sistema	<a href="#">recuperar</a>

**Ilustración 6-37** Al hacer clic en recuperar el formulario de inserción se reemplaza por los valores a recuperar.

Por lo tanto, el llenado podría consistir en que el usuario completa el campo *Detalle de lo ocupado*, presiona en Guardar como borrador y llena nuevamente el campo con lo siguiente que hizo en el día. Una vez hecho esto, recorre la tabla Borrador recuperando estos registros y completando los datos faltantes haciendo clic esta vez en Guardar.

#### Autocompletar de favoritas

La auto-completación de favoritas se implementó porque muchos usuarios se quejaban de que lo más tedioso de la planilla era tener que buscar cuál era la tarea en la que habían trabajado. Por lo tanto se propone este modelo de manera que los usuarios al momento de elegir en cual tarea trabajar, seleccionen de inmediato esas tareas como favoritas. Esto permite que al momento de hacer track, tengan en el menú contextual de la derecha la lista de esas tareas y puedan auto completar el formulario de inserción con un solo clic, además de filtrar el Track y ver el detalle de ésta.

**Agregar Track**

Tarea: Modulo Login:Diseñar flujo del proceso

Fecha: 7-11-2007

Ocupado: Horas: [ ] Detalle: Diseñar flujo del proceso

Restante: Horas: [ ] Detalle: [ ]

Quienes:  alvise  avillena  ggarcia  gnzurita  mtoro

Guardar Guardar como borrador

**Track**

Tarea	Fecha	Quien	Ocupado	Restante	Que	Faltante
Modulo Login:Diseñar flujo del proceso	7-11-2007	ggarcia, mtoro	3.0	0.0	Implementar flujo	
Modulo Login: Implementar flujo	2-11-2007	ggarcia, mtoro	1.0	4.0	Se implementó flujo en primera instancia	Estudiar falencias y mejorar

**Buscar**

Tareas Favoritas

Haga clic en una tarea para autocompletar el formulario de creación de un track

**Modulo Login**

Estudiar vulnerabilidad

Comentarios (0) Ver Track

**Diseñar flujo del proceso**

Comentarios (0) Ver Track

**Ilustración 6-38** Al hacer clic en la tarea, se autocompleta la Tarea y el detalle de lo ocupado

La auto-completación además de llenar el campo Tarea, llena el campo *Detalle de lo ocupado* con el nombre de la tarea, esto debido a que el nombre la mayoría de las tareas describe lo que se tiene que hacer por sí solo.

### Búsqueda y filtrado

Para buscar en el Track, basta con utilizar el cuadro de texto del menú contextual a la derecha. Además se puede filtrar haciendo clic en cualquier icono de Track dentro del sistema (por ejemplo Ver Track de la lista de favoritas)

Su búsqueda por la Tarea **Implementar flujo** obtuvo 2 resultados:

**Track**

Tarea	Fecha	Quien	Ocupado	Restante	Que	Faltante
Modulo Login: <b>Implementar flujo</b>	5-11-2007	ggarcia	3.0	0.0	<b>Implementar flujo</b>	
Modulo Login: <b>Implementar flujo</b>	2-11-2007	ggarcia, mtoro	1.0	4.0	Se implementó flujo en primera instancia	Estudiar falencias y mejorar

**Diseñar flujo del proceso**

Comentarios (0) Ver Track

**Implementar flujo**

Comentarios (0) Ver Track

**Tutoriales**

Poder escribir imagenes

**Ilustración 6-39** La búsqueda pinta de color naranja lo que se está buscando

### Edición rápida

La edición se efectúa al momento de hacer clic en el lápiz del registro a modificar. El formulario de ingreso de Track es reemplazado por uno de edición, de manera de no perder visibilidad con ventanas emergentes o cambiando la página, y además manteniendo el modelo mental del usuario para el ingreso de registros en el Track. El

cambio en el formulario se ve acompañado de efectos visuales que llaman la atención del usuario.

### Detalle funcionalidades

The screenshot displays the 'PAINLESSTRACKING' interface. At the top, there's a navigation bar with 'Plan', 'Track', and 'Opciones' tabs, and a user profile 'mtoro' with links for 'Mensajes (1)', 'Admin', 'Proyectos', and 'Salir'. The main content area is titled 'Modulo Login' and contains a description, a progress bar showing 17/22, and two action links: 'Ver track' and 'Eliminar funcionalidad'. Below this is a 'Comentarios' section with a comment from 'mtoro' dated 7-11-2007 19:04:43. At the bottom, there's a text area for 'Agregar comentario' and a 'Guardar' button. On the right, a 'Lo último' sidebar lists recent updates: 'Interfaz base' (6-11-2007), 'Diseñar flujo del proceso' (6-11-2007), 'Implementar flujo' (5-11-2007), and 'Implementar programa de aplicación que recibe datos del módulo' (7-11-2007).

**Ilustración 6-40 Vista de detalle de una Funcionalidad**

- Se puede editar rápidamente el nombre de la funcionalidad como también una descripción más extensa que solo es visible desde esta página de manera de no sobrecargar de información el plan.
- Se muestra también una barra de progreso que indica con azul cuanto se ha trabajado en total en esa funcionalidad (todas las tareas que la componen) y el restante estimado con gris.
- En el menú contextual de la derecha se muestran los últimos registros hechos en el Track de las tareas que la componen. Esto permite tener mejor visibilidad

de lo que se ha hecho a nivel más macro. Al pinchar en cualquier tarea, muestra el detalle de esta.

- Las acciones que permite son:
  - Ver Track. Redirige a la página de Track filtrando todos los registros que correspondan a tareas que componen la funcionalidad.
  - Eliminar funcionalidad. Esta eliminación es débil, es decir nunca se elimina completamente de la base de datos, de manera de no perder datos en el track.
- Se pueden agregar comentarios al final de la página. Esto permite una mayor discusión entre los integrantes acerca de la funcionalidad actual, que falta o anotaciones útiles.

## Detalle tareas

The screenshot displays the 'PAINLESSTRACKING' web interface. At the top, there is a navigation bar with 'Plan', 'Track', and 'Opciones' tabs, and a user profile 'mtoro' with links for 'Mensajes (1)', 'Admin', 'Proyectos', and 'Salir'. The project name 'Intranet' is shown in the top right. The main content area is titled 'Diseñar flujo del proceso' and includes a description, a progress bar (3.0/4.0), and a list of actions: 'Marcar como favorita', 'Ver track', and 'Eliminar tarea'. Below this is a 'Comentarios' section with two entries from 'mtoro' dated 7-11-2007. At the bottom, there is a text input field for 'Agregar comentario' and a 'Guardar' button.

Ilustración 6-41 Vista de detalle de una tarea

- Se puede editar rápidamente el nombre de la tarea como también una descripción más extensa que solo es visible desde esta página de manera de no sobrecargar de información el plan.
- Se muestra también una barra de progreso que indica con azul, cuanto se ha trabajado y en gris cuanto es el estimado restante.
- En el menú contextual de la derecha se muestran los últimos 5 registros hechos en el Track con respecto a la tarea
- Las acciones que permite son:

- Marcar como favorita. Si la estrella esta amarilla entonces esta favorita. Una vez que la tarea este favorita entonces se ve en el menú contextual de la derecha de la página de Track para un llenado más rápido.
- Ver Track. Redirige a la página de Track filtrando los registros que corresponden a esta tarea.
- Eliminar tarea. Esta eliminación es débil, es decir nunca se elimina completamente de la base de datos, de manera de no perder datos en el track.
- Se pueden agregar comentarios al final de la página. Esto permite una mayor discusión entre los integrantes acerca de la tarea actual, que falta o anotaciones útiles.

## Opciones

Indique las horas a la semana que trabaja cada desarrollador

Desarrollador	Horas
alvise	0
avillena	0
ggarcia	16
gnzurita	0
mtoro	20

Guardar

**Ilustración 6-42 Vista de las opciones del proyecto actual**

Esta página simplemente indica cuantas horas a la semana trabajan los desarrolladores de cada proyecto. Esto es necesario para calcular cuantas horas de trabajo disponible se tienen en un timebox y así corregir el *iceberg list*.

## Administrar proyectos

PAINLESSTRACKING mtoro | **Admin** | [Proyectos](#) | [Salir](#)

Administración de proyectos

### Administración de Proyectos

[Agregar Proyecto](#)

Nombre	Descripción	Integrantes	Opciones
Blogdecano		- alvise - avillena - ggarcia - gnzurita - mtoro	<a href="#">Editar</a>   <a href="#">Eliminar</a>
borrame			<a href="#">Editar</a>   <a href="#">Eliminar</a>
Intranet		- alvise - avillena - ggarcia - gnzurita - mtoro	<a href="#">Editar</a>   <a href="#">Eliminar</a>
Proyecto de prueba	aca	- avillena - ggarcia	<a href="#">Editar</a>   <a href="#">Eliminar</a>
proyecto nuevo	asdasd	- borrar	<a href="#">Editar</a>   <a href="#">Eliminar</a>
Webmail FEN		- alvise - ggarcia - gnzurita - mtoro	<a href="#">Editar</a>   <a href="#">Eliminar</a>
yea	123	- avillena - borrar	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Ilustración 6-43 Vista de administración de proyectos

Página para crear, editar y eliminar proyectos, además se indica quienes son los usuarios de ellos.

## Administrar Usuarios

**PAINLESSTRACKING** mtoro | [Admin](#) | [Proyectos](#) | [Salir](#)

Administrar usuarios | Administrar proyectos

### Administración de Usuarios

[Agregar Usuario](#)

Login	Opciones
alvise	<a href="#">Editar</a>   <a href="#">Eliminar</a>
avillena	<a href="#">Editar</a>   <a href="#">Eliminar</a>
borrar	<a href="#">Editar</a>   <a href="#">Eliminar</a>
ggarcia	<a href="#">Editar</a>   <a href="#">Eliminar</a>
gnzurita	<a href="#">Editar</a>   <a href="#">Eliminar</a>
mtoro	<a href="#">Editar</a>   <a href="#">Eliminar</a>

Ilustración 6-44 Vista de administración de usuarios

Página para crear, editar y eliminar usuarios

## Mensajes

**PAINLESSTRACKING** mtoro | [Mensajes \(1\)](#) | [Admin](#) | [Proyectos](#) | [Salir](#)

### Mensajes recibidos

[Redactar](#)  
Recibidos  
[Enviados](#)

mtoro	test 123	15-11-2007 11:41:46
ggarcia (3)	fuiste a comprar? fuiste a compraR?	14-11-2007 15:43:53
ggarcia (3)	hola 123	14-11-2007 15:27:59
ggarcia	Hola caaa asada s dfdsafasasdf	14-11-2007 15:10:44
mtoro	hola hola	14-11-2007 10:06:21
iortega	[Sin asunto] que tal	14-11-2007 09:15:11
ggarcia (5)	[Sin asunto] hola	14-11-2007 09:14:56

Ilustración 6-45 Mensajes recibidos o Inbox

Se muestra con un número entre paréntesis la cantidad de hilos, threads o respuestas que tiene cada mensaje.

## Redactar mensaje



**Redactar mensaje**

Enviar Cancelar

Para: \_\_\_\_\_

Asunto: \_\_\_\_\_

\_\_\_\_\_

Enviar Cancelar

Ilustración 6-46 Vista de redacción de un mensaje

## Ver mensaje

### fui<sup>ste</sup> a comprar?



De: **ggarcia**  
Fecha: 14-11-2007 15:43:53  
Para: **mtoero**  
fui<sup>ste</sup> a compraR?  
[Responder](#)

De: **mtoero**  
Fecha: 14-11-2007 15:44:06  
Para: **ggarcia**  
q cosa  
?

14-11-2007 15:43:53 ggarcia escribió:  
fui<sup>ste</sup> a compraR?  
[Responder](#)

Ilustración 6-47 Notar que los mensajes que tienen relación se muestran en la misma página.

Los mensajes se muestran agrupados según hilo de conversación, tal como lo hace gmail.

## Tabla Casos de Tareas versus Páginas

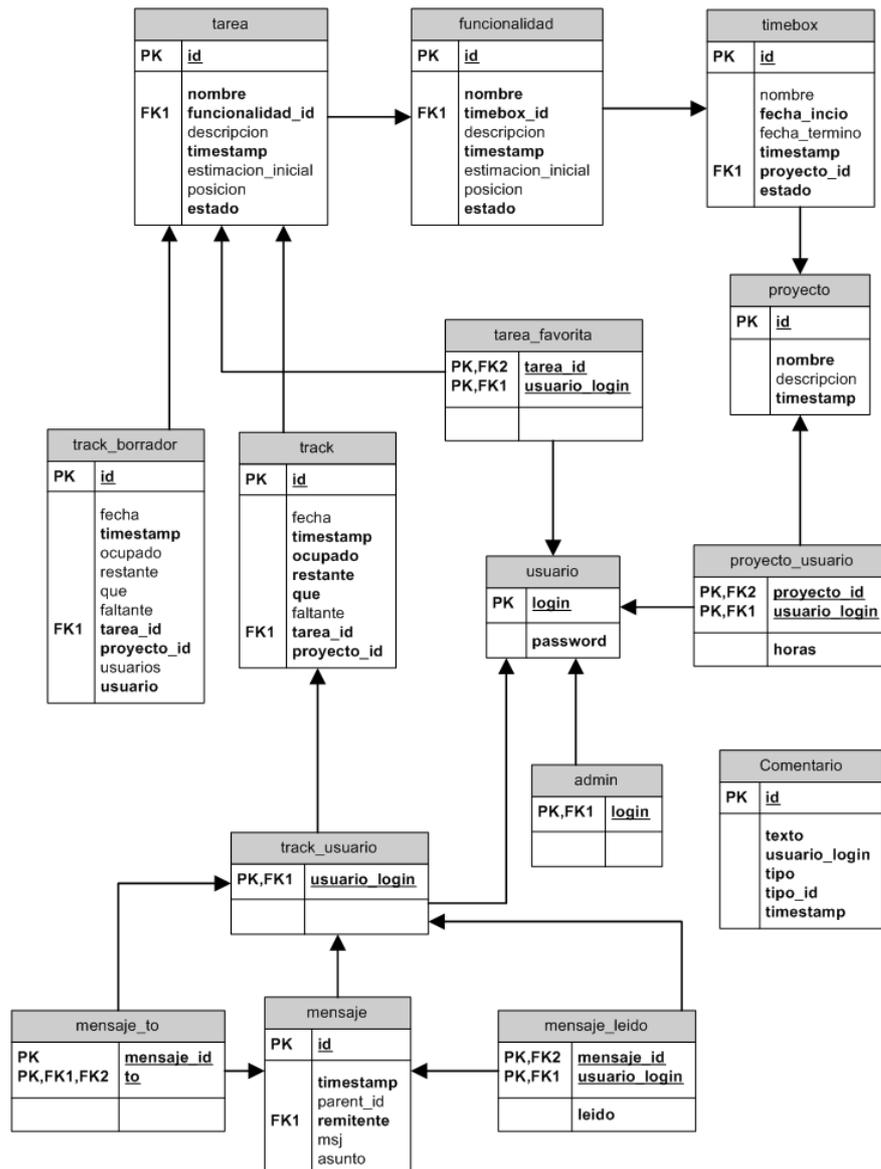
Con respecto a los Casos de Tarea que no tienen página asociada, fue porque tenían poca prioridad y no se pudieron completar por tiempo (al estilo *Iceberg list*).

Caso de tarea / Página	Login	Proyectos	Plan	Track	Detalle funcionalidades	Detalle tareas	Opciones	Administrar proyectos	Administrar Usuarios	Mensajes	Redactar mensajería	Ver mensaje
CT4.1 Ver vista general del proyecto		X										
CT4.2 Cambiar de proyecto		X	X	X	X	X	X	X	X	X	X	X
CT5.1 Consultar estado de avance		X	X									
CT1.1 CRUD funcionalidad			X		X							
CT1.2 CRUD tareas			X			X						
CT1.3 Priorizar funcionalidad			X									
CT1.4 Priorizar tarea			X									
CT1.5 Inscribirse en una tarea			X			X						
CT2.1 Ver historial tareas				X								
CT2.2 Buscar una tarea.				X								

CT2.3 CRUD Tracking				X								
CT6.1 Ver comentarios					X	X						
CT6.2 Realizar comentario					X	X						
CT5.2 Obtener informe imprimible de avance			X									
CT5.3 Revisar las métricas												
CT5.3 Exportar datos a csv												
CT7.1 Mandar mensaje privado									X	X	X	
CT7.2 Ver mensaje privados									X	X	X	
CT8.1 CRUD conocimiento												
CT8.2 Enlazar tarea a conocimiento												

Tabla 4 Casos de tareas versus páginas.

**Modelo de la base de datos**



**Ilustración 6-48 Base de datos final**

### 6.3.10. Validación con usuarios finales

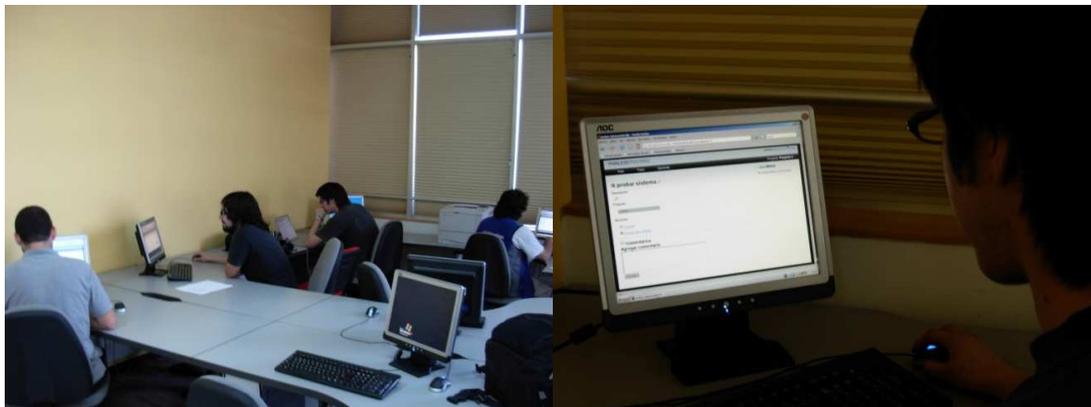


Ilustración 6-49 Pruebas de usuario

Se realizó la prueba final con seis usuarios en dos sesiones, una con cuatro integrantes y otra con dos. A cada usuario se le creó un usuario y proyecto nuevo para que pudieran trabajar sin interferencias.

El objetivo principal era que probaran el Plan y el Track, herramientas que más se usaban en la versión web, en especial el Track.

Para ello se le facilitó a cada usuario una Guía de Trabajo y una Pauta de Evaluación. La guía de trabajo se construyó de manera de cumplir con todos los Casos de Tareas o Tareas de usuarios. Dentro de la guía de trabajo debían seguir los siguientes pasos:

#### Plan

1. Crear Iteración
2. Crear dos funcionalidades
3. Editar el nombre de una de ellas
4. Dentro de la primera crear cuatro tareas
5. Editar una de ellas
6. Mover una tarea de la primera funcionalidad a la segunda
7. Priorizar la segunda funcionalidad como más importante que la primera (mover al principio de la lista)
8. Eliminar una tarea

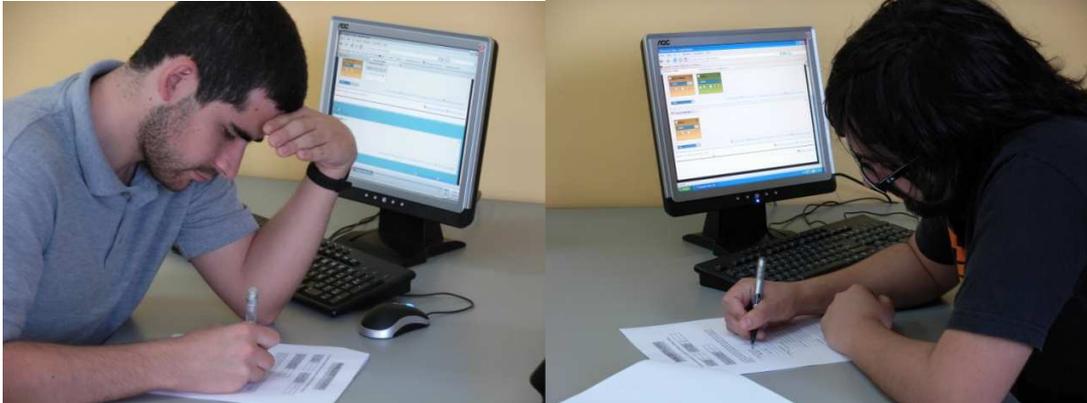
9. Hacer favoritas dos tareas
10. Ver el detalle/comentarios de una tarea.
11. Realizar un comentario

### **Track**

1. Ingresar tres registros en el Track, completar a lo más una tarea
2. Utilizar al menos una vez la opción "Guardar como borrador". Recuperar y grabar.
3. Editar un registro del Track

Las pruebas de usuarios ocurrieron de la siguiente manera:

1. Se entregó Pauta de trabajo y de evaluación, a los usuarios.
2. Se les creó un proyecto a cada usuario para que pudieran trabajar en él. Además de ese proyecto se les asignó uno que tenían todos los usuarios de prueba en común.
3. Se les entregó login y password para que pudieran ingresar el sistema.
4. Una vez comenzada la sesión solamente se observó, evitando responder preguntas hasta el final de ella. Para ello se aconsejó escribir dudas o comentarios en la pauta de evaluación para luego ser analizados en profundidad.
5. Una vez terminada la guía, los usuarios llenaron la pauta de evaluación.
6. Una vez que todos hubieran terminado, se procedió a tener una discusión grupal acerca de la usabilidad del sistema tomando nota de las ideas.



**Ilustración 6-50 Evaluación de la aplicación.**

Según lo que se observó, la creación de elementos fue muy rápida y no tuvieron mayor problema.

Se observó además una buena aceptación del sistema y el proceso fue rápido y de gran éxito. Hubo usuarios que preguntaron si es que el sistema iba a estar disponible para bajarlo desde Internet.

### ***Difficultades***

El taller de evaluación con usuarios finales, fue exitoso, se tuvo mucha cooperación de parte de ellos y el trabajo fue rápido, esto quizás debido a que eran usuarios con conocimientos previos acerca de pruebas de usuario.

Uno de los problemas fue la coordinación con los usuarios, ya que fue difícil juntarlos a todos a la misma hora para hacer las pruebas. Otro problema fue que no manejaban el concepto de Timebox (sólo conocían las Iteraciones). Con respecto a la pauta de evaluación algunos se quejaron de que era muy larga. Con respecto a la guía de trabajo, ésta no fue lo suficientemente amplia para probar todas las funcionalidades del sistema pensando que no iba a alcanzar el tiempo, pero sobró tiempo y se mostró al final de la evaluación el resto de las funcionalidades, como por ejemplo el “iceberg list”.

### 6.3.11. Evaluación de aplicación

A continuación se muestra un pequeño extracto de las pautas de evaluación:

#### Plan

- **Comentarios positivos**
  - Es fácil de usar, rápido acceso a comentarios, tareas, etc.
  - Tiene una interfaz intuitiva, bonita y agradable de usar
  - Ofrece una perspectiva global del proyecto
  - Se ve de manera gráfica un proyecto, similar a los post-it
  - Permite una fácil manipulación de los elementos con drag-and-drop
  - Muy rápida la creación de Timeboxes, funcionalidades y tareas
- **Cambios propuestos**
  - Al agregar una tarea la posición del cursor debería colocarse en el nombre del nuevo input formado
  - No se sabe cuál es la escala de medición para el tiempo estimado
  - Falta un botón de volver en el detalle
  - Las funcionalidades son muy grandes recién creadas
  - Agregar con más claridad el deadline y el tiempo restante de un Timebox
  - Al hacer clic en "Agregar favorita" cambiar link a "Quitar favorita"
  - Falta una simbología de los colores en el plan

#### Track

- **Comentarios positivos**
  - Track cómodo y rápido para agregar y editar datos
  - Se hace entretenido llenarlo
  - No es necesario volver al plan para ver a que tarea corresponde un registro
  - Es fácil agregar a más personas en un registro
  - Aparece información ya lista en el formulario de ingreso
  - Muy útil la opción "Guardar como borrador"

- **Cambios propuestos**

- El input para los detalles es muy chico
- La funcionalidad de autocompletar con las tareas favoritas no es fácil de darse cuenta que existe
- No saber que es Timebox
- El input para buscar no es visible, lo agregaría al comienzo del Track
- Debería existir la posibilidad de ordenar las columnas del Track
- Agregar una opción para ver la historia de una tarea (cambios de nombre, comentarios, track, etc)

## **Globales**

- **Comentarios positivos**

- La versión web da gusto usarla, evita las complicaciones del uso del Excel
- La nueva versión es más amigable y colaborativa, que es lo fundamental
- Fácil de uso, mejor acceso a los datos.

- **Cambios propuestos**

- Faltan métricas generales, un Help o FAQ.

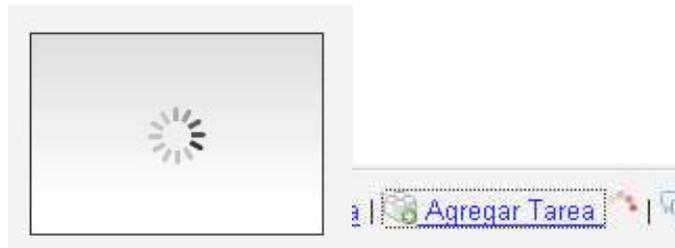
### **6.3.12. Últimos cambios**

Qué cosas se mejoraron después de las pruebas de usuario:

- **Plan**

- Al agregar una tarea colocar la posición del cursor en el nombre del nuevo input formado
- Explicar la escala de medición de los tiempos estimados
- Se mejoró la visibilidad del deadline y horas restantes/ocupadas dentro de un Timebox
- Al hacer clic en "Agregar favorita" se cambia el texto a "Quitar favorita" y viceversa.

- Se agregaron animaciones que representan que el sistema está cargando, cada vez que se activa una acción asincrónica. Esto de manera de entregarle mayor feedback al usuario cuando realice este tipo de acciones.



**6-51 Ejemplos de feedback que se muestra al usuario al realizar acciones. A la izquierda se muestra una tarea en proceso de edición. A la derecha se muestra una animación mientras el formulario todavía no llega del servidor**

○

- Track

- El texto explicativo de la funcionalidad Autocompletar se movió al final de la sección, porque hacía parecer una división de funcionalidades
- Para la funcionalidad de Autocompletar se disminuyó la separación entre tarea y funcionalidad ya que no se demostraba relación entre estas dos.
- Se alargó el campo Detalle de Restante y Ocupado en el formulario para Agregar Track.
- El agregado de registros al Track es mediante asincrónico usando AJAX. Esto permite que este proceso sea más rápido.

**Agregar Track**

Tarea: No Planificado

Fecha: 29-11-2007

Ocupado: Horas: 2 Detall

Restante: Horas: 0 Detall

Quienes:  ggarcia  mto

Guardar Guardar como borrador

**Track**

Tarea	Fecha	Qui
No Planificado	29-11-2007	mto

**6-52 Al agregar un registro al Track, se muestra feedback del tiempo en el que tarda**

- o Las columnas son ordenables ahora

Tarea	Fecha	Quien	Ocupado	Restante	Que	Faltante
No Planificado	29-11-2007	mto	2	0.0	asdasd	

**6-53 Al hacer clic sobre el nombre de una columna, la tabla se ordena por ese campo. Si se hace clic nuevamente, entonces se ordena en el orden inverso**

- Detalles
  - o Se agregó un botón "Volver" en las páginas de detalle tanto de Tareas como de Funcionalidades
  - o Al agregar un comentario se cambió el texto "Guardar" por "Insertar comentario" porque incitaba a pensar que ese botón guardaba toda la información de la página

## **7. Conclusiones**

Uno de los objetivos principal de esta memoria fue que los usuarios lograran obtener una mejor experiencia de la Painless Tracking y aprovechar así, todos los beneficios que ésta entrega, focalizando el trabajo en la usabilidad. Para ello se usó una metodología de trabajo ágil orientada a la usabilidad que fue adaptada de manera de lograr mejores resultados.

### **7.1. Sobre la metodología**

Esta metodología arrojó muy buenos resultados, ya que permitió una buena estructuración de los elementos, centrándose en el usuario final, para evitar y facilitar la re programación al final del desarrollo. Está propuesto usar esta metodología para proyectos más complejos, pero se estima, según la experiencia obtenida, que con el suficiente tiempo, orden y modularidad, esta metodología también podría ser aplicada.

La experiencia obtenida fue muy positiva, entregando un producto muy rápido para trabajar, muy dinámico, incluyendo en la etapa de diseño a usuarios finales y expertos, logrando así, la mayor cantidad posible de información fundamental para el proyecto, que de haber sido individual el trabajo no se hubiera podido percibir. Además esta metodología, permite que la modificación de elementos e ideas sea rápida y de gran utilidad ya que se tuvo que poner en el lugar de los usuarios finales muchas veces. Se priorizaron los elementos más importantes obteniendo así un producto mucho más útil para el usuario. Por otro lado, el trabajo por talleres - donde todos participan - es muy agotador, debido a que todo el grupo trata de manifestar sus ideas, tratando de dar el 100% de cada uno. Esto se puede mejorar disminuyendo la duración y aumentando la frecuencia de los talleres.

También hubo errores conceptuales al momento de definir roles de usuarios, ya que no estaba claro, por ejemplo, que planificador y tracker podían ser la misma persona que tomaba distintos roles en distintas ocasiones.

### **7.2. Sobre el valor agregado**

Para el producto final se quiso potenciar la rapidez de uso de la planilla para que esta fuera realmente "indolora" y que no quite tanto tiempo, como lo hacía la versión Excel.

Se generó valor agregado al cambiar de plataforma Excel a Web, ya que por un lado es colaborativa, es decir, el track ya no es necesario que se haga de manera “serial”.

Se incorporó el concepto de “proyectos” en el plan, lo que permite que este software sirva para equipos de trabajos que trabajan en varios proyectos a la vez, brindando una visión global de estos, al ver gráficamente el avance en la página inicial de la Painless Tracking.

#### **7.2.1. Sobre el Plan**

Con respecto a la administración de Plan, ya sea tareas o funcionalidades es mucho más rápida y organizada. Las tareas que pertenecen a una misma funcionalidad se ven como un grupo y no quedan en cualquier lugar. La priorización se refleja ahora, no como un campo dentro de un ítem, sino como la posición relativa entre estos, todo esto generado por el concepto de “*iceberg list*”, además priorizar es mucho más intuitivo y fácil de lograr debido al *drag-and-drop* entre los elementos. Por otro lado, los colores y las barras de avances ayudan a ver el estado de una funcionalidad con solo una mirada general.

Otro tema que genera valor es el concepto de *Timebox*: ahora no es necesario calcular manualmente cuantas horas de trabajo quedan para finalizar una iteración. Gráficamente se muestra cuando el *timebox* se copa de funcionalidades, porque el sistema empieza a mover automáticamente las funcionalidades que no caben hacia *timeboxes* o iteraciones futuras. Esto permite mayor rapidez en lo que es la negociación y re negociación de los elementos a entregar al finalizar el tiempo, esto muy bien evaluado por los usuarios.

En general, se logró potenciar la rapidez para crear y modificar los elementos dentro del plan, logrando además que visualmente fuera atrayente para usuarios viejos y nuevos.

#### **7.2.2. Sobre el Track**

Con respecto al Track, lo que más se trató de lograr, fue facilitar el ingreso de registros a éste por parte de los usuarios, y para ello se agregó el concepto de “Tarea favorita”, esto debido a que la acción más tediosa era elegir la tarea correspondiente al registro.

Ahora la mitad de los datos a ingresar en un registro se llena haciendo un solo clic. Con la edición ocurre lo mismo, ésta no abre una página nueva o tapa información con una ventana *popup* si no que se aprovecha el mismo formulario de ingreso para la edición. En general a los usuarios les gustó mucho la rapidez con la que se llenaba el *Track*.

Otro valor que se agregó fue el hecho de que un registro del *track* lo pueden componer varios desarrolladores, esto permite por ejemplo, que equipos que trabajan con metodología de programación de a pares solo tengan que ingresar un solo registro, y no dos por cada desarrollador como ocurría en la versión original.

Además se incorporó el concepto de “registros borrador”, el cual brinda flexibilidad al momento de llenar el Tracking, debido a que permite por ejemplo, escribir y guardar rápidamente lo que se hizo para que posteriormente se complete con la información restante (que es más lenta de llenar), generando valor a la rapidez de su uso.

También se agregó el módulo de mensajería, que permite a los usuarios tener una mayor comunicación entre ellos dentro de la misma aplicación.

### **7.3. Experiencia personal**

Este software se usó desde el momento en el que fue creado hasta el día de hoy, para manejar un proyecto real, obteniendo así, una experiencia personal verdadera. Esta experiencia entregó mucha información que por sí sola no era fácil de visualizar. Por ejemplo, el *Track* a pesar de ser rápido, producía que al insertar un registro, la página se recargaba completamente. Esto restaba segundos a la continuidad de este proceso, y con el tiempo se convirtió en un problema a considerar. Es por esto que se cambió el sistema de llenado de *Track* a uno asíncrono, aprovechando las características de AJAX, haciendo de este proceso una experiencia mucho más agradable. A raíz de esto, también se pudo mejorar el feedback que obtiene el usuario al realizar acciones, mejorando claramente la usabilidad del sitio.

En resumen se tuvo una muy buena aceptación de la planilla, con pocas críticas, obteniendo con éxito una herramienta que motiva su uso por parte de los usuarios finales.

## **8. Trabajo futuro**

Dentro de las tareas propuestas que faltan por hacer en la versión Web, están:

- Agregar un Help o FAQ para el sitio
- Implementar el módulo de Conocimiento o Wiki
- Cohesionar memoria con memoria paralela que ve el tema de las métricas
- En el Track:
  - Agregar una opción para ver la historia de una tarea (cambios de nombre, comentarios, track, etc)
- Poder exportar los datos a CSV

## 9. Bibliografía

- [1] *agile-usability*. (s.f.). Obtenido de agile-usability: Agile Usability: <http://tech.groups.yahoo.com/group/agile-usability/>
- [2] Alistair A.R. Cockburn (s.f.). *Earned-value and burn charts*. Obtenido de [http://alistair.cockburn.us/index.php/Earned-value\\_and\\_burn\\_charts](http://alistair.cockburn.us/index.php/Earned-value_and_burn_charts)
- [3] Ambler, S. W. (s.f.). *User Experience Activities on Agile Development Projects*. Obtenido de Introduction to Agile Usability: <http://www.agilemodeling.com/essays/agileUsability.htm>
- [4] Bardram, J. E., Bunde-Pedersen, J., & Soegaard, M. (2006). Support for Activity-Based Computing in a Personal Computing Operating System.
- [5] Beck, Kent, y Martin Fowler. *Planning Extreme Programming*.
- [6] Beck, K., & Cleal, D. (s.f.). *Optional Scope Contracts*. Obtenido de <http://www.plonesolutions.com/about/OptionalScopeContracts.pdf>
- [7] Berteig, M. (17 de 2 de 2006). *Timeboxing: A Critical Agile Work Practice*. Obtenido de Agile Advice: [http://www.agileadvice.com/archives/2006/02/timeboxing\\_a\\_cr.html](http://www.agileadvice.com/archives/2006/02/timeboxing_a_cr.html)
- [8] Cutrell, E., Robbins, D. C., Dumais, S. T., & Sarin, R. (2006). Fast Flexible Filtering with Phlat – Personal Search and Organization Made Easy.
- [9] Neustaedter, C., & Bernheim Brush, A. (2006). LINC-ing the Family: The participatory Design of an Inkable Family Calendar.
- [10] Nielsen, J. (1999). *Designing Web Usability : The Practice of Simplicity (Paperback)* . Peachpit Press.
- [11] Pasquero, J., Luk, J., Little, S., MacLean, K., Lévesque, V., & Hayward, V. (s.f.). A role for Haptics in Mobile Interaction: Initial Design Using a Handheld Tactile Display Prototype.
- [12] Patton, J. (s.f.). Hitting the Target: Adding Interaction Design to Agile Software Development. Obtenido de <http://oopsla.acm.org/extra/pracreports/HittingTheTargeReport.pdf>
- [13] Patton, J. (s.f.). *User Centered Agile Software Development*. Obtenido de AgileProductDesign.com: [http://www.agileproductdesign.com/presentations/user\\_centered\\_agile\\_development/index.html](http://www.agileproductdesign.com/presentations/user_centered_agile_development/index.html)
- [14] *Rememberthemilk*. (s.f.). Obtenido de <http://rememberthemilk.com/>
- [15] Spolsky, J. (s.f.). *Painless Software Schedules*. Obtenido de <http://www.joelonsoftware.com/articles/fog0000000245.html>
- [16] *TickSpot*. (s.f.). Obtenido de <http://www.tickspot.com/>

[17] *Timesheets*. (s.f.). Obtenido de <http://www.timesheetsmts.com/>

[18] *Todoist*. (s.f.). Obtenido de <http://todoist.com/>

[19] Villena, A. (s.f.). La propuesta ágil.

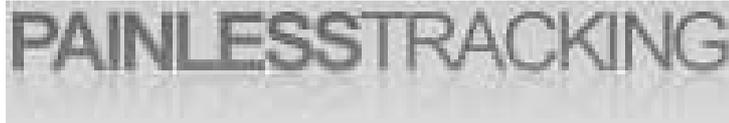
[20] *VisionOne*. <http://www.versionone.net>.

[21] *XPlanner*. <http://www.xplanner.org/>.

[22] Villena, A. (s.f.). Documentos varios

## 10. Anexos

### 10.1. Pauta de evaluación



#### Encuesta de Satisfacción de usuarios de Software Painless Tracking

<i>FECHA</i>	<i>DIA</i>	
	<i>MES</i>	

<i>ENCUESTADO</i>	<i>SEXO</i>	
	<i>EDAD</i>	

<i>EXPERIENCIA</i>	<i>INEXPERTO</i>	
	<i>AVANZADO</i>	

Le solicitamos que conteste el presente cuestionario considerando que la mayoría de las preguntas son de opinión (por lo que no existen respuestas correctas o incorrectas) y de carácter anónimo, donde sus respuestas serán consideradas sólo en forma agregada e impersonal.

1. Painless Tracking Excel
  - a. Que es lo bueno

---

---

b. Que es lo malo

---

---

2. Painless Tracking Web

a. Plan

i. Que es lo bueno

---

---

---

ii. Que es lo malo

---

---

---

iii. Que agregaría

---

---

---

iv. Que sacaría

---

---

---

v. ¿Qué tanto le costó realizar las tareas?

1. Crear Iteracion

---

---

2. Crear dos funcionalidades

---

---

3. Editar el nombre de una de ellas

---

---

4. Dentro de la primera crear cuatro tareas

---

---

5. Editar una de ellas

---

---

6. Mover una tarea de la primera funcionalidad a la segunda

---

---

7. Priorizar la segunda funcionalidad como más importante que la primera (mover al principio de la lista)

---

---

8. Eliminar una tarea

---

---

9. Hacer favoritas dos tareas

---

---

10. Ver el detalle/comentarios de una tarea.

---

---

11. Realizar un comentario

b. Track

i. Que es lo bueno

---

---

---

ii. Que es lo malo

---

---

---

iii. Que agregaría

---

---

---

iv. Que sacaría

---

---

---

v. ¿Cómo encuentra la funcionalidad de autocompletar?

1. Ingresar tres registros en el Track, completar a lo más una tarea

---

---

2. Utilizar al menos una vez la opción "Guardar como borrador". Recuperar y grabar.

---

---

3. Editar un registro del Track

---

---

3. Comparación

a.Cuál prefiere, ¿por qué?

---

---

---

Muchas gracias por su tiempo y su colaboración

