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

# DESARROLLO DE UNA INTERFAZ DE GESTIÓN PARA PAINLESS TRACKING WEB

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

**RODRIGO IGNACIO OJEDA CÁRCAMO**

**SANTIAGO DE CHILE**
**2008**

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

# DESARROLLO DE UNA INTERFAZ DE GESTIÓN PARA PAINLESS TRACKING WEB

**RODRIGO IGNACIO OJEDA CÁRCAMO**

**MIEMBROS DE LA COMISIÓN EXAMINADORA**

**SR. AGUSTÍN VILLENA, PROFESOR GUÍA**
**SRA. MARÍA CECILIA BASTARRICA**
**SR. KURT SCHWARZE**

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

**SANTIAGO DE CHILE**
**2008**

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN
POR: RODRIGO IGNACIO OJEDA CÁRCAMO
PROF. GUÍA: SR. AGUSTÍN VILLENA

# Desarrollo de una interfaz de gestión para Painless Tracking Web

El desarrollo de un área de gestión más acabada para la nueva versión de la herramienta Painless Tracking versión Excell llamada Painless Tracking Web, motiva el trabajo realizado en esta memoria.

A través de este él se hará una investigación en primera instancia sobre el proceso de gestión en proyectos que utilicen metodologías ágiles como trasfondo de desarrollo. Para ello se ahondará en temas como procesos de medición, estructura adecuada de métricas y reportes y la utilización de estos mismos.

Luego se hará una revisión sobre las distintas métricas y reportes presentes en las distintas metodologías ágiles actuales y sobre la misma herramienta Painless Tracking Excell.

Después se procederá a realizar una selección sobre las métricas y reportes investigados en el punto anterior, para luego realizar una aplicación práctica de los mismos con datos reales de un proyecto de desarrollo.

Luego se implementarán usando desarrollo guiado por tests como metodología y tecnologías basadas en Python, XML y Flash.

Finalmente se concluirá sobre los componentes desarrollados para la interfaz de gestión de Painless Tracking Web, el valor generado por esta investigación y las proyecciones futuras sobre la misma.

## Agradecimientos

Gracias de todo corazón a quienes estuvieron siempre conmigo, tanto en los momentos felices como en los más difíciles: a mi amada Elsa Eugenia, a mis padres, Alicia e Ignacio, a mi hermana y a mi sobrina, Tamara e Isidora, a mi tía y madre de mi amada, Elsa, a mi profesor y amigo Agustín y a todos mis familiares y amigos cuyas palabras fueron siempre de ánimo, y en especial, a mi querida abuela Lidia que desde el cielo nos cuida.

*“ Toca el cielo,*

*con ustedes gran bosque,*

*el verde gracias. ”*

Para ustedes, Rodrigo.

# Índice

# 1. Introducción

El objetivo primordial de la ingeniería del software es producir un sistema, aplicación o producto de alta calidad llevando a cabo el mejor desarrollo posible, dados los recursos, tiempo y desarrolladores disponibles. Esta restricción es la que conlleva a aplicar un proceso de gestión adecuado para que los recursos disponibles produzcan el mayor valor posible. Para lograr este objetivo, los ingenieros de software deben emplear métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de gestión, operación y desarrollo del software. Al mismo tiempo, un buen ingeniero de software y los buenos gestores de la ingeniería de software deben medir si la alta calidad se va a llevar a cabo.

La calidad de un sistema, aplicación o producto es tan buena como los requisitos que detallan el problema, los diseños que modelan la solución, los códigos que se transfieren a un programa ejecutable y las pruebas que ejercita el software para detectar errores. Un buen ingeniero del software emplea mediciones que evalúan la calidad del análisis y los modelos de diseño, así como el código fuente y los casos de prueba que se han establecido al aplicar la ingeniería del software. Para obtener esta evaluación de calidad, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad. Asimismo, un buen administrador de proyectos debe evaluar la calidad objetivamente y no subjetivamente. A medida que el proyecto progresa el administrador del proyecto siempre debe valorar la calidad. Aunque se pueden recopilar muchas medidas de calidad, el primer objetivo en el proyecto es medir errores y defectos. Las métricas que provienen de estas medidas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad del desempeño de los desarrolladores. Por ejemplo los errores detectados por hora de revisión y los errores detectados por hora de prueba suministran una visión profunda de la eficacia de cada una de las actividades relacionadas a la métrica. Así los datos de errores se pueden utilizar también para calcular la eficiencia de eliminación de defectos en cada una de las actividades del marco de trabajo del proceso.

Para las metodologías relacionadas con la tendencia ágil de desarrollo de software no son válidas las mismas métricas que para la tendencia tradicional, ya que éstas tienen filosofías de fondo distintas. A los modelos ágiles les interesa medir el trabajo realizado como base para estimar el trabajo futuro. Están más enfocadas en estimar lo que viene que llevar una bitácora del pasado. Un ejemplo con las líneas de código muestra muy bien esta diferencia. Para los modelos tradicionales es un punto de referencia y una medición importante para determinar el trabajo o el tamaño. Sin embargo, los modelos ágiles incentivan reducir las líneas de código. Es uno de los objetivos de la técnica ágil de refactorización. Desde el punto de vista ágil, pasar de 100.000 a 70.000 líneas de código sería un avance y desde el punto de vista tradicional, el proyecto estaría yendo hacia atrás [3].

Las áreas de medición ágiles se centran en:

- Funcionalidades desarrolladas / funcionalidades pendientes de desarrollo.
- Satisfacción del cliente
- Tiempo pendiente para cerrar el ciclo de desarrollo. [3]

Los principales reportes que manejan los modelos ágiles son los gráficos de flujo de avance (Cumulative Flow Diagrams) y los gráficos de avance de tareas (Burn-Down Charts) y de valor generado (Burn-Up Charts). [4][5][6]

Lo principal que podemos destacar sobre la medición y reportes en las metodologías ágiles es que el aspecto más importante para asegurar la calidad del software es el valor y la satisfacción entregados al cliente.

La programación extrema o eXtreme Programming (XP) [1] es un enfoque de la ingeniería de software formulado por Kent Beck [2]. Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser

capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Para ello se aplican modelos de gestión flexibles al cambio, enfocados a la generación de valor en forma temprana. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software.

Las características fundamentales del método son:

- Desarrollo iterativo e incremental que permita la generación de valor en forma temprana.
- Frecuente interacción del equipo de programación con el cliente o usuario para gestionar en forma óptima el cambio de requisitos constante.
- Entregas frecuentes por la misma razón del primer punto.
- Refactorización del código para generar un software escalable en el tiempo y flexible a la adición de nuevos requerimientos.
- Simplicidad en el código, para construir un software de calidad y libre de complicaciones que no generen valor.

Siguiendo la filosofía detrás de las metodologías ágiles, principalmente XP, el profesor Agustín Villena ha desarrollado la herramienta Painless Tracking. Ésta es una herramienta de uso intuitivo para seguir el desarrollo de un software de manera indolora. Sus principales aspectos se enfocan en:

- Permitir que los desarrolladores construyan fácilmente las listas de las características solicitadas para el software, desgranándolas en tareas ad-hoc para después seguir el progreso de cada una de ellas.
- Apoyar en el desarrollo a múltiples usuarios, proyectos y más.

- Permitir la comparación de las estimaciones originales con el avance en tiempo real.
- Permitir el filtrado de tareas según sus características (responsable, avance, etc.)
- Incorporar las notas detalladas sobre cada característica y tarea.
- Manejar una historia completa de todos los cambios de la tarea.

Las métricas que maneja actualmente Painless Tracking son las siguientes:

- Velocidad de desarrollo: horas ideales ocupadas en funcionalidades finalizadas, versus horas totales ocupadas.
- Error de estimación: para las tareas finalizadas, relación entre tiempos estimados y los realmente ocupados.
- Disponibilidad: relación entre lo ocupado en horas de tareas planificadas y el total de horas ocupadas en el trabajo.
- Tiempo estimado de finalización del software.

Todas ellas están enfocadas a medir el **valor** entregado al cliente, principalmente la velocidad de desarrollo.

# 2. Motivación

Se vio anteriormente la necesidad de medir como una forma de conocer la calidad del software que se está desarrollando. Así mismo, las metodologías ágiles también necesitan métricas y reportes que entreguen una visualización del valor y esfuerzo generado durante el desarrollo de los proyectos. Esta visualización corresponde aplicarla tanto a nivel de los equipos de desarrollo con fines organizativos del trabajo a realizar como a nivel gerencial para la gestión y planificación de los negocios en que se enfocará la empresa.

Además si se pretende aplicar metodologías ágiles en los desarrollos en una empresa, se debe contar con herramientas capaces de cubrir las necesidades informativas tanto por el lado de la empresa como el del cliente. Por ejemplo, una herramienta comúnmente utilizada para realizar esto es la Carta Gantt. Es muy útil a nivel de metodologías tradicionales pero no es usable en el ámbito ágil debido a la filosofía detrás éste. Principalmente porque la Carta Gantt nos entrega un informe difícil de modificar constantemente junto con las complicaciones de ajustar o mover adecuadamente las tareas. En el área ágil se busca una visión hacia el futuro, por lo que se necesita una herramienta que soporte ser modificada constantemente para poder llevar una mejor estimación de lo que se avecina en un futuro próximo.

Actualmente la herramienta Painless Tracking se encuentra en un proceso de cambio de plataforma, para el cual se están realizando diversos proyectos en paralelo cubriendo las siguientes líneas de trabajo:

- Presencia on-line en la Web.
- Usabilidad a nivel multiusuario.
- Metodologías para insertar su uso en una organización y medir el grado de adopción.
- Visualización del valor generado en los proyectos de software a través de métricas y reportes de acuerdo al enfoque de las metodologías ágiles.

Este último punto es el motivo de esta investigación, para lo cual se requiere diseñar y construir para la nueva versión Web de Painless Tracking (que denominaremos “Painless Tracking Web”) la capacidad de entregar información estratégica para la gestión de proyectos de software, a través de métricas y reportes ágiles, algunos de los cuáles no ha sido posible implementar en la actual versión de la Painless Tracking, basada en Excel. Esto se debe principalmente a limitaciones del modelo lógico y de dicha plataforma.

# 3. Objetivos

Actualmente la herramienta Painless Tracking provee una muy buena interfaz operativa para el control y seguimiento de los proyectos pero carece de una interfaz tanto administrativa como de gestión. El objetivo general es el desarrollo de una interfaz de gestión para la Painless Tracking Web que permita entregar la información necesaria para la toma de decisiones en los proyectos involucrados. Para ello se deben definir y desarrollar métricas y reportes para hacer de esta herramienta más completa, consistente con la metodología ágil detrás de ella y competidora con otras herramientas de gestión y control de proyectos.

Los objetivos específicos detrás de este desarrollo son:

1. Administrar y gestionar los proyectos en forma más eficiente a partir de la definición de la información necesaria para que el proceso de toma de decisiones se haga en forma temprana.
2. Definir y desarrollar métricas y reportes adecuados para la Painless Tracking Web a partir de la información anterior. Esto es, que puedan ser obtenidos a partir de la información que maneja la herramienta.
3. Generar valor con la herramienta tanto para el cliente, los gestores de los proyectos y el equipo de desarrollo en general, a través de los siguientes puntos:
    a. Proveer un control minucioso de los proyectos.
    b. Obtener una visualización del avance para todos los agentes involucrados en los proyectos.
    c. Obtener estimaciones de avance y alcance para los hitos próximos dentro de los proyectos actuales y los futuros.

# 4. Metodología

Para este desarrollo, los pasos a seguir son los siguientes:

1. Revisar trabajos publicados y memorias anteriores con temas relacionados. Se seleccionarán temas afines y se evaluará la reutilización de conceptos, diseños e implementaciones presentes en estos trabajos. [1 a 6]
2. Definir la información relevante al plano estratégico relacionado con las metodologías ágiles a partir de la revisión en el punto anterior.
3. Investigar métricas en metodologías ágiles actuales:
    a. Programación Extrema, es uno de los ejemplos más exitosos de metodología ágil. [1]
    b. Scrum. [7]
    c. Crystal. [8]
    d. Evolutionary Project Management (Evo). [9]
    e. Feature Driven Development (FDD). [10]
    f. Adaptive Software Development (ASD). [11]
    g. Lean Development (LD) y Lean Software Development (LSD). [12]
4. Estudiar el estado del arte externo e interno de la herramienta Painless Tracking. Realizar una nueva selección de reportes a partir de este estudio.
5. Crear los diseños pertinentes para su adaptación a la herramienta Painless Tracking Web realizando un trabajo en conjunto con la Interfaz Operativa a ser desarrollada en paralelo a esta memoria.
6. Implementación de los diseños anteriores sobre una plataforma base construida en conjunto con los otros participantes en las líneas de desarrollo de la versión web de la Painless Tracking
7. Medición del desempeño de las métricas y reportes desarrollados en proyectos reales.

# 5. Antecedentes

En todo trabajo de ingeniería es importante conocer cuales son los avances que existen en la solución de problemas similares a los enfrentados. En el caso de este trabajo, se presentarán lo investigado sobre herramientas de apoyo a la gestión ágil y su trasfondo teórico en el contexto de la industria del software, y el estado actual de desarrollo de la Painless Tracking en lo que se refiere al aspecto de medición de avance de proyectos de software.

## 5.1. Enfoque del proceso de medición

Un proceso de medición que acompañe al proceso de desarrollo de software y se integre con él, ayudará en la interpretación, control y mejora de cada una de las actividades que se llevan a cabo dentro del mismo. Las medidas y las métricas son herramientas muy útiles para el control y la gestión de los procesos involucrados, las cuales deben estar orientadas a obtener resultados concretos y ser comparables en puntos bien definidos, por ejemplo, aspectos comunes a ser medidos o las unidades de medida utilizadas. Además, como son utilizadas para una mejor gestión de los proyectos, las medidas y métricas deben ser diseñadas para soportar los objetivos del negocio de la organización.

Toda métrica debe tener un objetivo claro y definido para el cual ha sido creada. Con frecuencia, una métrica o un indicador por sí solo no es lo suficientemente significativo. Es la correlación de métricas y/o indicadores lo que permite extraer conclusiones determinantes. Las métricas e indicadores se obtienen a partir de medidas previamente tomadas. En este contexto, es claro que las medidas a tomar deben estar dirigidas por las metas y objetivos que se desean alcanzar. Algunas razones para medir el proceso de desarrollo de software serían [17]:

- **Caracterización**: para facilitar el entendimiento de los procesos, productos, recursos y entornos, y para establecer líneas de base que sirvan de referencia en estimaciones futuras.

- **Evaluación**: para determinar el estado actual respecto de los planes. Las medidas son los sensores que avisan cuando los proyectos y procesos se están desviando de los planes, y por tanto, permiten corregir esa desviación. También se realizan evaluaciones para estimar o determinar el nivel de cumplimiento de los objetivos, para determinar el impacto de mejoras en aspectos como la tecnología y para medir la calidad de los procesos.

- **Pronóstico temprano**: principalmente para poder planificar a tiempo. Medir para poder pronosticar genera un mejor entendimiento de las relaciones existentes entre los procesos de desarrollo y los productos resultantes de ellos. Si se construyen modelos de estas relaciones, los valores de algunos atributos pueden ser usados para predecir otros. Esto se hace porque se quieren establecer objetivos alcanzables respecto al costo, los plazos de entrega y la calidad, asignando los recursos apropiados a cada tarea. Las medidas orientadas al pronóstico se usan para la extrapolación de tendencias, es decir, estimaciones de costo, plazos y calidad que pueden ser mejoradas teniendo en cuenta las evidencias actuales. Los datos históricos pueden ser útiles para analizar las desviaciones de las estimaciones respecto de la realidad y para corregir desviaciones a fin de disminuir riegos futuros.

- **Mejoramiento**: la información cuantitativa debe ayudar a identificar las barreras e ineficiencias del proceso de desarrollo, así como las oportunidades de mejoramiento. De igual manera, las medidas actuales proporcionarán la información de referencia una vez se hayan aplicado acciones de mejora a los procesos para así poder determinar si esas acciones surtieron el efecto esperado. Tomar medidas correctas también hará más fácil la comunicación de objetivos sobre nuevas mejoras.

## 5.2. Enfoque y estructura de las métricas, indicadores y reportes

De los aspectos anteriores, se puede definir el enfoque de las métricas, indicadores o reportes en forma más precisa con los siguientes puntos: [18]

- Generación de un soporte para la toma de decisiones:
- Generación oportuna de reportes, métricas o indicadores representativos de un objetivo en la toma de decisiones, los deben estos ser aceptados además por el tomador de decisiones.
- Generación de información comparativa en base a cierto número de expectativas:
    - la información debe ser establecida o inferida a partir de las métricas, reportes y/o indicadores generados,
    - las expectativas deben ser definidas en forma cuantitativa.
- Generación de información:
    - sobre metas logradas,
    - sobre nuevas iniciativas,
    - para el cliente y el equipo a cargo.

Además, es posible observar la estructura de las métricas, indicadores o reportes utilizando una perspectiva en capas [18]:

- Capa de análisis: método o proceso de cómo usar la información para tomar decisiones.
- Capa de presentación: salidas de las métricas (gráficas y reportes).
- Capa de datos: la información de fondo que soporta la capa de presentación.

- Capa de procesos: el proceso que la información representa y como es obtenida de éste.

## 5.3. El proceso de gestión ágil

En general, las variables utilizadas en proceso de gestión de desarrollo de software son [19]:

- Tiempo disponible para el proyecto.
- Recursos disponibles o los costos asociados al proyecto.
- Alcance, conjunto de funcionalidades o productos que se desarrollarán.
- Calidad (eficacia, resistencia a fallas, eficiencia, etc.).

La constante aparición de cambios o nuevos requerimientos impacta directamente sobre estas variables. En general, esto se ataca de la siguiente forma:

- Tiempo y costo no son en realidad muy flexibles.
- El alcance está fijo por el diseño y el plan definido.
- Variable que sufre: la Calidad, lo que afecta el valor generado para el cliente.
- Lo anterior, genera conflicto, ya que el cliente siempre querrá el mayor valor por sus recursos invertidos y el desarrollador la mayor calidad.

Es por lo anterior que la gestión ágil se enfoca no en ajustar la calidad, sino que el alcance del proyecto, o sea, la cantidad de funcionalidades desarrolladas y las características asociadas a ellas [13]. Para ello se definen plazos de entrega cortos de duración fija (sin atrasos) donde se consideran las siguientes reglas, según el modelo simplificado de Villena [16] de la declaración de deberes y derechos de clientes y desarrolladores planteada por Beck y Fowler [3], tal como se presentan en la Tabla 1:

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

**Tabla 1: Derechos y deberes de clientes y desarrolladores para gestión ágil**

El objetivo es que todos adquieran la confianza de que las metas del proyecto (de negocio y técnicas) son alcanzables.

Es aquí donde las métricas, indicadores y reportes cobran vital importancia, ya que son las herramientas disponibles para generar una base de buenas estimaciones enfocadas en generar valor a corto plazo, ya sea para el cliente o para el mismo equipo de desarrollo. Esto es observable como fundamento de metodologías tales como XP, SCRUM, Crystal Clear, etc. Esto es la base para atacar tres aspectos muy importantes, mencionados anteriormente:

- Lograr un gran poder de estimación que permita una mejor definición del alcance de los proyectos.
- Ser inmune al cambio y poseer una base sólida y ágil de respuesta frente a nuevos requerimientos.
- Y lo más importante, generar valor tanto para el cliente como para el equipo de desarrollo.

De acá surgen preguntas simples y claves sobre lo principal que se espera al momento de medir y lo que las métricas, gráficas y/o reportes que se generen deben responder [14]:

- ¿Se está bien de tiempo con lo que se lleva avanzado?
- ¿Se está aún dentro del presupuesto?

Pero además, medir en forma adecuada generará estimaciones ad-hoc para los objetivos del negocio que se pretenden alcanzar. A partir de ello, una buena estimación debe responder por lo menos a:

- Permitir controlar el proyecto (básico).
- Ser completa y comprensible.
- Ser realista desde el punto de vista de los recursos disponibles.
- Estar basada en un enfoque creíble de tareas y productos medibles.
- Ser uniforme (reproducible utilizando la información equivalente necesaria).
- Considerar las incertidumbres y los riesgos.

Con lo anterior, surge una tercera pregunta a la cual debe responder nuestra forma de medir las cosas:

- ¿Con la información obtenida se es capaz de estimar en forma adecuada?

Sobre este punto, podemos destacar, que la herramienta sobre la cuál se generarán los reportes y las métricas, la Painless Tracking, a parte de su utilidad operativa evidente debe ser también considerada una herramienta académica que permite el aprendizaje de generación de buenas estimaciones. [13]

## 5.4. Estado del arte externo

A continuación se presentan los reportes y métricas recogidos de las distintas metodologías ágiles.

### 5.4.1. Reportes para proyectos ágiles

***Burn-up y Burn-Down Charts*** [5 a 12]

Estos reportes miden la cantidad completada del proyecto versus el tiempo transcurrido. En ellos podemos observar en forma clara y simple el progreso del desarrollo y el valor generado. Para ello, es importante saber elegir una unidad adecuada para medir estos aspectos. Las unidades más utilizadas en este tipo de reporte son: historias de usuario completadas o funcionalidades terminadas. También se puede elegir como unidad el número de tareas terminadas pero sólo como herramienta para medir el desempeño interno del equipo de desarrollo.

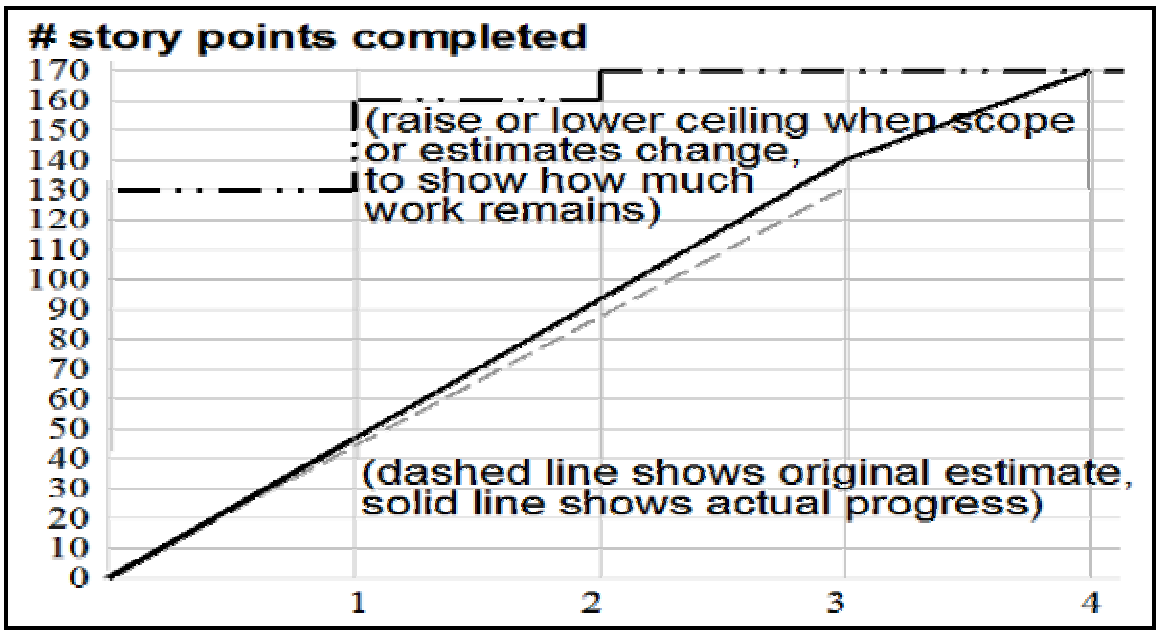


**Ilustración 1 : Burn-up Chart**

La figura anterior muestra un reporte tipo Burn-Up, dónde se puede observar con línea punteada la estimación original y con línea sólida el progreso actual. Se mostrará a continuación como cumple los aspectos mencionados anteriormente.

El “techo” del reporte se puede ir modificando a medida que nuevos requerimientos se agreguen previa negociación.

La unidad de medida seleccionada es las historias de usuarios completadas. Al modificar las tareas dentro de una historia de usuario, puede que se incremente el tiempo necesario para ser terminada, pero sigue siendo la misma historia de usuario. Además, para completar una historia de usuario se requiere una cantidad finita de tiempo no muy prolongada, lo que nos permite una visión temprana de nuestro avance y la posibilidad de tomar las acciones adecuadas en caso de algún problema. De ser muy extenso el tiempo para completar una historia de usuario, probablemente la historia en sí es muy compleja y en realidad se está encapsulando varias historias en una.

Podemos acomodar cuales historias van primero que otras ya que eso no compete al reporte, sino a la planificación del proyecto. Pero en sí, el reporte otorga la posibilidad de subir el techo del plazo final en caso de cualquier eventualidad.

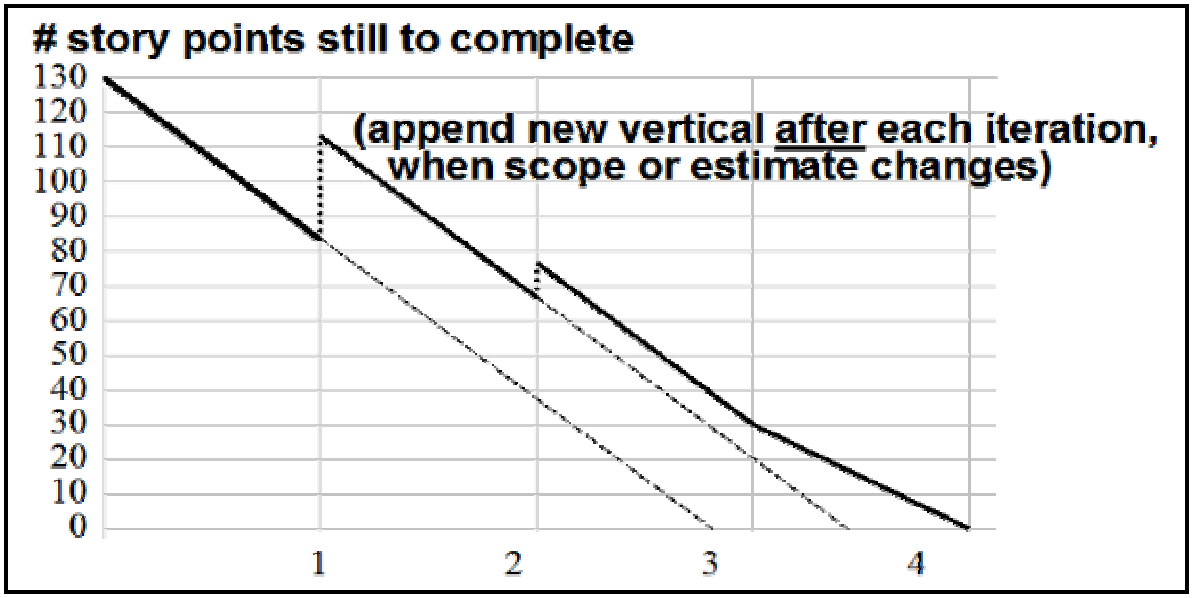


**Ilustración 2 : Burn-down Chart con línea de estimación acomomodada**

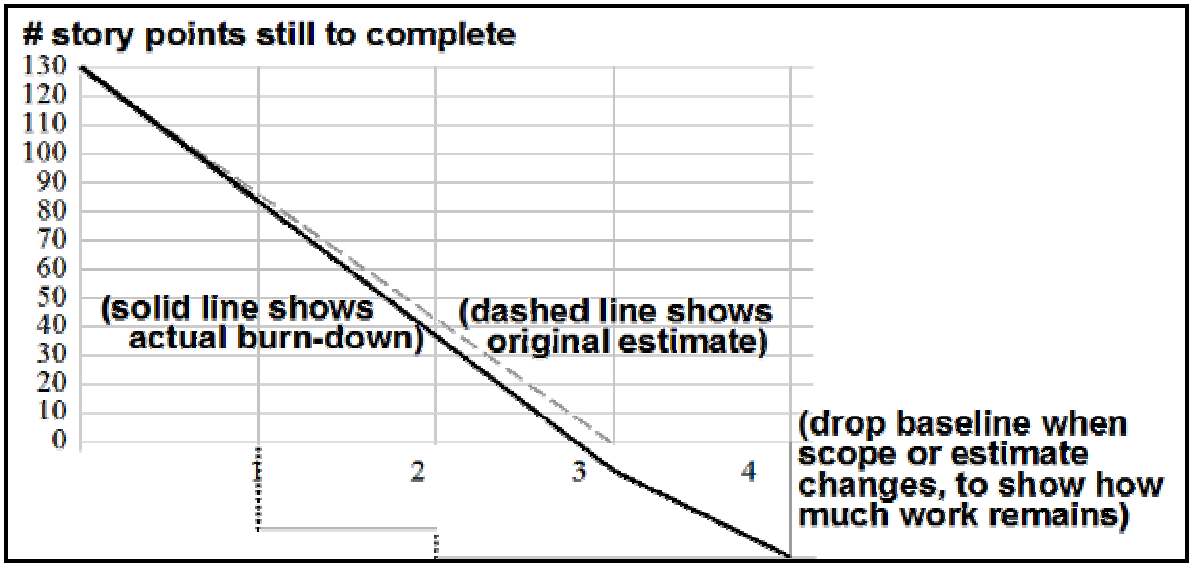


**Ilustración 3: Burn-down Chart con piso acomodado**

Las figuras 2 y 3 muestran los reportes tipo Burn-down. Principalmente estos reportes son usados en vez del tipo Burn-up por la simple razón de proveer una visión emocional más poderosa sobre el avance del proyecto. Esto es porque en vez de mostrar cuanto porcentaje de avance se ha logrado, muestra cuanto falta por terminar.

Se puede ver que entre las figuras 2 y 3 hay una diferencia en cómo es atacado la agregación de nuevos requerimientos. En la figura 2, se manejan nuevas líneas de estimación para cada iteración que presente el proyecto, en cambio, la figura 3 sigue el estilo del reporte tipo Burn-up, pero en vez de modificarse el techo, se modifica el piso del reporte.

Para observar los síntomas de que no va muy bien el proyecto en estos tipo de reporte, basta observar que se está bajo la línea de estimación, en el caso Burn-up, o sobre ella, en el caso Burn-down, a pesar de ir modificando la estimación continuamente.

***Time and Budget Chart*** [14]

Este reporte puede considerarse como una extensión del reporte tipo Burn-up que agrega además el concepto de presupuesto en el área de gestión.

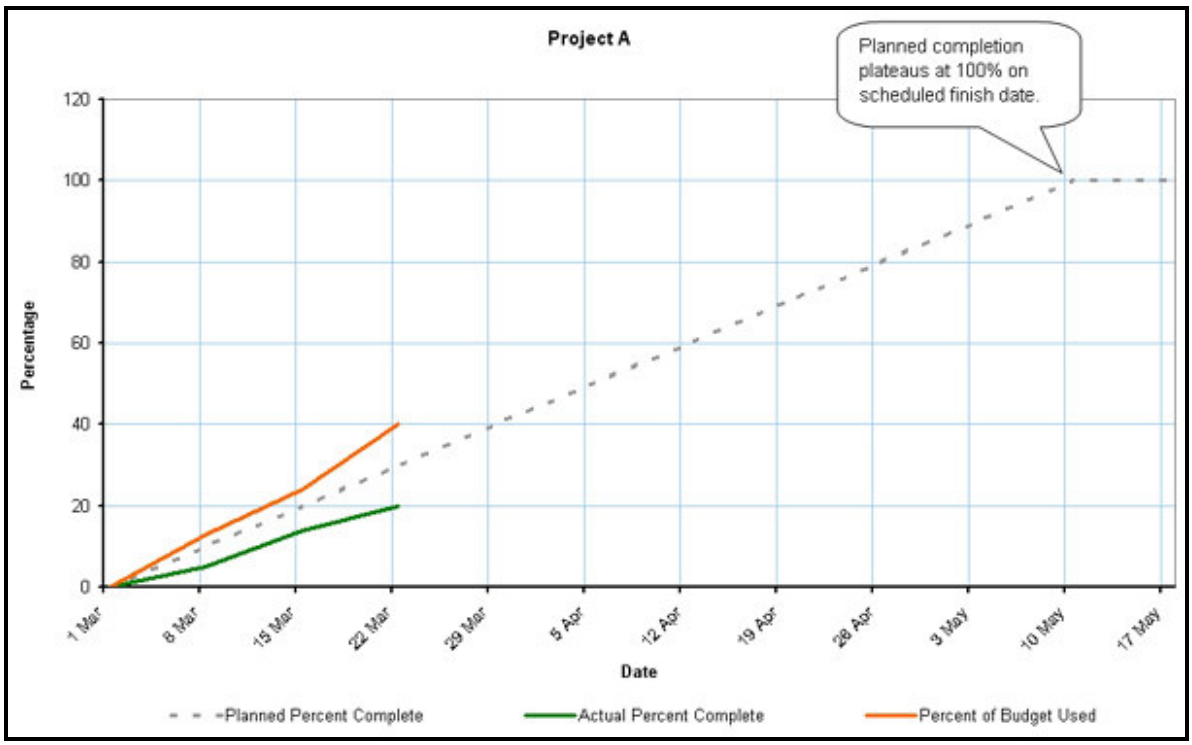


**Ilustración 4: Time and Budget Chart**

La línea segmentada muestra la estimación original e ideal para el proyecto. El eje vertical es usado tanto para mostrar el porcentaje de avance del proyecto, observable a través de la línea verde, como para mostrar que porcentaje del presupuesto se ha utilizado hasta el momento, observable a través de la línea naranja.

Los síntomas a observar de que no va bien o el proyecto serían:

- Va bien el avance del proyecto pero se ha ocupado más del presupuesto estimado en ese momento. Podrían no ser suficientes los recursos para finalizar el proyecto.

- Se está dentro del presupuesto esperado pero el avance del proyecto está bajo lo estimado.
- El presupuesto utilizado está sobre lo esperado y el avance está bajo lo estimado. Éste es el peor de los casos.

***Cummulative Flow Diagram (CFD)*** [5 a 12]

En general, los CFDs muestran una comparativa entre el total a realizar, lo que está en progreso y lo finalizado, en cada punto del tiempo en el cual se está observando. Se puede fragmentar en aún más elementos a comparar, por ejemplo, separar el diagrama en procesos (análisis, diseño, implementación, test, etc.). Pero, como en el caso de los reportes anteriores, es importante saber que unidad elegimos para medir.

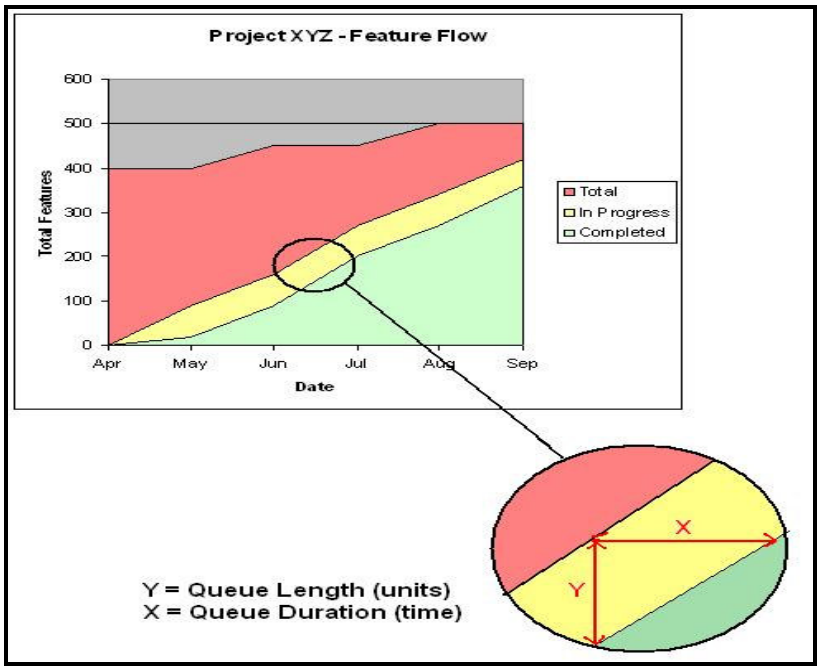


**Ilustración 5 : Cummulative Flow Diagram**

El aporte de este tipo de reportes es que se pueden observar varias cosas a simple vista, en base a las cuales, se pueden tomar decisiones adecuadas en el caso que se presente algún problema. Si se toma como unidad de medida las funcionalidades entregadas, se tendría:

- Tiempo de demora en la entrega de las funcionalidades. Observable en X.
- En cuántas funcionalidades se está trabajando. Observable en Y.
- Posibles cuellos de botella. Generalmente esto es observable cuando se presentan áreas delgadas bajo un área más grande. Esto quiere decir, que para pasar de una etapa a otra, se está realizando trabajo muy puntual, el cual debe ser investigado dado que podría generar algún problema si se trata de algún proceso delicado.

***Trend analysis chart*** [5 a 12]

Básicamente, en este reporte se desglosan las funcionalidades o productos por algún parámetro en especial, entre los cuales se pueden destacar: la prioridad, iteración a la que pertenecen o simplemente su complejidad.

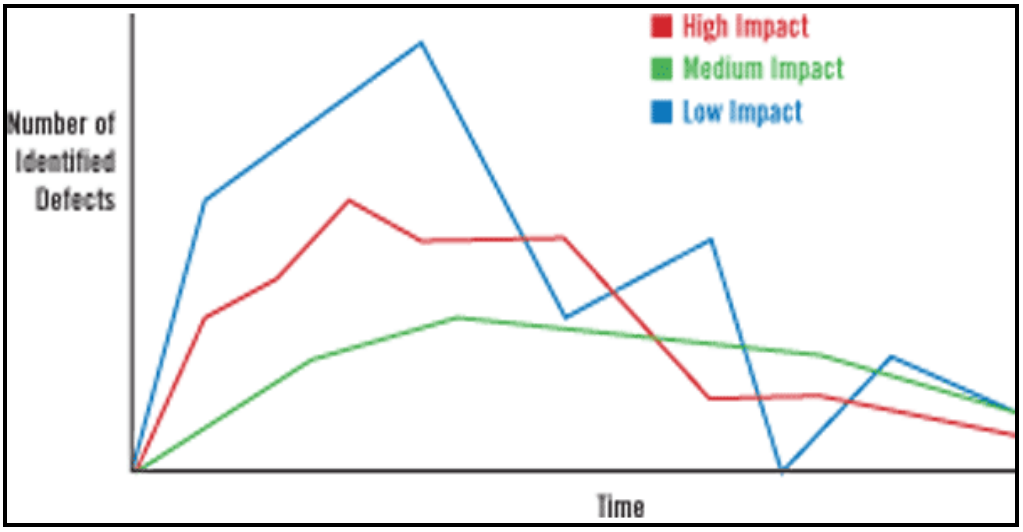


**Ilustración 6 : Trend Analysis Chart**

En general, proveen la posibilidad de generar análisis comparativo, dado el parámetro de desglose. Con ello se puede, por ejemplo, observar que tan bien se están priorizando las cosas o si la carga de funcionalidades es balanceada para cada iteración.

***Agile-V Scorecard*** [5 a 12]

Es un reporte que se genera a partir de un burn-up y un burn-down chart asociado al proyecto. Pero en cada uno se presenta un aspecto distinto. En el burn-down se muestran las horas restantes para terminar el proyecto mientras que en el burn-up se muestran las funcionalidades o productos terminados.

Con este reporte se puede observar, por ejemplo, que faltan muchas horas de desarrollo pero se han logrado muchas funcionalidades, probablemente de complejidad menor. De todos modos, esto depende que unidad de medición se use para generar el reporte.

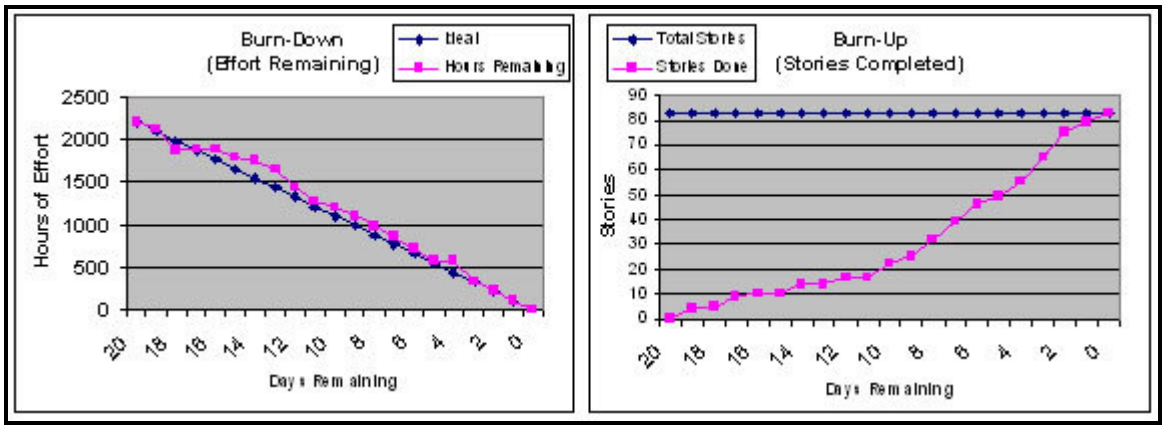


**Ilustración 7 : Agile-V Scorecard**

### 5.4.2. Métricas para proyectos ágiles

***Factor de corrección de estimación.***

Corresponde a la proporción entre el total de tiempo estimado y el total del tiempo realmente utilizado, para las tareas ya finalizadas. Cuando no existan tareas finalizadas, se asume un factor 1 como inicial (100%). Al aparecer tareas finalizadas con este factor es posible corregir las nuevas estimaciones que se vayan presentando y con ello, minimizar la diferencia entre el tiempo que realmente se utilizará para desarrollar las tareas. Ojo, que el valor es sobre las tareas, para tener una mejor precisión de este valor.

$$\frac{\Sigma\ (\text{Tiempo Estimado})_{[\text{tareas terminadas}]}}{\Sigma\ (\text{Tiempo Real})_{[\text{tareas terminadas}]}}$$

***Velocidad de desarrollo.***

Corresponde a la proporción entre el tiempo utilizado en funcionalidades ya terminadas y el total del tiempo utilizado en todo el proyecto. Esto quiere decir que en el caso de no existir productos terminados, la velocidad de desarrollo correspondiente es cero. Esto es así dado que esta métrica pretende principalmente indicar el valor entregado al cliente con el desarrollo en proceso.

$$\frac{\Sigma\ (\text{Tiempo Real})_{[\text{funcionalidades terminadas}]}}{\Sigma\ (\text{Tiempo Real})_{[\text{proyecto}]}}$$

***Estimated Time of Arrival (E.T.A.)***

Corresponde al factor entre el tiempo restante en tareas actualmente reestimadas (estimación original si no han sido iniciadas) y el inverso del factor de corrección de estimación. Con este valor se puede observar, en forma aproximada, el momento de finalización del proyecto. Se utilizan las últimas estimaciones para tener un valor más certero.

$$\frac{\Sigma\ (\text{Tiempo Restante})_{[\text{tareas estimadas}]}}{(\text{Factor de Corrección de Estimación})}$$

***Avance por funcionalidad: tiempo estimado vs. real.***

Corresponde a la proporción entre el tiempo estimado actual y el tiempo real utilizado para cada funcionalidad en proceso de desarrollo y cada funcionalidad finalizada. Se utiliza la última estimación para obtener una aproximación mejor. Ésta métrica es la que informa básicamente cuanto se ha avanzado por cada funcionalidad o producto del proyecto.

Para funcionalidades terminadas o en avance:

$$\frac{\text{Tiempo Real Ocupado}}{\text{Tiempo Estimado Actual}}$$

***Avance por funcionalidad: tiempo estimado original vs. estimado actual.***

Corresponde a la proporción entre el tiempo estimado inicial y el tiempo estimado actual para cada funcionalidad que esté en proceso de desarrollo. Este factor es principalmente didáctico y muestra la capacidad de predicción del usuario o el grupo que esté realizando la gestión a través de la interfaz de gestión.

Para funcionalidades en avance:

$$\frac{\text{Tiempo Estimado Original}}{\text{Tiempo Estimado Actual}}$$

## 5.5. Estado del arte de la Painless Tracking

### 5.5.1. Painless Tracking versión Excel

A continuación se describe la estructura y capacidades actuales de la herramienta Painless Tracking en su versión Excel. Principalmente consta de dos sectores diferenciables:

1. Sector operativo: es donde se lleva la constancia de los avances diarios y el manejo tanto de las funcionalidades con las que hay que cumplir como las tareas asociadas a desarrollar para lograrlo.
2. Métricas y reportes: es donde se visualizan los avances y las proyecciones del trabajo realizado.

Ahora se procede a detallar cada uno de ellos.

### 5.5.2. Sector operativo

***Plan***

Es la página principal de la herramienta donde se ingresan las tareas del proyecto. Presenta dos sectores bien definidos: plan (propiamente tal) y un sector que se definirá como dashboard desde ahora.

En el sector del plan es donde se ingresa:

- la iteración en la que se está trabajando,
- los productos, que engloban una o más tareas,
- las tareas para cada producto,
- la prioridad para cada tarea (fundamental, importante o deseable)
- un ID Tarea el cual es utilizado como identificador de la tarea para el alineamiento del Plan con el Tracking (revisar Tracking),
- una descripción de la tarea (también usable para ingresar observaciones),
- un responsable de la tarea y,
- una estimación inicial para cada tarea,

El sector dashboard es calculado automáticamente, y en el se reflejan los siguientes datos:

- la última fila en el Tracking donde se ingresó valores para la tarea,
- una fecha asociada a la fila mencionada arriba,
- una campo que indica el tiempo ocupado actualmente en la tarea,
- una campo que indica la última reestimación de tiempo restante para finalizar la tarea y,

- el total actual que incluye la suma de los dos campos anteriores.

En las figuras siguientes se muestran ambos sectores:

| Iteración | Producto | Tarea | Prioridad | ID Tarea | Descripción Tarea | Responsable | Asignados |
|---|---|---|---|---|---|---|---|
| 0 | Hola mundo | Obtener acceso a recursos de la empresa | 1 | hola.1 | | mauro | mauro |
| 0 | Hola mundo | Instalación de herramientas de desarrollo | 1 | hola.2 | | mauro | mauro |
| 0 | Hola mundo | Aprendizaje de posibles herramientas de desarrollo | 1 | hola.3 | Hacer tutoriales | mauro | mauro |
| 0 | Hola mundo | Análisis y elección de herramientas de desarrollo | 1 | hola.4 | | mauro | mauro |
| 0 | Hola mundo | Probar herramientas de desarrollo (hola mundo) | 1 | hola.5 | | mauro | mauro |
| 1 | Entendimiento del problema | Reuniones con cliente para especificar mejor el problema | 1 | entend.1 | | cris | cris |
| 1 | Entendimiento del problema | Catalogar y analizar los servicios existentes | 1 | entend.2 | | carlos | carlos |
| 1 | Entendimiento del problema | Investigacion sobre SNMP | 2 | entend.4 | | felipe | felipe |
| 1 | Entendimiento del problema | Investigación sobre patrones existentes | 1 | entend.3 | | carlos | carlos |

**Tabla 2: Interfaz de Planificación (sector plan) Painless Tracking Excel**

| Iteración | Producto | Prioridad | ID Tarea | Orig. Est. | Last Updated Row | Fecha Ultima Actividad | Ocup. Actual. | Rest. Est. | Total Actual. |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Hola mundo | 1 | hola.1 | 14 | 295 | 15-09-06 | 9 | 1 | 10 |
| 0 | Hola mundo | 1 | hola.2 | 14 | 51 | 17-08-06 | 3 | 0 | 3 |
| 0 | Hola mundo | 1 | hola.3 | 50 | 114 | 23-08-06 | 58,5 | 0 | 58,5 |
| 0 | Hola mundo | 1 | hola.4 | 28 | 293 | 15-09-06 | 0 | 0 | 0 |
| 0 | Hola mundo | 1 | hola.5 | 3,5 | 101 | 23-08-06 | 8 | 0 | 8 |
| 1 | Entendimiento del problema | 1 | entend.1 | 6 | 41 | 16-08-06 | 6 | 0 | 6 |
| 1 | Entendimiento del problema | 1 | entend.2 | 28 | 152 | 29-08-06 | 22,5 | 0 | 22,5 |
| 1 | Entendimiento del problema | 2 | entend.4 | 4 | 34 | 16-08-06 | 0,5 | 0 | 0,5 |
| 1 | Entendimiento del problema | 1 | entend.3 | 14 | 48 | 17-08-06 | 3 | 0 | 3 |
| 1 | Entendimiento del problema | 2 | entend.5 | 5 | 122 | 24-08-06 | 3,5 | 0 | 3,5 |

**Tabla 3: Interfaz de Planificación (sector dashboard) Painless Tracking Excel**

### *Tracking*

Es donde se lleva el registro periódico de los avances de las tareas. Cuenta con los siguientes campos para ingresar información:

- **Producto**: alineado con el mismo producto en la página Plan.
- **Iteración**: alineada con la iteración a la cual pertenece la tarea.
- **Fecha**: fecha en que se ingreso el registro correspondiente para esta tarea.
- **ID Tarea**: también alineado a la página Plan.
- **Ocupado**: cuánto tiempo se utilizó en el desarrollo de esta tarea para este registro ingresado.
- **Restante Estimado**: desde este registro de avance de la tarea cuánto estima el desarrollador que falta para completar la tarea. Al ingresar un valor 0 se finaliza la tarea.
- **Desarrollador**: responsable de la tarea.
- **Qué se hizo**: referente al registro para esta tarea.
- **Qué falta por hacer**: para completar la tarea.

Los campos Fecha, ID Tarea, Ocupado y Restante Estimado están perfectamente alineados con la página Plan y son utilizados para los cálculos presentes en la página Plan – Dashboard mediante funciones y macros ad-hoc. La figura siguiente muestra un Tracking habitual:

| Producto | Iteración | Fecha | ID Tarea | Ocupado | Restante Estimado | Desarrollador | Qué se hizo | Qué falta por hacer |
|---|---|---|---|---|---|---|---|---|
| Otros | 0 | 15-08-06 | otros | 5 | 0 | felipe | Feriado | |
| Otros | 0 | 15-08-06 | otros | 7,5 | 0 | carlos | Feriado | |
| Otros | 0 | 15-08-06 | otros | 3 | 0 | tguridi | Feriado | |
| Otros | 0 | 15-08-06 | otros | 5 | 0 | ftroncos | Feriado | |
| Hola mundo | 0 | 14-08-06 | hola.1 | 1,5 | 12,5 | carlos | Llegamos, mi computador funciona | Tarjeta entrada |
| Hola mundo | 0 | 14-08-06 | hola.2 | 1 | 13 | carlos | Instale turbogears, actualizacion de python en mi Mac | Falta instalar django y ver las conexiones al servidor de aplicaciones |
| Otros | 0 | 14-08-06 | otros | 0,5 | 0 | carlos | Tiempo de espera de llegada | |
| Hola mundo | 0 | 14-08-06 | hola.3 | 1 | 83 | carlos | Pruebas de Python, lectura de manuales | Seguir estudiando el ambiente python. |

**Tabla 4: Tracking**

### 5.5.3. Métricas y reportes

***Métricas***

A continuación se describen cada uno de los valores presentes en esta página de la herramienta junto a la figura:

| Datos del proyecto | | Recursos disponibles para el proyecto. |
|---|---|---|
| Cantidad de Desarrolladores | 7 | Con cuantos desarrolladores cuenta este proyecto. |
| Horas de trabajo por desarrollador por semana | 16 | Horas disponibles ideales de desarrollo de un desarrollador. |

| Tareas Finalizadas | | Las siguientes métricas se calculan sobre tareas terminadas.<br>Los valores son obtenidos desde la página Plan sector Dashboard. |
|---|---|---|
| Total Estimaciones | 575,5 | Suma de las estimaciones originales de todas las tareas finalizadas. |
| Total Ocupado | 487,5 | Suma de los Totales Actualizados de todas las tareas finalizadas. |
| Velocidad relativa | 118% | Relación entre el tiempo estimado y el tiempo realmente ocupados en tareas finalizadas. Ej.: "50%" implica que se avanza a la mitad de la rapidez estimada. |

| Disponibilidad | | Los valores siguientes se enfocan en generar valores comparativos entre lo planificado, lo real y lo no planificado. |
|---|---|---|
| Total ocupado en tareas no planificadas | 181 | A partir de la página Tracking se calcula el tiempo utilizado en el item “Otros” que corresponde a las tareas no planificadas. |
| Total de horas trabajadas | 597,25 | Tiempo utilizado en todas las tareas trabajadas. |
| Disponibilidad | 70% | Relación entre lo planificado y no planificado |
| Velocidad de Desarrollo Calculada | 82% | Se puede usar para ajustar la velocidad estimada de desarrollo |
| Velocidad de Desarrollo Estimada | 70% | Aquí se define la velocidad que se cree será la correcta. |

| Tiempo Estimado de Llegada | | Estos valores se enfocan en estimar cuanto falta para terminar el proyecto. |
|---|---|---|
| Horas restantes planificadas | 224,0 | Suma de los tiempos correspondientes a las tareas no iniciadas (Tiempo Estimado Inicial) y las tareas en avance (en el caso de presentar Restante Estimado se usa ese valor sino se usa el Tiempo Estimado Inicial). |
| Tiempo Estimado de Arribo (horas) | 320,0 | Tiempo para finalizar el proyecto considerando un solo desarrollador con dedicación completa. Esta relación se genera a partir de las Horas restantes planificadas y la Velocidad de Desarrollo Calculada o Estimada, por lo que, puede variar su valor dependiendo que se use. |
| Tiempo Estimado de Arribo Corregido (semanas) | 2,86 | Cuantas semanas faltan para terminar el proyecto considerando los recursos disponible Número de desarrolladores y jornada semanal. |

| Ultima Fila de Tracking reportada | 437 | Último registro de una tarea. No necesariamente es el término del proyecto. |
|---|---|---|

**Tabla 5: Métricas**

## ***Reportes***

### Avance de Productos

Consiste en un reporte generado a través de una tabla pivote utilizando las funcionalidades de las tablas dinámicas de Excel. Informa el porcentaje de avance para cada producto a partir del tiempo utilizado actualmente para su desarrollo y el tiempo restante reestimado. A partir de esa información calcula los totales de avance por iteración y en general para todo el proyecto.

| | | Datos | | |
|---|---|---|---|---|
| Iteración | Producto | Suma de Porcentaje Avance | Suma de Ocup. Actual. | Suma de Rest. Est. |
| 1 | Entendimiento del problema | 100,0% | 109,85 | 0 |
| | Sistema para agregar nuevos tests | 100,0% | 186,4 | 0 |
| | Tests reales | 100,0% | 9 | 0 |
| | daemon que corra scripts | 100,0% | 49,5 | 0 |
| | Planificacion | 96,9% | 15,5 | 0,5 |
| Total 1 | | 99,9% | 370,25 | 0,5 |
| 2 | Vista de tags | 50,9% | 14 | 13,5 |
| | Correción de problemas varios | 69,2% | 4,5 | 2 |
| | Funcionalidades interfaz | 68,0% | 51 | 24 |
| | Funcionalidades demonio | 83,7% | 10,25 | 2 |
| | Reportes | 46,7% | 22,75 | 26 |
| | Documentos | 3,3% | 2,5 | 73,5 |
| | Reuniones con Hernán | 41,1% | 9,75 | 14 |
| | Reuniones grupo | 100,0% | 19,5 | 0 |
| | Upgrade interfaz | 100,0% | 5,5 | 0 |
| Total 2 | | 39,1% | 142,75 | 222,5 |
| Total general | | 69,7% | 513 | 223 |

**Tabla 6: Reporte de Avance por Productos**

## Horas trabajadas

Este reporte presenta, también a través de una tabla pivote, el tiempo trabajado de cada desarrollador por día e iteración, calculando totales de tiempo trabajado en la iteración y el proyecto por cada desarrollador, para cada día e iteración de todos los desarrolladores y el total del tiempo trabajado en el proyecto.

| Suma de Ocupado | | Desarrollador | | | | |
|---|---|---|---|---|---|---|
| Iteración | Fecha | felipe | carlos | tguridi | ftroncos | Total general |
| 1 | Mié 16 Ago | 2 | | | | 2 |
| | (en blanco) | | | | 4 | 4 |
| | Jue 17 Ago | 3 | | | | 3 |
| | Lun 21 Ago | | 4 | 5 | 2,5 | 11,5 |
| | Mar 22 Ago | 5 | 5,5 | 3 | 1 | 14,5 |
| | Mié 23 Ago | 6,5 | 2 | 5 | | 13,5 |
| | Jue 24 Ago | 5,5 | | 3 | 5 | 13,5 |
| | Vie 25 Ago | | 4,5 | | 3 | 7,5 |
| | Lun 28 Ago | 5,5 | 6 | 5 | | 16,5 |
| | Mar 29 Ago | 5 | 7,5 | 3 | 5,5 | 21 |
| | Mié 30 Ago | 4 | | 5 | | 9 |
| | Jue 31 Ago | | | 3 | 6,5 | 9,5 |
| | Vie 01 Sep | 1,5 | 2,5 | | 4 | 8 |
| | Lun 04 Sep | 5,75 | 8,5 | 5 | 3 | 22,25 |
| | Mar 05 Sep | 5,5 | 7,5 | 3 | 5 | 21 |
| | Mié 06 Sep | 5 | | 5 | | 10 |
| | Jue 07 Sep | | | 3 | 5 | 8 |
| | Vie 08 Sep | | | | 3 | 3 |
| | Lun 11 Sep | 6 | 8,5 | 5 | 4,5 | 24 |
| | Mar 12 Sep | 5,5 | 7,5 | 3 | 4 | 20 |
| | Mié 13 Sep | 3,5 | | 5 | 3 | 11,5 |
| | Jue 14 Sep | | | 3 | 4,5 | 7,5 |
| Total 1 | | 69,25 | 64 | 64 | 63,5 | 260,75 |
| 2 | Lun 25 Sep | 9 | 8,5 | 4 | 4 | 25,5 |
| | Mar 26 Sep | | 7,5 | 3 | | 10,5 |
| | Mié 27 Sep | | | 5,5 | 4 | 9,5 |
| | Jue 28 Sep | | | 3,5 | 6 | 9,5 |
| | Vie 29 Sep | 6 | | | 2 | 8 |
| | Lun 02 Oct | 5,5 | 9 | 4 | 4 | 22,5 |
| | Mar 03 Oct | | 7,5 | 3,25 | 4,5 | 15,25 |
| | Mié 04 Oct | | | 5,5 | 3 | 8,5 |
| Total 2 | | 20,5 | 32,5 | 28,75 | 27,5 | 109,25 |
| Total general | | 89,75 | 96,5 | 92,75 | 91 | 370 |

**Tabla 7: Reporte de Horas trabajadas**

## Horas trabajadas por Producto

El reporte es similar al anterior pero se hace un cambio de ejes para separar el trabajo de cada desarrollador en los productos en que se divide el proyecto, con lo que se agregan también los totales para cada producto en la iteración y en el proyecto en general.

| Suma de Ocupado | | Itera-ción | Fecha | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | | | | Total 1 | 2 | | | Total 2 | Total general |
| Desarrollador | Producto | 25-08-06 | 28-08-06 | 29-08-06 | 30-08-06 | | 29-09-06 | 02-10-06 | 03-10-06 | | |
| felipe | Otros | | 1 | | | 1 | 2,5 | 0,5 | | 3 | 4 |
| | Sistema para agregar nuevos tests | | 4,5 | 5 | 4 | 13,5 | | | | | 13,5 |
| | Vista de tags | | | | | | | 2 | | 2 | 2 |
| | Reuniones grupo | | | | | | | 2 | | 2 | 2 |
| | Correción de problemas varios | | | | | | 1 | | | 1 | 1 |
| | Funcionalidades interfaz | | | | | | | 1 | | 1 | 1 |
| | Upgrade interfaz | | | | | | 2,5 | | | 2,5 | 2,5 |
| Total felipe | | | 5,5 | 5 | 4 | 14,5 | 6 | 5,5 | | 11,5 | 26 |
| carlos | Otros | | | 2 | | 2 | | | | | 2 |
| | Entendimiento del problema | 4,5 | 5 | 5,5 | | 15 | | | | | 15 |
| | Sistema para agregar nuevos tests | | 1 | | | 1 | | | | | 1 |
| | Vista de tags | | | | | | | | 1 | 1 | 1 |
| | Reuniones grupo | | | | | | | 3,5 | | 3,5 | 3,5 |
| | Funcionalidades demonio | | | | | | | | 1 | 1 | 1 |
| | Reportes | | | | | | | 5,5 | 5,5 | 11 | 11 |
| Total carlos | | 4,5 | 6 | 7,5 | | 18 | | 9 | 7,5 | 16,5 | 34,5 |
| Total general | | 4,5 | 11,5 | 12,5 | 4 | 32,5 | 6 | 14,5 | 7,5 | 28 | 60,5 |

**Tabla 8: Reporte de Horas Trabajadas por Producto**

*Comparación de tiempos*

Este reporte simplemente presenta una comparativa entre el Tiempo Estimado Inicial y el Tiempo Ocupado Actual, para cada producto, cada iteración y el proyecto en general. Presenta sólo suma de los campos mencionados en general para una vista rápida con menos detalles que los reportes anteriores.

| | | Datos | |
|---|---|---|---|
| Iteración | Producto | Suma de Ocup. Actual. | Suma de Orig. Est. |
| 0 | Hola mundo | 78,5 | 109,5 |
| Total 0 | | 78,5 | 109,5 |
| 1 | Entendimiento del problema | 109,85 | 98 |
| | Sistema para agregar nuevos tests | 186,4 | 227 |
| | Tests reales | 9 | 26 |
| | daemon que corra scripts | 49,5 | 61 |
| | Planificacion | 15,5 | 10 |
| Total 1 | | 370,25 | 422 |
| 2 | Vista de tags | 14 | 27 |
| | Correción de problemas varios | 4,5 | 11 |
| | Funcionalidades interfaz | 51 | 39 |
| | Funcionalidades demonio | 10,25 | 18 |
| | Ejecución concurrente | 2 | 4 |
| | Diseño test | 1 | 5 |
| | Reportes | 22,75 | 36 |
| | Periodicidad | 0 | 27 |
| | Alarmas | 0 | 30 |
| | Documentos | 2,5 | 76 |
| | Planificación Iteración 3 | 0 | 10 |
| | Reuniones con Hernán | 9,75 | 28 |
| | Reuniones grupo | 19,5 | 28 |
| | Upgrade interfaz | 5,5 | 2 |
| Total 2 | | 142,75 | 341 |
| 3 | Funcionalidades interfaz | 5,75 | 1 |
| Total 3 | | 5,75 | 1 |
| Total general | | 597,25 | 873,5 |

**Tabla 9: Comparación de Tiempos**

*Gráfico de Comparación de Tiempos*

Versión gráfica del reporte anterior.

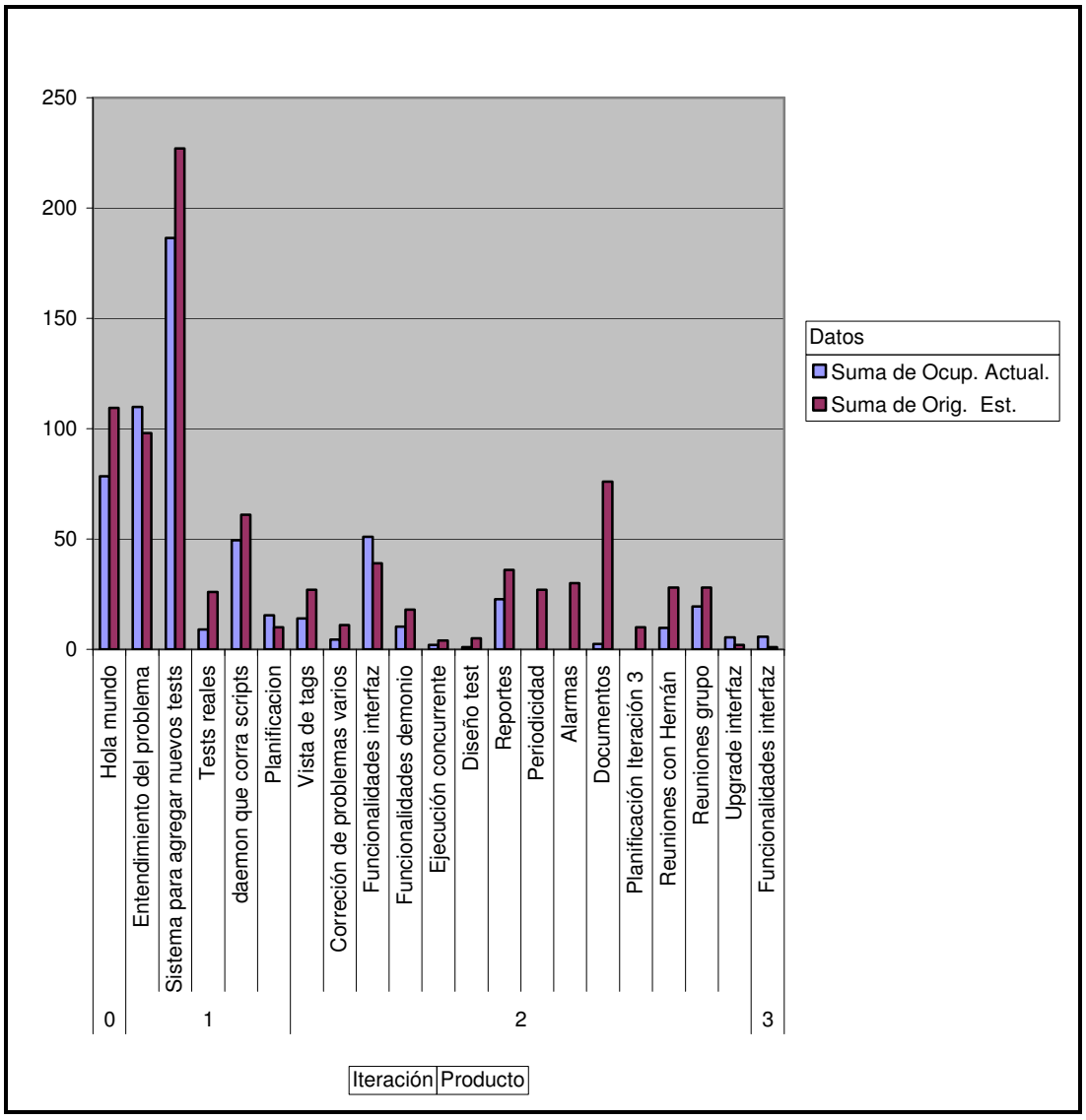


**Ilustración 8 : Gráfico de Comparación de Tiempos**

*Plan a priori*

Este reporte presenta un resumen del Tiempo Estimado Inicial total para cada producto por iteración y el proyecto entero, además de los totales para cada iteración y el proyecto en general.

| Suma de Orig. Est. | Iteración | | | | |
|---|---|---|---|---|---|
| Producto | 0 | 1 | 2 | 3 | Total general |
| Hola mundo | 109,5 | | | | 109,5 |
| Entendimiento del problema | | 98 | | | 98 |
| Sistema para agregar nuevos tests | | 227 | | | 227 |
| daemon que corra scripts | | 61 | | | 61 |
| Tests reales | | 26 | | | 26 |
| Vista de tags | | | 27 | | 27 |
| Planificacion | | 10 | | | 10 |
| Reuniones con Hernán | | | 28 | | 28 |
| Reuniones grupo | | | 28 | | 28 |
| Correción de problemas varios | | | 11 | | 11 |
| Funcionalidades interfaz | | | 39 | 1 | 40 |
| Ejecución concurrente | | | 4 | | 4 |
| Diseño test | | | 5 | | 5 |
| Funcionalidades demonio | | | 18 | | 18 |
| Upgrade interfaz | | | 2 | | 2 |
| Documentos | | | 76 | | 76 |
| Reportes | | | 36 | | 36 |
| Periodicidad | | | 27 | | 27 |
| Alarmas | | | 30 | | 30 |
| Planificación Iteración 3 | | | 10 | | 10 |
| Total general | 109,5 | 422 | 341 | 1 | 873,5 |

**Tabla 10: Plan a Priori**

*Gantt a Posteriori*

Este reporte informa sobre el tiempo real ocupado para cada producto por día, por iteración y el proyecto completo hasta el momento. Además informa sobre los totales del tiempo ocupado para cada día, iteración y proyecto en todos los productos.

| Suma de Ocupado | Iteración | Fecha | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | | | Total 1 | 2 | | | | Total 2 | Total general |
| Producto | | 23-08-06 | 24-08-06 | 25-08-06 | 28-08-06 | | 25-09-06 | 26-09-06 | 27-09-06 | 28-09-06 | | |
| Otros | 7,5 | 1 | 0,5 | 4 | | 13 | 29,5 | 8,5 | 19 | 2 | 59 | 72 |
| Hola mundo | 3 | | | | | 3 | | | | | | 3 |
| Entendimiento del problema | 1,5 | 2 | 4,5 | 5 | | 13 | | | | | | 13 |
| Sistema para agregar nuevos tests | 21,5 | 10,5 | 12,5 | 25,5 | | 70 | 2 | | | | 2 | 72 |
| Planificacion | | | | | | | | | 1,5 | | 1,5 | 1,5 |
| (en blanco) | | | | | | | 2,5 | | | | 2,5 | 2,5 |
| Reuniones con Hernán | | | | | | | 5 | | | | 5 | 5 |
| Reuniones grupo | | | | | | | 7,5 | | | | 7,5 | 7,5 |
| Correción de problemas varios | | | | | | | 0,5 | | 0,75 | | 1,25 | 1,25 |
| Funcionalidades interfaz | | | | | | | | 2 | 6,75 | 7,5 | 16,25 | 16,25 |
| Ejecución concurrente | | | | | | | 0,5 | | | | 0,5 | 0,5 |
| Diseño test | | | | | | | | | 1 | | 1 | 1 |
| Funcionalidades demonio | | | | | | | | | 0,5 | | 0,5 | 0,5 |
| Total general | 33,5 | 13,5 | 17,5 | 34,5 | | 99 | 47,5 | 10,5 | 29,5 | 9,5 | 97 | 196 |

**Tabla 11: Gantt a Posteriori**

# 6. Trabajo Realizado

## 6.1. Colaboración con las otras líneas de desarrollo de la Painless Tracking Web

### 6.1.1. Prototipo Plataforma Pylons Painless Tracking versión Web CC62V

Se trabajó con los alumnos del curso CC62V observando el trabajo realizado por ellos durante el primer semestre. Se plantearon algunas sugerencias y principalmente se orientó en el uso y funcionamiento de la Painless Tracking versión Excel.

El trabajo realizado por ellos se centró en una traducción literal muy básica de la herramienta en su formato actual. Este desarrollo ya ha sido recibido de los alumnos y aportará siendo la base sólida de código funcional sobre la cual construir las interfaces operativa y de gestión. En esta instancia no se contribuyó en el diseño de la interfaz de la aplicación ni la implementación de esta misma, esto queda para el trabajo a realizar durante los siguientes pasos de la metodología. (Ver sección de Metodología, punto 5.6)

Basada en la versión Excel, se desarrolló en el curso, la primera versión Web de esta herramienta. El modelo de datos correspondiente al trabajo en conjunto con los alumnos se puede observar en la siguiente figura:

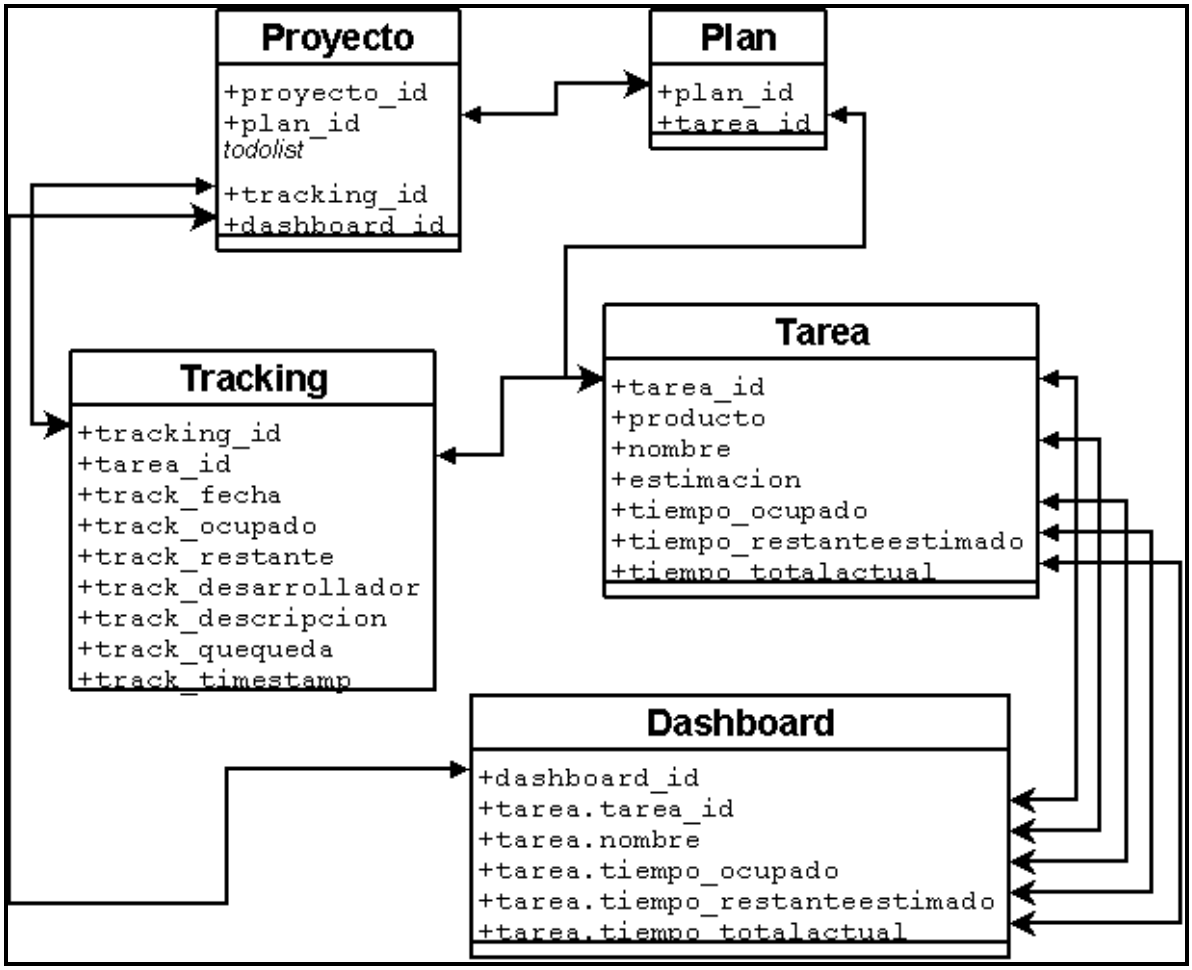


**Ilustración 9 : Modelo de Datos Painless Tracking versión Web CC62V**

Se pueden observar los siguientes elementos trasladados de la versión Excel (con algunas diferencias):

- El plan, el dashboard y el tracking son tratados en forma separada, no así en la versión Excel, donde se puede observar que el plan y dashboard trabajan en forma conjunta.
- El modelo soporta la operabilidad entre varios proyectos a diferencia de la versión Excel, donde se debe utilizar una instancia nueva de la herramienta (planilla) para cada proyecto.
- Las tareas poseen identificadores más consistentes permitiendo la futuras operaciones sobre ellas aparte de la creación y lectura.

Deficiencias del modelo:

- No es posible determinar el orden de creación de las tareas.
- No es posible obtener un historial de reestimaciones hechas sobre las tareas.

### 6.1.2. Painless Tracking en conjunto con línea de usabilidad y adopción

Se trabajó en conjunto con los alumnos memoristas Matías Toro, encargado de la línea de usabilidad, el alumno Roberto Carrasco, encargado de la línea de adopción y el Profesor Agustín Villena, creador de la herramienta Painless Tracking y tutor de las diferentes memorias referidas a esas líneas, incluyendo esta misma, la línea de gestión.

A través de talleres de diseño y entrevistas se aportaron las experiencias sobre el uso de la herramienta Painless Tracking Excell y ideas generales, críticas y mejoras sobre esta misma, con el propósito de que las diferentes líneas confluyeran en una futura versión mejorada de la herramienta: Painless Tracking Web.

**Ilustración 10: Trabajo conjunto**

## 6.2. Reportes y Métricas ágiles implementadas

### 6.2.1. Elección de los reportes y métricas a utilizar

***Aspectos de selección***

Después de hacer una revisión por distintas metodologías ágiles de desarrollo, para seleccionar los reportes y métricas a ser incluidos en la interfaz de gestión de la Painless Tracking Web, se han considerado los siguientes aspectos:

1. Soporte para la agregación de nuevos requerimientos o tareas.
2. Dimensión adecuada de las unidades de medición. Ni tan pequeñas, para controlar adecuadamente el reporte, ni tan grandes para permitir visibilidad a corto plazo.
3. Soporte para el cambio de prioridades.

***Reportes y métricas seleccionados***

| | **Agregación requerimientos nuevos** | **Unidades de medición adecuadas** | **Cambio de prioridades** |
|---|---|---|---|
| Burn-up & Burn-down charts | Basta con agrandar el techo del total de funcionalidades a desarrollar | Se elige como unidad el número de funcionalidades, cuantificando aquellas finalizadas | Son independientes de este tipo de cambios dada la unidad de medida elegida |
| Time & Budget charts | Basta con agrandar el techo del total de funcionalidades a desarrollar | Se elige como unidad el porcentaje de funcionalidades y el porcentaje de recursos utilizados, para generar la comparativa por definición de este tipo de reportes | Varia el porcentaje de funcionalidades/recursos medidos pero el reporte es capaz de ajustarse |

| | | | |
|---|---|---|---|
| CFD | Basta con agrandar el techo del total de funcionalidades a desarrollar | Se elige como unidad el número de funcionalidades, cuantificando aquellas finalizadas, en avance y no comenzadas | Es independiente de este tipo de cambios dada la unidad de medida elegida |
| Trend analysis charts | Se puede agrandar el techo del reporte para cuantificar un aumento en el número de funcionalidades | Se elige como unidad el número de funcionalidades, cuantificando aquellas finalizadas | Falla al cambiar las prioridades. Por ejemplo al cambiar todas las prioridades a críticas, este reporte entregaría casi nada de valor |
| Agile-V Scorecard | Al ser combinación de Burn-down y burn-up, cumple la condición | No es lo ideal utilizar una unidad distinta como lo hace este reporte. Se genera más valor utilizando una unidad equivalente para ambos. | Es independiente de este tipo de cambios dada la unidad de medida elegida |
| Factor de corrección de estimación | (*) | (*) | (*) |
| Velocidad de desarrollo | (*) | (*) | (*) |
| E.T.A. | (*) | (*) | (*) |
| AxF: Tiempo Estimado vs. Real(**) | (*) | (*) | (*) |
| AxF: Tiempo Estimado Original vs. Estimado Actual(**) | (*) | (*) | (*) |

**Tabla 12: Reportes y Métricas seleccionados**

(*) Métrica extraída de Painless Tracking Excell. Construida originalmente para cumplir este aspecto.

(**) En Painless Tracking Excell, los reportes son llamados indicados son llamados “Avance por Producto” pero para la versión Web serán catalogados como “Avance por Funcionalidad”.

En resumen, los reportes y métricas elegidos son:

- Burn-down chart
- Time and Budget Chart
- CFD
- Factor de correción de estimación
- Velocidad de desarrollo
- E.T.A.
- Avance por Funcionalidad: Estimado vs. Real
- Avance por Funcionalidad: Estimado Original vs. Estimado Actual

La razón para descartar el reporte Burn-up es que sencillamente aporta la misma información que Burn-down. Además Alistair Cockburn muestra en su publicación que reportar cuanto falta puede generar un factor psicológico más positivo en vez de mostrar cuanto se ha avanzado.

Algunos reportes presentes en Painless Tracking Web no fueron elegidos ya que se dejaron para futuras implementaciones e integraciones con herramientas del mundo OLAP.

### 6.2.2. Aplicación de los reportes y métricas seleccionados sobre un ejemplo real

Los reportes y métricas se aplican sobre datos reales (se adjunta planilla con los datos en la sección de Anexos). Se consideran para ello 18 productos con 90 tareas asociadas.

***Burn-Down Chart***

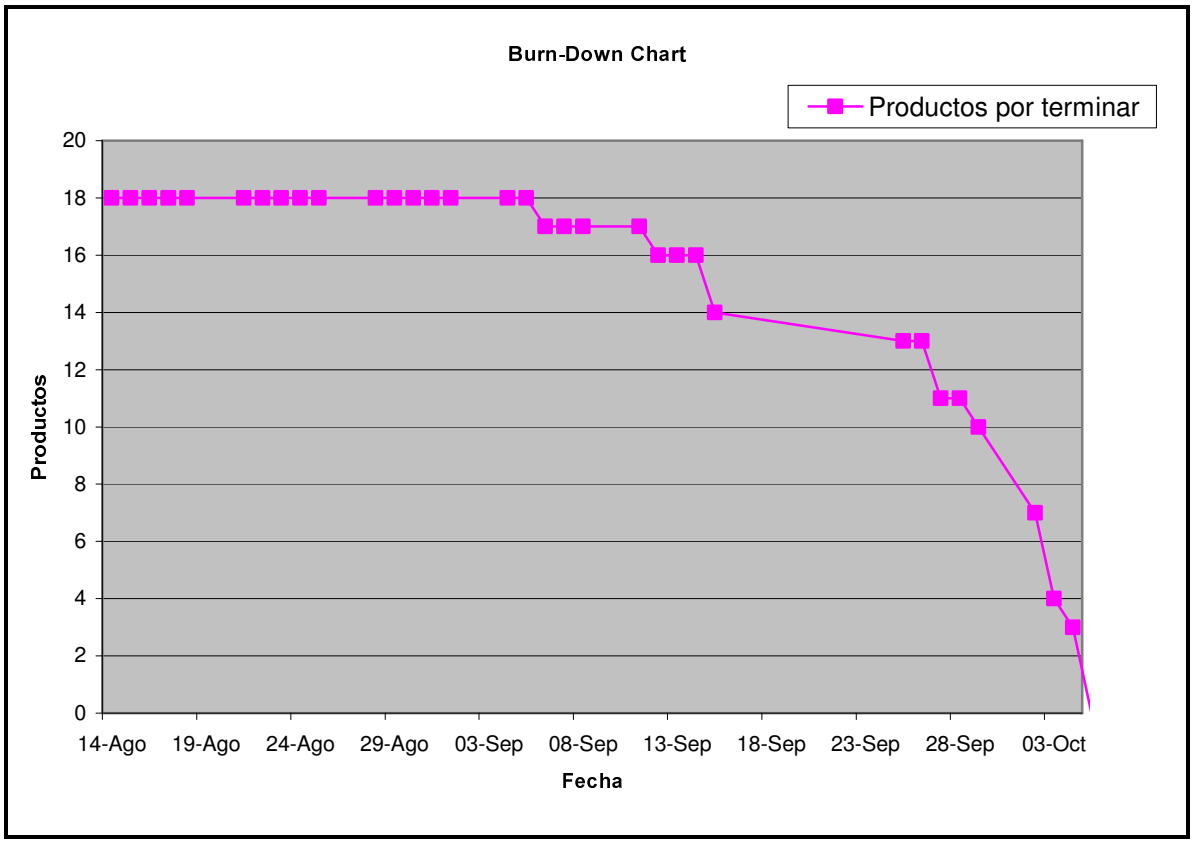


**Ilustración 11: Burn-down con datos reales**

De este gráfico se puede observar lo siguiente:

- Hubo un extenso período de investigación para madurar el problema.
- Hubo en período extenso de días donde no se terminaron productos. Esto fue debido a que se consideraron los días festivos

- Los saltos en la finalización de varios productos a la vez indica que el equipo no concentró sus esfuerzos en terminar productos sino que en avanzar en variadas tareas a la vez.

***CFD***

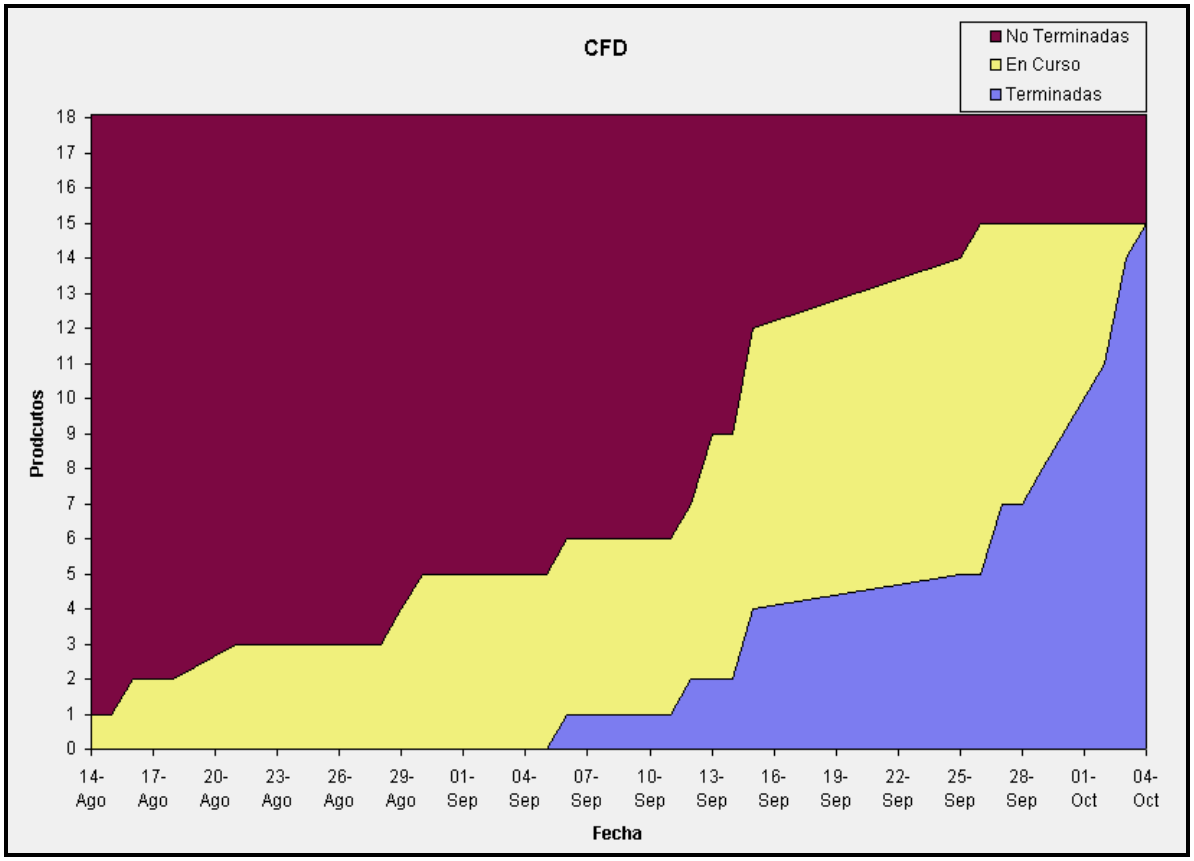


**Ilustración 12: CFD con datos reales**

Este reporte reafirma los puntos expresados en el gráfico anterior:

- El tiempo de entrega de productos terminados en ciertos niveles es muy extenso (ancho del área amarilla).
- En algunos períodos la cantidad de productos en curso es grande (en proporción a la cantidad de productos). Probablemente hubo un cuello de botella en el desarrollo en esos momentos.

- El esfuerzo se enfocó en avanzar en varios productos a la vez. Por consecuencia se terminaron varios productos en ciertas fechas específicas. Basta con observar los saltos en el área de los productos terminados.

***Time & Budget***

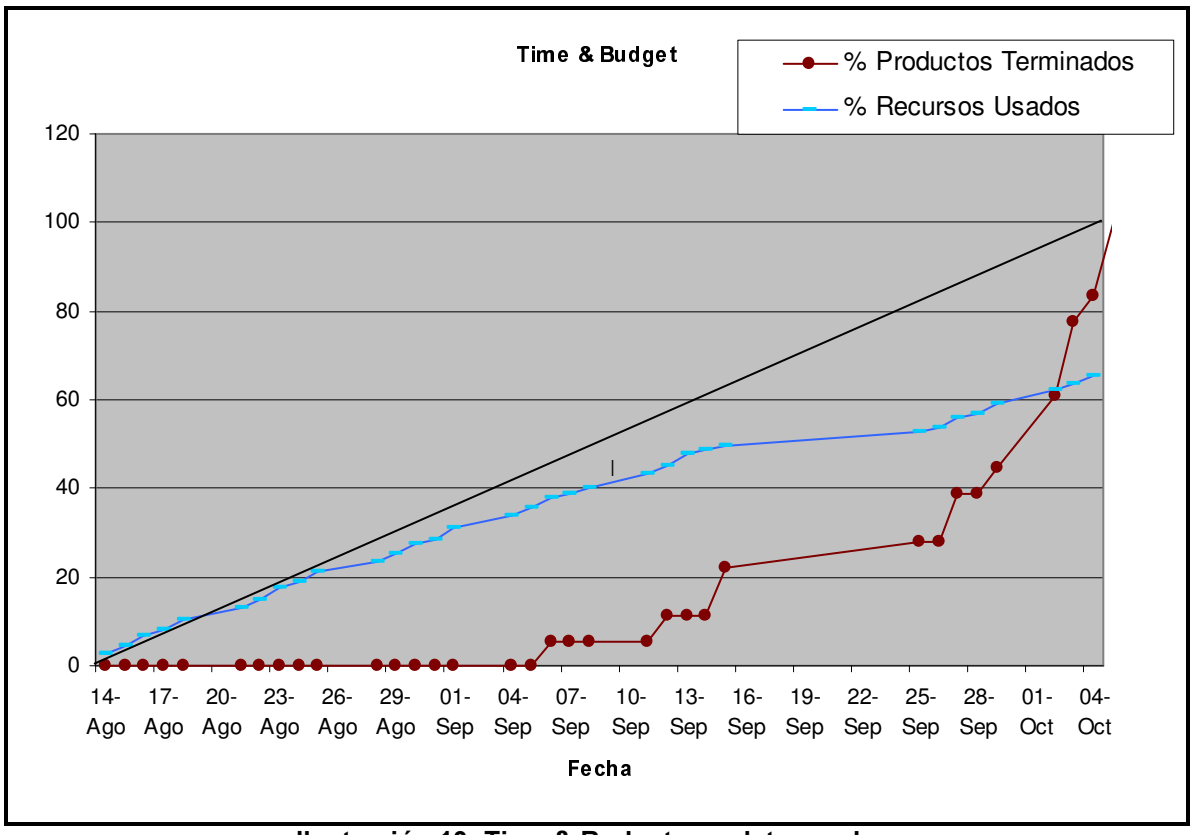


**Ilustración 13: Time & Budget con datos reales**

Ya que los datos utilizados contenidos en la Painless Tracking Excell de ejemplo no contienen información de presupuesto invertido, se tradujo el esfuerzo en recursos humanos (% Recursos humanos usados del total disponible) en los días de desarrollo del proyecto.

Del gráfico podemos observar lo siguiente:

- En general los recursos no era posible distribuirlos de manera uniforme dada la disponibilidad horaria de los recursos.

- A medida que avanzó el desarrollo fue decayendo la cantidad de recursos utilizados, lo que fue arrastrándose a través de los días e impactando al porcentaje de productos terminados.
- En general, se observa una gran inversión inicial en investigación. Como se constata, se iniciaron tareas el 14 de Agosto pasando casi un mes antes de generar un producto terminado, realizándose una inversión en horas-hombre que en términos de costo-beneficio es desfavorable. Podría haberse evitado esta situación asignando y dividiendo la investigación en productos bien determinados para poder hacerles seguimiento y control.

***Métricas***

| | |
|---|---|
| **Factor de corrección de estimación** | 0,81 |
| **Velocidad de desarrollo** | 0,83 |
| **E.T.A.** | 2,86 semanas (para terminar las funcionalidades que quedaron pendientes en este proyecto) |
| **APP: Estimado vs. Real** | 0,92 |
| **APP: Estimado Original vs. Estimado Actual** | 0,68 |

**Tabla 13: Métricas calculadas con datos reales**

De las métricas:

- El factor de corrección muestra que al final del proyecto las estimaciones estaban siendo más certeras.
- La velocidad de desarrollo muestra que quedaron funcionalidades o productos por terminar, sin embargo, una alta cantidad de valor fue entregada al cliente, ya que si se asumiera productos con pesos equivalentes dentro del proyecto, unos 14 de 18 productos fueron entregados.

- Casi tres semanas para la entrega total, lo que indica el E.T.A., no es valor muy alentador. Baja el perfil del las métricas anteriores, ya que considerando que la duración del proyecto fue tres meses, la cantidad de tiempo restante para terminar es grande. Sin embargo, hay una metodología detrás de esta herramienta que asegura que lo entregado era de mayor importancia a lo que quedó pendiente.
- El valor de avance por producto, tiempo estimado versus real, se muestra en promedio para resumir el cuadro global. Indica buenas estimaciones al final del proyecto, dado que lo real gastado en desarrollar los productos es muy cercano a lo que se estaba proyectando.
- El valor de avance por producto, tiempo estimado original versus tiempo estimado actual, también se muestra en promedio para resumir el cuadro global. Muestra en general que las primeras estimaciones no estaban muy cerca de la realidad pero tampoco fueron de tan mala calidad, dado que el valor no es menor a un 0,5.

## 6.3. Implementación módulo de gestión Painless Tracking Web

### 6.3.1. Metodología de Desarrollo

La idea principal de este desarrollo es generar las funcionalidades necesarias para calcular las métricas y reportes seleccionados anteriormente, poder ensamblarlas posteriormente a una interfaz operativa que siga el modelo simple utilizado y generar con ellas el despliegue necesario de acuerdo a la interfaz gráfica que se elija para Painless Tracking Web.

Para ello se realizaron los siguientes pasos:

- Generar, analizar y adoptar un modelo simplificado.
- Desarrollo guiado por tests: casos de prueba implementados sobre el modelo.
- Implementación de módulos de métricas.
- Implementación de módulos de reportes.
- Mostrar resultados a través de una herramienta gráfica compatible.

### 6.3.2. Ambiente de desarrollo

Para este desarrollo se trabajo usando las siguientes tecnologías:

- Desarrollo sobre sistema operativo Windows XP.
- Entorno Portable Python 1.0 [20], que permite el desarrollo independiente de la máquina utilizada (sólo en S.O. Windows), el cual incluye:
  - Lenguaje Python versión 2.5. [21]
  - Nose: extensión basada en el lenguaje anterior para la realización de test de unidad sobre las funcionalidades desarrolladas. [22]
  - SQL Alchemy: Database Toolkit para Python y Object Relational Mapper. [23]
  - Librerías y Scripts basados en Pylons Python Web Framework 0961 [24]. Dentro de ellas se cuentan librerías para el manejo de string, fechas y sobre todo, la creación y manejo de documentos XML.
- XML/SWF Charts Tool versión 4.6, que permite generar gráficos basados en documentos XML usando tecnología Flash. [25]

### 6.3.3. Modelo Entidad-Relación

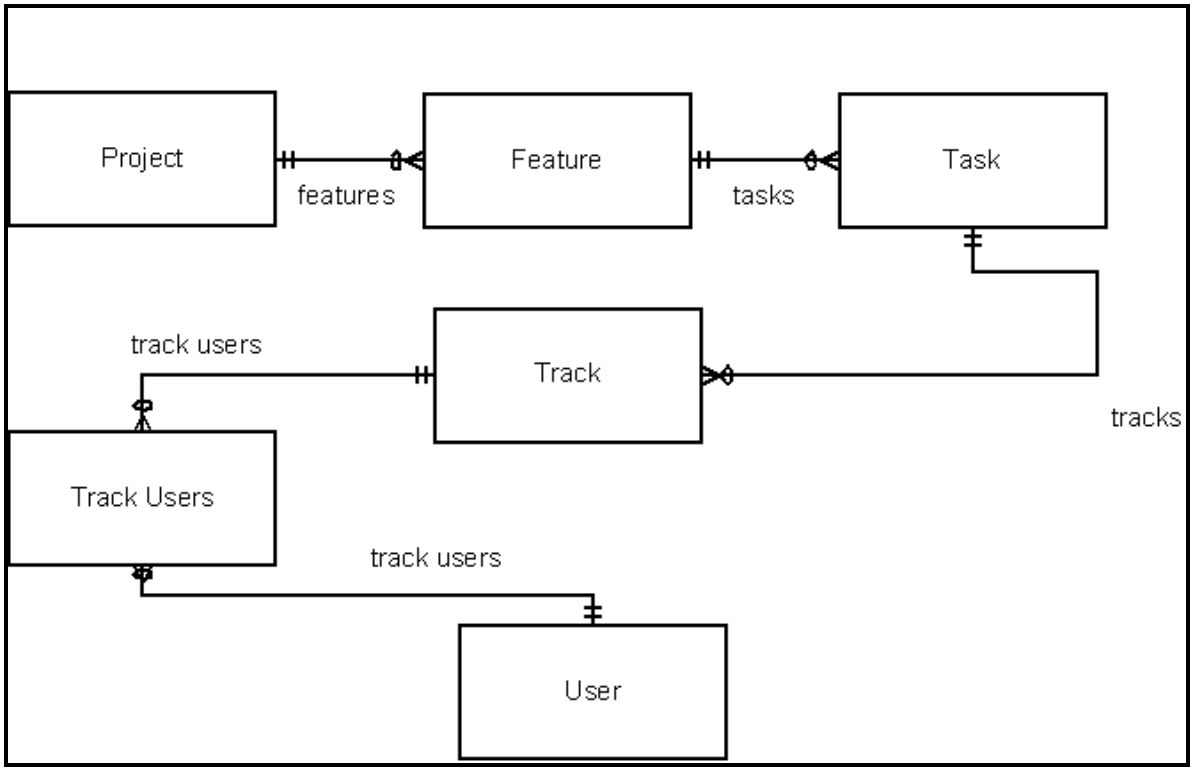


**Ilustración 14: Modelo E/R**

Este es modelo simple para la Painless Tracking Web contiene las entidades y relaciones necesarias para poder realizar control y seguimiento sobre el proyecto. Se describen sus partes a continuación:

Entidades

- Una entidad Project describe un proyecto en el cual se está trabajando.
- Una entidad Feature describe una funcionalidad de un proyecto dado.
- Una entidad Task describe una tarea asociada a una funcionalidad de un proyecto dado.
- Una entidad Track describe un registro de avance sobre una tarea específica.

- Una entidad Track Users describe los tracks asociados a los usuarios que trabajaron en la tareas a cada track.
- Una entidad User describe un usuario de un proyecto dado.

Relaciones

- Una entidad Project de divide en ninguna o varias entidades Features.
- Una entidad Feature se divide en ninguna o varias entidades Tasks.
- Una entidad Task contiene ninguna o varias entidades Tracks.
- Una entidad Track está asociada a ninguna o varias entidades Track Users.
- Una entidad User está asociada a ninguna o varias entidades Track Users.
- Una entidad Feature pertenece a una única entidad Project.
- Una entidad Task pertenece a una única entidad Feature.
- Una entidad Track pertenece a una única entidad Task.
- Una entidad Track Users se asocia a tuplas únicas asociadas a las entidades Track y User.

### 6.3.4. Modelo de datos

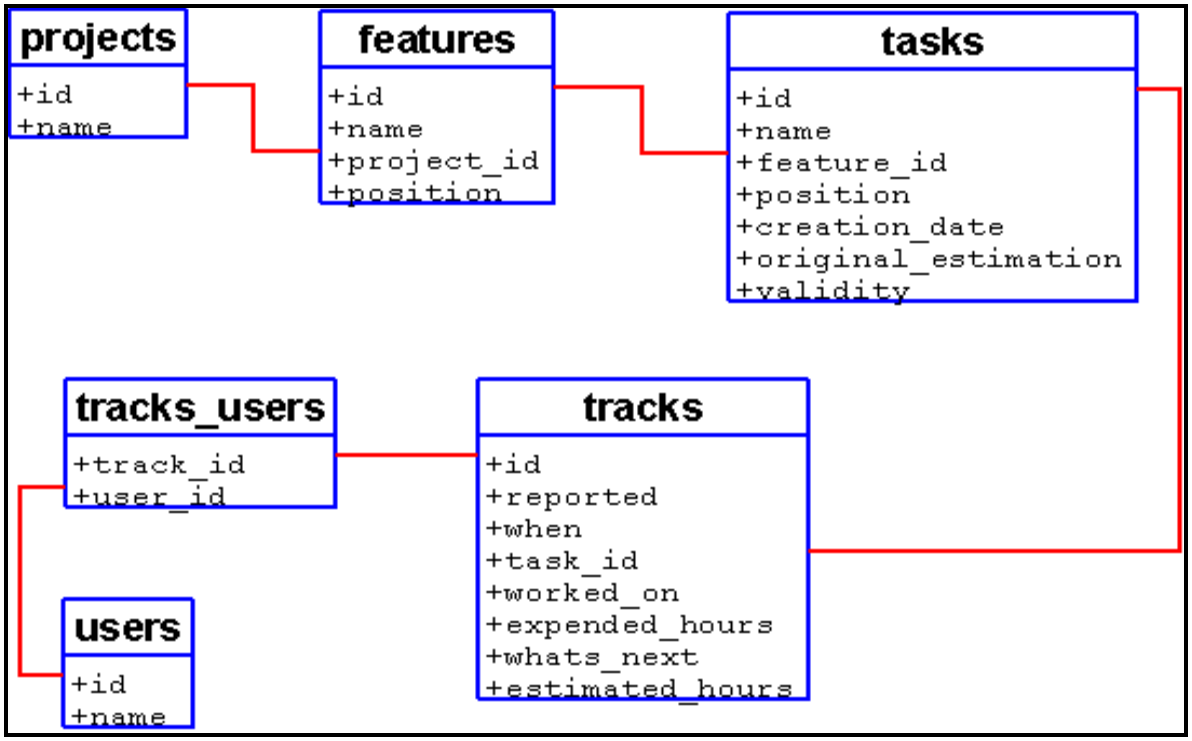


**Ilustración 15: Modelo de datos**

En el modelo anterior se describió las entidades de este modelo y las relaciones entre ellas. Ahora se procede es describir la estructura de los datos para cada entidad:

- Un ***project*** posee:
  - un identificador único (*id*) y,
  - un nombre que lo describe (*name*).
- Una ***feature*** posee:
  - un identificador único (*id*),
  - un nombre que la describe (*name*),
  - un identificador al proyecto que pertenece (*project_id*) y,
  - una posición dentro de las otras features (*position*).

- Una ***task*** posee:
    - un identificador único (*id*),
    - un nombre que la describe (*name*),
    - un identificador a la funcionalidad que pertenece (*feature_id*),
    - una posición dentro de las otras tasks (*position*),
    - una fecha de creación (*creation_date*),
    - una estimación inicial (*original_estimation*) como punto de comparación con futuras estimaciones presentes en los tracks asociados a ésta y,
    - una validez (*validity*) dentro del proyecto (una tarea inválida queda fuera de la parte operativa de la herramienta).
- Un ***track*** posee:
    - un identificador único (*id*),
    - una timestamp de cuando fue ingresado el avance en la herramienta (*reported*),
    - una fecha que indica cuando se avanzó en una tarea específica (*when*),
    - un identificador de tarea a la cual está asociado el track (*task_id*),
    - una descripción de lo realizado en este registro de avance (*worked_on*),
    - el tiempo trabajado sobre la tarea que se está reportando (*expended_hours*),
    - una descripción de que falta por hacer u observaciones si es necesario (*whats_next*) y,
    - las horas estimadas (*estimated_hours*) para finalizar la tarea (un valor cero indica que la tarea ha finalizado).

- Un ***tracks_users*** posee:
    - un identificador al track registrado por el usuario (*track_id*) y,
    - un identificador del usuario responsable de tal track (*user_id*).
- Un ***user*** posee:
    - Un identificador único (*id*) y,
    - Un nombre que los describe (*name*).

### 6.3.5. Desarrollo guiado por tests

Para la implementación de todas las funcionalidades para el cálculo de métricas y reportes, incluyendo las funcionalidades asociadas, se uso la extensión **nose** para el desarrollo en Python.

La metodología seguida es básicamente: primero construir el test y luego implementar la funcionalidad que luego será comprobada con la extensión.

A continuación se describirá un test para una función en particular. El resto de los tests podrán ser encontrados en la sección de Anexos.

El siguiente test fue implementado para la funcionalidad que calcula el factor de correción de estimación del proyecto:

1. Se limpian los datos de sesión para llevar a cabo el test en un ambiente despejado.

```
@with_setup(setup_dbtest, teardown_dbtest)
```

2. Se define el nombre del test.

```
def testFactor_Correccion_Estimacion():
```

3. Se crea un estado inicial del proyecto, con dos features y cada feature con dos tasks.

```
p1 = create_project("P1")
p1_f1 = create_feature_in_project("P1.F1",p1)
p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")
p1_f2 = create_feature_in_project("P1.F2",p1)
p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")
```

4. Se testea el factor al inicio del proyecto. Un valor 1 es sobre el supuesto de que al inicio se está estimando correctamente.

```
# Check State 0
factor = 1.0
assert_equals( factor, getFactor_Correccion_Estimacion(p1) )
```

5. Se agregan tracks para las tareas, finalizando una feature y avanzando otra.

```
# Apply function
user1 = create_user("John Doe")
add_track_to_task('2007-11-07', "Many Things", 5, 0, "Nothing", user1, p1_f1_t1)
add_track_to_task('2007-11-07', "Many Things", 5, 0, "Nothing", user1, p1_f1_t2)
add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
```

6. Se testea la funcionalidad que debe arrojar el valor indicado, calculado sobre las features terminadas.

```
# Check State 1
factor = 3.0/10.0
assert_equals( factor, getFactor_Correccion_Estimacion(p1) )
```

7. Se finaliza la otra feature.

```
# Apply function
user2 = create_user("Joe Black")
add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)
```

8. Se testea el factor, ahora con dos features terminadas.

```
# Check State 2
factor = 17.0/26.0
assert_equals( factor, getFactor_Correccion_Estimacion(p1) )
```

### 6.3.6. Implementación de módulos de métricas

A continuación se describirá cada funcionalidad implementada para calcular cada métrica seleccionada. Los códigos de estas funcionalidades pueden ser encontrados en la sección de Anexos.

Factor de corrección de estimación

- **getFactor_Correccion_Estimacion(project)**: dado un proyecto, se calcula el factor sobre features válidas. Se obtienen el tiempo ocupado en tasks terminadas y el tiempo total de todo el proyecto para calcular este valor.
- **getFactor_Correccion_Estimacion_on_date(project,date)**: realiza el mismo cálculo de la funcionalidad anterior pero además lo hace para una fecha determinada del proyecto (date).

Velocidad de desarrollo

- **getVelocidad_Desarrollo(project)**: calcula la velocidad para el proyecto dado. Esto se realiza sobre el tiempo de features terminadas y válidas y el tiempo total del proyecto.
- **getVelocidad_Desarrollo_on_date(project,date)**: realiza el mismo calculo de la funcionalidad anterior pero lo hace para una fecha determinada.

Estimated Time of Arrival (E.T.A.)

- **getETA(project)**: calcula el tiempo para finalizar el proyecto. Este valor es una proyección de las horas que faltan para terminar según los tracks ponderado por el inverso del factor de corrección de estimación.

- **getETA_on_date(project,date)**: realiza el mismo calculo de la funcionalidad anterior pero lo hace para una fecha determinada.

Avance por Funcionalidad: Tiempo Estimado vs. Tiempo Real

- **getAvance_Funcionalidad_Estimado_VS_Real(feature)**: calcula el avance según el tiempo real que lleva desarrollándose el feature dado y el tiempo estimado (actual) de éste.
- **getAvance_Funcionalidad_Estimado_VS_Real_on_date(feature,date)**: realiza el mismo calculo de la funcionalidad anterior pero lo hace para una fecha determinada.

Avance por Funcionalidad: Tiempo Estimado Original vs. Tiempo Estimado Actual

- **getAvance_Funcionalidad_Estimado_Original_VS_Actual(feature)**: calcula la relación entre la estimación original y la estimación actual para un feature dado.
- **getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(feature,date)**: realiza el mismo calculo de la funcionalidad anterior pero lo hace para una fecha determinada.

Funcionalidades usadas por las anteriores descritas

- **get_feature_validity(feature_id)**: retorna la validez de un feature a partir de la validez de alguna o todas sus tareas asociadas.

Sólo para las funcionalidades que reciben un parámetro date:

- **get_task_status(task,date)**: calcula si una task está no iniciada, iniciada o finalizada a la fecha indicada en date.
- **get_feature_status(feature,date)**: usando la funcionalidad anterior, se calcula lo mismo pero para todo un feature.

- **get_notstarted_features(project,date)**: a partir de la funcionalidad get_feature_status(feature,date) se obtiene el número de features no iniciados a la fecha indicada en date.
- **get_started_features(project,date)**: a partir de la funcionalidad get_feature_status(feature,date) se obtiene el número de features iniciados pero no terminados a la fecha indicada en date.
- **get_finished_features(project,date)**: a partir de la funcionalidad get_feature_status(feature,date) se obtiene el número de features terminados a la fecha indicada en date.

### 6.3.7. Implementación de módulos de reportes

Se describirán inicialmente las funcionalidades adicionales para implementar los reportes, ya que en sí mismas, las funcionalidades encargadas de generar los reportes lo único que hacen es hacer algunos cálculos a partir de las demás, para luego enmascarar y generar un documento XML con los datos para graficar. Los códigos de estas funcionalidades pueden ser encontrados en la sección de Anexos.

Funcionalidades adicionales

- **get_notstarted_features(project,date)**: descrita anteriormente.
- **get_started_features(project,date)**: descrita anteriormente.
- **get_finished_features(project,date)**: descrita anteriormente.
- **get_valid_dates(project,date)**: para fecha indicada en date, retorna el conjunto válido de fechas para el proyecto hasta ese momento. Estas fechas son recogidas tanto de las tasks como de los tracks asociadas a éstas.
- **get_total_features(project,date)**: para fecha indicada en date, retorna el número de features del proyecto hasta ese momento.
- **get_project_users(project,date)**: para fecha indicada en date, retorna el número de usuarios asociados al proyecto hasta ese momento.
- **user_has_tracks_on_date(project,date,user)**: para la fecha indicada en date, esta funcionalidad indica si el usuario ha tenido actividad en el proyecto en ese momento.
- **get_resources_usage(project,dates)**: para un set de fechas indicado en dates, retorna un conjunto que indica para cada fecha el porcentaje de recursos (en este caso recursos humanos) para ese día. El cálculo de uso de recursos es acumulativo a medida que se avanza en el conjunto de fechas.

Funcionalidades encargadas de los reportes

- **draw_burn_down(project,date)**: para la fecha indicada en date se genera el conjunto de fechas válidas para el proyecto, se obtiene el total de features iniciales y se van descontado, en cada fecha, las features terminadas ese día. Se almacenan los datos en un árbol XML junto con la cáscara necesaria para generar un gráfico.
- **draw_CFD(project,date)**: para la fecha indicada en date se genera el conjunto de fechas válidas para el proyecto y para cada una de éstas se calculan las features no iniciadas, iniciadas y terminadas. Se almacenan los datos en un árbol XML junto con la cáscara necesaria para generar un gráfico.
- **draw_time_and_budget(project,date)**: para la fecha indicada en date se genera el conjunto de fechas válidas para el proyecto y para cada una de éstas se calcula el porcentaje de features terminadas del total de features hasta esa fecha y el uso de recursos humanos en cada día. Se almacenan los datos en un árbol XML junto con la cáscara necesaria para generar un gráfico.

Funcionalidades extras

Con el fin de mostrar el cálculo de métricas y aprovechar la implementación del cálculo de métricas para una fecha dada, se generaron las siguientes funcionalidades que generan el árbol XML con los datos necesarios para graficar el historial de cada métrica para un conjunto de fechas válidas al proyecto. Éstas serían:

- **draw_factor_correccion_historial(project,dates)**: para el historial del factor de corrección de estimación.
- **draw_velocidad_desarrollo_historial(project,dates)**: para el historial de la velocidad de desarrollo del proyecto.

- **draw_eta_historial(project,dates)**: para el historial de los E.T.A.'s asociados al proyecto.
- **draw_avance_real_vs_estimado_historial(project,dates)**: para el historial del avance por funcionalidad del tiempo estimado versus el real.
- **draw_avance_original_vs_actual_historial(project,dates)**: para el historial del avance por funcionalidad de tiempo estimado original versus el estimado actual.

### 6.3.8. Resultados gráficos

Usando la herramienta XML/SWF Charts Tool versión 4.6, los outputs de las funcionalidades de reportes anteriores (en formato XML) y la creación de páginas html ad-hoc, se generó el despliegue de los gráficos de reportes anteriormente mencionados. Se usó para ello un proyecto de prueba contenido también en los nosetests:

| | |
|---|---|
| **2007-11-06** | Se crea el proyecto con 2 features y 2 tasks para cada feature. |
| **2007-11-07** | Se ingresa avance y estimación para las tasks de la feature 1. |
| **2007-11-08** | Se ingresa avance y estimación para las tareas de las features 1 y 2.<br>Se ingresa avance y cierre para las tasks de la feature 2. |
| **2007-11-09** | Se crean 2 nuevas features (3 y 4) con 2 tasks cada una.<br>Se ingresa avance y cierre para las tasks de la feature 1.<br>Se ingresa avance y estimación para las tareas de la feature 3. |
| **2007-11-10** | Se ingresa avance y estimación para las tareas de la feature 3.<br>Se ingresa avance y estimación para las tareas de la feature 4. |
| **2007-11-11** | Se ingresa avance y cierre para las tasks de la feature 3.<br>Se ingresa avance y estimación para las tareas de la feature 4. |
| **2007-11-12** | Se ingresa avance y cierre para las tasks de la feature 4.<br>Fin del proyecto. |

## Burn-down Chart

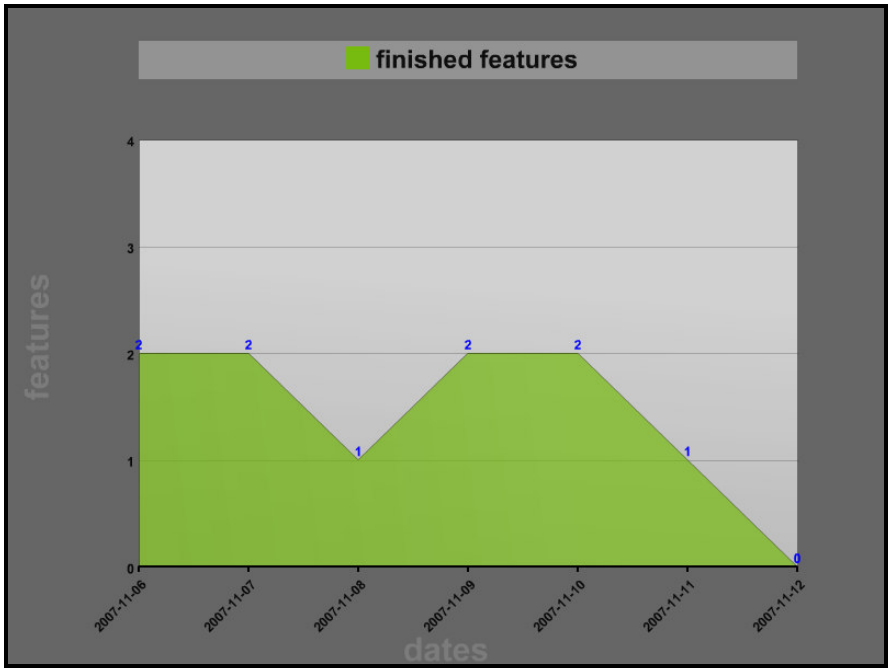


**Ilustración 16: Burn-down graficado**

## CFD

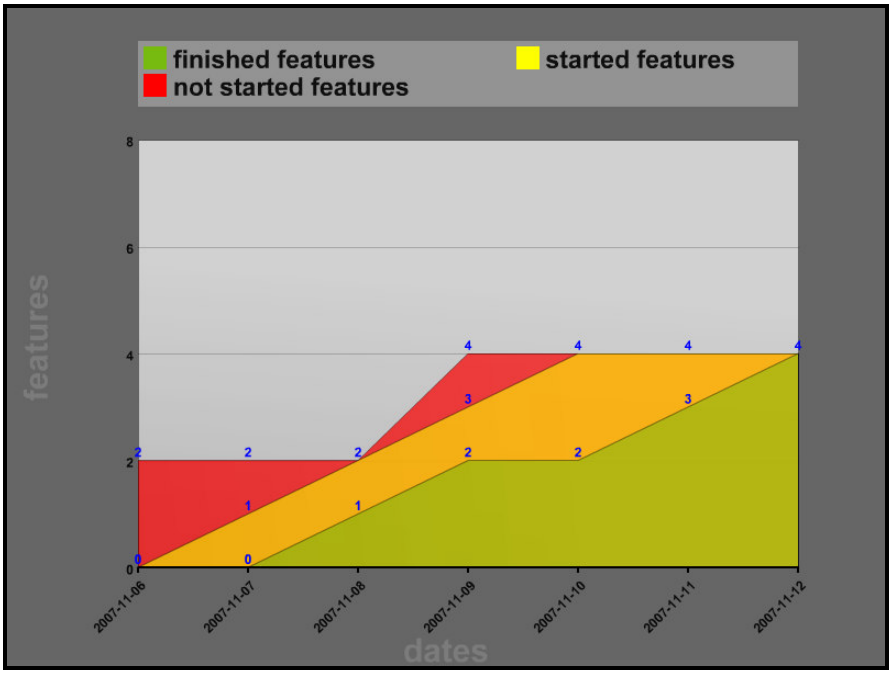


**Ilustración 17: CFD graficado**

## Time & Budget Chart

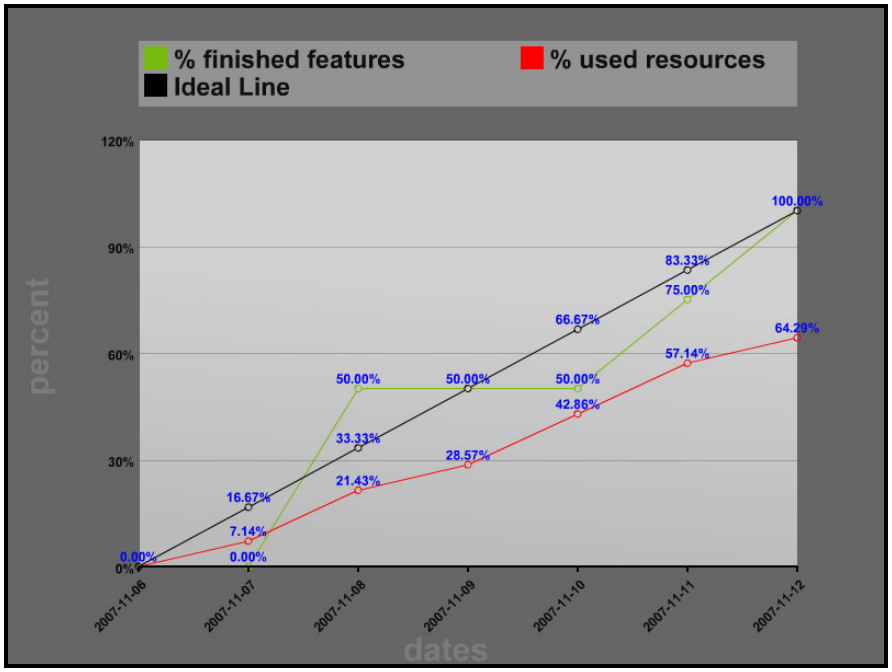


**Ilustración 18: Time&Budget graficado**

## Historial del Factor de Corrección de Estimación

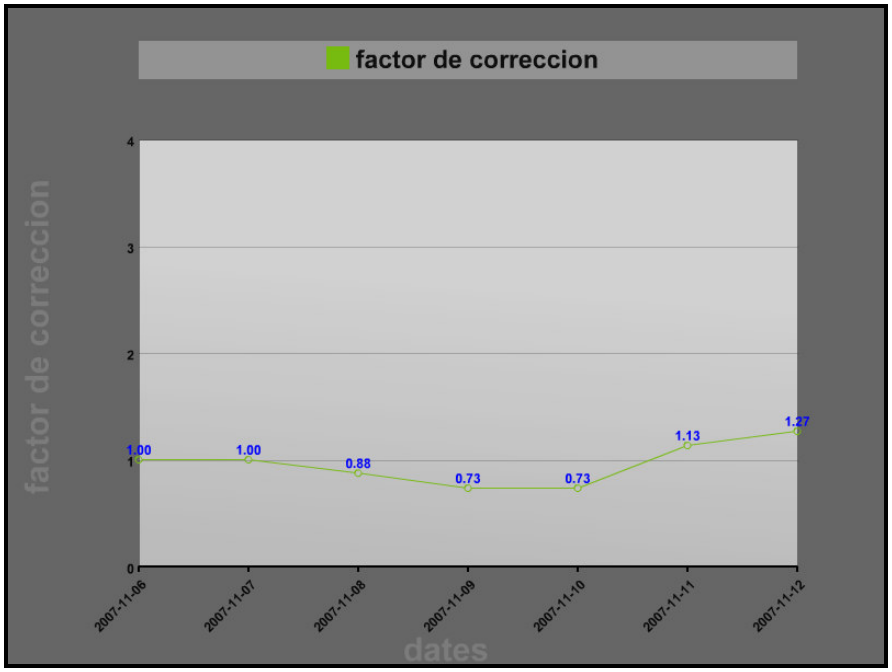


**Ilustración 19: Historial Factor de Correción**

## Historial de la Velocidad de Desarrollo

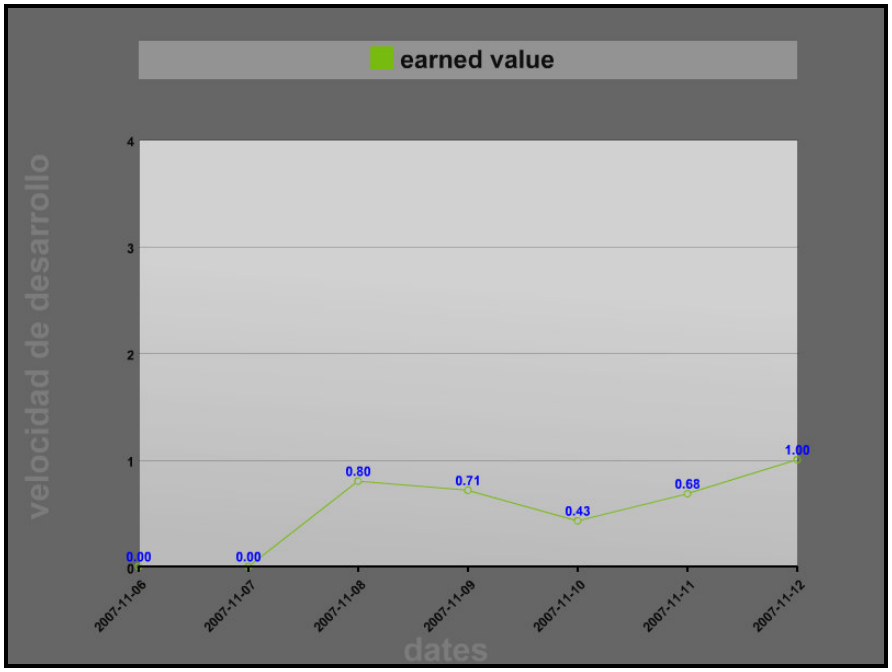


**Ilustración 20: Historial Velocidad de Desarrollo**

## Historial de los E.T.A.'s

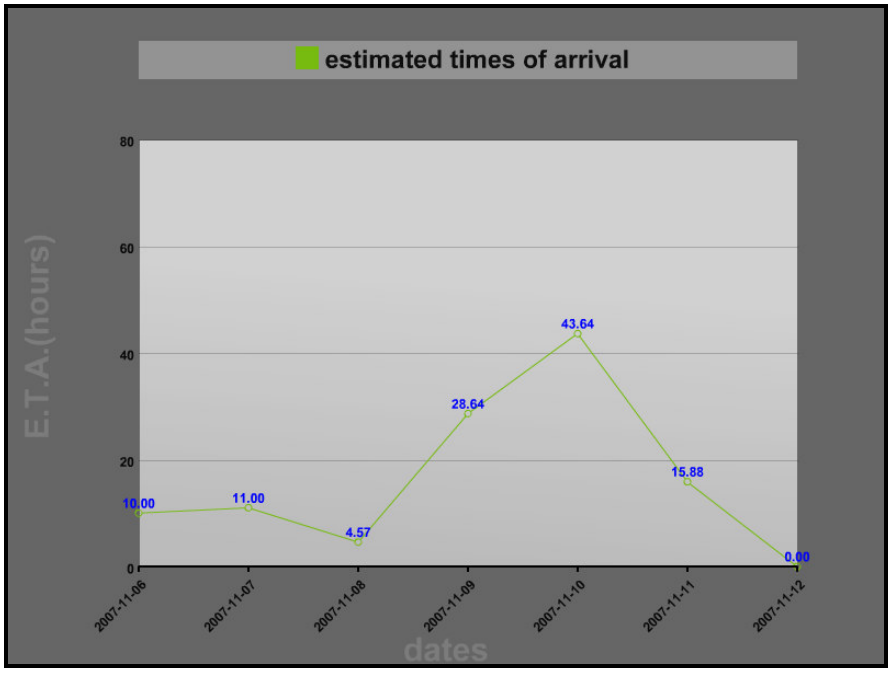


**Ilustración 21: Historial E.T.A.'s**

## Historial de Avance por Funcionalidad: Tiempo Estimado vs. Real


**Ilustración 22: Historial AxF: Tiempo Estimado vs Real**

## Historial de Avance por Funcionalidad: Tiempo Estimado Original vs. Actual

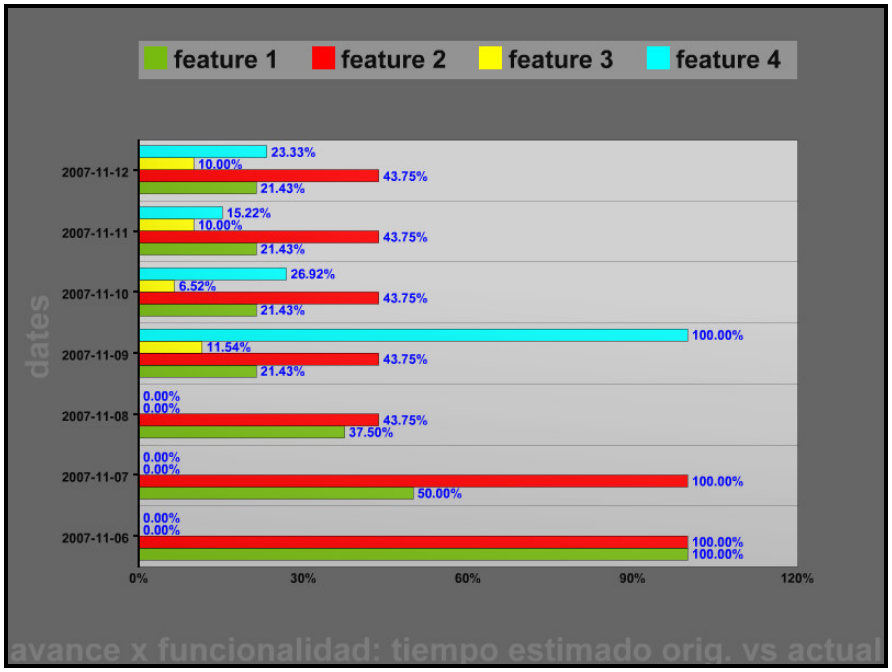


**Ilustración 23: Historial AxF: Tiempo Estimado Original vs Actual**

# 7. Conclusiones

## 7.1. Sobre el marco teórico y el estado del arte investigado

Gracias a la investigación teórica se pudo determinar la forma y las metas que deben perseguir las métricas y reportes a incluir en la interfaz de gestión de la Painless Tracking Web.

Se concluyó que la forma de las métricas debe seguir los siguientes parámetros:

- Soporte para la agregación de nuevos requerimientos o tareas.
- Dimensión adecuada de las unidades de medición. Ni tan pequeñas, para controlar adecuadamente el reporte, ni tan grandes para permitir visibilidad a corto plazo.
- Soporte para el cambio de prioridades.

Los objetivos que se alcanzan al usar las métricas y reportes seleccionados en este trabajo de memoria son los siguientes:

- Gran poder de estimación que permite una mejor definición de los alcances de los proyectos.
- Gran inmunidad frente al cambio. Se genera una base sólida y ágil de respuesta frente a nuevos requerimientos.
- Generación de valor tanto para el cliente como para el equipo de desarrollo.

## 7.2. Sobre la metodología conceptual detrás de este trabajo

Gracias a la adopción de las prácticas y valores presentes en las metodologías ágiles, se generaron métricas y reportes que responden en forma simple y flexible las preguntas que surgen en la gestión y control de proyectos ágiles.

Al usar metodologías ágiles como trasfondo, principalmente XP, se obtuvieron las métricas y reportes necesarios para **generar valor** tanto para el cliente y como para el equipo de desarrollo a cargo del proyecto.

Gracias al desarrollo presente en esta memoria, la interfaz de gestión de la Painless Tracking Web posee métricas y reportes que sólo apuntan a la generación de valor, tanto en el área de recursos como en el área didáctica. A parte de mostrar al cliente cuanto está ganando al invertir en el proyecto, el usuario o el equipo encargado del área de gestión va obteniendo la experiencia necesaria para refinar sus habilidades estimativas.

## 7.3. Sobre la metodología usada para la implementación

Gracias a poner en práctica el desarrollo guiado por tests, se han conseguido funcionalidades de calidad libre de errores, capaces de ser integradas a un modelo más global que incluya las distintas interfaces necesarias para la construcción de la herramienta Painless Tracking Web.

El uso de un modelo simplificado para la implementación de las distintas funcionalidades necesarias para la generación de las métricas y reportes seleccionados para la interfaz de gestión, provee una base flexible para ser extendida con nuevas funcionalidades que pueda requerir la herramienta en el futuro.

## 7.4. Experiencia personal

El autor de esta memoria ha obtenido experiencia sobre:

- Gestión de proyectos ágiles.
- Habilidades estimativas sobre la misma gestión.
- Generación de código de calidad al usar desarrollo guiado por tests.
- Conocimientos técnicos sobre las distintas herramientas y ambientes utilizados para el desarrollo de las métricas y reportes.

# 8. Trabajo Futuro

## 8.1. En el área investigativa sobre gestión de proyectos

Las distintas métricas y reportes generados en este trabajo de memoria nos generan tanto valor en recursos como didáctico. Se puede continuar este trabajo en relación al tema de las capacidades estimativas, investigando y generando nuevos reportes y métricas que apunten a indicar tendencias en los valores y generar mejoras en las habilidades predictivas.

En el futuro, se puede generar mucha experiencia con el uso de estos reportes y métricas que podría permitir la clasificación de proyectos y generar una mejor gestión sobre los mismos, identificando características iniciales y comportamientos comunes.

## 8.2. En el área de desarrollo de la herramienta

Las funcionalidades implementadas o el modelo seguido para hacerlo, pueden ser utilizados para generar nuevos reportes a medida que la misma herramienta, en particular el área de gestión, lo vayan necesitando.

En el futuro se pueden construir nuevos reportes en base a los ya implementados o generar nuevos análisis sobre los mismos.

Se puede ir integrando esta herramienta con otras pertenecientes al mundo OLAP y generar nuevos reportes ad-doc. La herramienta podría manejar gran cantidad de datos surgiendo la necesidad de obtener informes sobre hechos, análisis y estadísticas.

# 9. Bibliografía

[1]. Extreme Programming (XP): http://www.extremeprogramming.org/.

[2]. Beck, Kent, "Extreme Programming Explained", Addison-Wesley Professional, 1st Edition, 1999.

[3]. Beck, K., Fowler, M, “Planning Extreme Programming”, Addison-Wesley Professional, 1st edition, 2000.

[4]. Mike Griffiths, “Most Software Development Metrics Are Misleading And Counterproductive”, http://www.agilejournal.com/content/view/107/.

[5]. Mike Griffiths, LeadingAnswers: Leadership and Agile Project Management Blog, http://leadinganswers.typepad.com/.

[6]. Guy Beaver, The Agile-V Scorecard, http://www.agilejournal.com/content/view/274/.

[7]. Alistair Cockburn, "Crystal Clear", Chapter 3, Addison-Wesley, 2004. http://alistair.cockburn.us/index.php/Earned-value_and_burn_charts.

[8]. SCRUM: http://www.controlchaos.com/.

[9]. Crystal : http://alistair.cockburn.us/index.php/Crystal_methodologies_main_foyer.

[10]. Evolutionary Project Management (Evo): http://www.spipartners.nl/data/Evo99.PDF.

[11]. Feature Driven Development (FDD): http://www.featuredrivendevelopment.com/.

[12]. Adaptive Software Development (ASD): http://www.adaptivesd.com/.

[13]. Lean Development (LD) y Lean Software Development (LSD): http://www.poppendieck.com/.

[14]. Análisis y Evaluación de Impacto de un Modelo Empírico de Enseñanza de Metodologías Ágiles para Alumnos de Ingeniería de Software: el caso del

curso cc62v – taller de metodologías ágiles de desarrollo de software. Tesis para optar al grado de Magíster en Ciencias de la Computación, Villena 2007.

[15]. Time And Budget Chart: Http://Www.Agilekiwi.Com/Agile_Charts.Htm.+

[16]. Villena, Agustín. "Análisis y evaluación de impacto de Un modelo empírico de enseñanza de metodologías ágiles para alumnos de ingeniería de software: El caso del curso cc62v – Taller de metodologías ágiles de desarrollo de software", Tesis para optar al grado de Magister en Ciencias de la Computación, Universidad de Chile, 2007.

[17]. Sanchis, Francisco. "La medición del software", Publicación Programa de Doctorado, Departamento de Organización y Estructura de la Información, Universidad Politécnica de Madrid, 2004.

[18]. Pederson, Jim. "Why Software Metric Programs Fail". Software Process Improvement Network (SPIN), 2001. http://www.uces.csulb.edu/spin/.

[19]. Kent, Beck y Cleal, Dave. "Optional Scope Contracts". www.xprogramming.com/ftp/Optional+scope+contracts.pdf.

[20]. Portable Python: http://www.portablepython.com/

[21]. Python Official Web Site: http://www.python.org/

[22]. Nose: http://nose.python-hosting.com/

[23]. SQL Alchemy: http://www.sqlalchemy.org/

[24]. Pylons Python Web Framework: http://pylonshq.com/

[25]. XML/SWF Charts: http://www.maani.us/xml_charts/index.php

# 10. Anexos

## 10.1. Tests sobre el modelo a implementar

**tests\test_models.py**

```
# -*- coding: UTF-8 -*-
from painlessspike.tests import *
from painlessspike.model import *
from painlessspike.tests.test_model_utils import *

import sqlalchemy
from nose.tools import *

# ########################################################################
# funciones de incicialización y limpieza del modelo
# ########################################################################
def setup_dbtest():
    Session.remove()
    metadata.create_all(Session.bind)

def teardown_dbtest():
    metadata.drop_all(Session.bind)

# ########################################################################
# funciones de testing
# ########################################################################
@with_setup(setup_dbtest, teardown_dbtest)
def testCreateProject():
    # Check State 0
    assert_raises( sqlalchemy.exceptions.InvalidRequestError,
                   Session.query(Project).one )

    # Apply function
    new_project = create_project("Test Project")

    # Check State 1
    assert_equals("Test Project",get_project("Test Project").name)

@with_setup(setup_dbtest, teardown_dbtest)
def testAddFeatureToProject():
    new_project = create_project("Test Project")

    # Check State 0
    assert_equals( new_project.features, [] )

    # Apply function
    new_feature = create_feature_in_project("First Feature",new_project)

    # Check State 1
    assert_equals( "First Feature", get_project("Test Project").features[0].name)
    assert_equals( "Test Project", get_project("Test Project").features[0].project.name )


@with_setup(setup_dbtest, teardown_dbtest)
def testAddTaskToFeature():
    new_project = create_project("Test Project")
    new_feature = create_feature_in_project("First Feature",new_project)

    # Check State 0
    assert_equals( new_project.features[0].tasks, [] )

    # Apply function
    new_task = create_task_in_feature("First Task", 1.5, new_feature,'2007-11-06',"V")

    # Check State 1
    assert_equals( "First Task" , get_project("Test Project").features[0].tasks[0].name)
```

```
    assert_equals( "Test Project" , get_project("Test
Project").features[0].tasks[0].feature.project.name )
    assert_equals( 1.5, get_project("Test Project").features[0].tasks[0].original_estimation
)
    assert_equals( '2007-11-06', get_project("Test
Project").features[0].tasks[0].creation_date )
    assert_equals( "V", get_project("Test Project").features[0].tasks[0].validity )

@with_setup(setup_dbtest, teardown_dbtest)
def testAddTrackToTask():
    new_project = create_project("Test Project")
    new_feature = create_feature_in_project("First Feature",new_project)
    new_task = create_task_in_feature("First Task", 1.5, new_feature,'2007-11-06',"V")
    new_user = create_user("John Doe")

    # Check State 0
    assert_equals( new_task.tracks, [] )
    assert_raises( sqlalchemy.exceptions.InvalidRequestError, Session.query(Track).one )
    assert_equals( new_user.tracks, [] )

    # Apply function
    new_track = add_track_to_task('2007-11-06', "Many Things", 4.1, 0, "Nothing", new_user,
new_task)

    # Check State 1
    tquery = Session.query(Track)
    loaded_track = tquery.filter_by(worked_on="Many Things").one()

    assert_equals("2007-11-06", loaded_track.when)
    assert_equals("Many Things", loaded_track.worked_on)
    assert_equals(4.1, loaded_track.expended_hours)
    assert_equals(0, loaded_track.estimated_hours)
    assert_equals("Nothing", loaded_track.whats_next)

    assert_equals("2007-11-06", get_project("Test
Project").features[0].tasks[0].tracks[0].when)
    assert_equals("Test Project", get_project("Test
Project").features[0].tasks[0].tracks[0].task.feature.project.name)

    assert_equals("2007-11-06", get_user("John Doe").tracks[0].when)
    assert_equals("John Doe", get_project("Test
Project").features[0].tasks[0].tracks[0].users[0].name)

@with_setup(setup_dbtest, teardown_dbtest)
def testCreateUser():
    # Check State 0
    assert_raises( sqlalchemy.exceptions.InvalidRequestError,
                   Session.query(User).one )

    # Apply function
    new_user = create_user("John Doe")

    # Check State 1
    assert_equals("John Doe",get_user("John Doe").name)

@with_setup(setup_dbtest, teardown_dbtest)
def testOrderedFeaturesInProject():
    # Check State 0
    new_project = create_project("Proyecto")
    new_feature =  create_feature_in_project("Funcionalidad",new_project)

    Session.save(new_project)
    Session.commit()

    assert_equals(0, new_feature.position)

    # Apply function
    other_feature = Feature()
    other_feature.name = "Otra"
    new_project.features.insert(0,other_feature)
```

```
    # Check State 1
    assert_equals(1,new_feature.position)
    assert_equals(0,other_feature.position)

@with_setup(setup_dbtest, teardown_dbtest)
def testOrderedTasksInFeature():
    # Check State 0
    new_project = create_project("Proyecto")
    new_feature =  create_feature_in_project("Funcionalidad",new_project)
    new_task = create_task_in_feature("Primera tarea", 1.5, new_feature,'2007-11-06',"V")

    assert_equals(0,new_task.position)

    # Apply function
    other_task = Task()
    other_task.name = "Otra tarea"
    new_feature.tasks.insert(0,other_task)

    # Check State 1
    assert_equals(1,new_task.position)
    assert_equals(0,other_task.position)

@with_setup(setup_dbtest, teardown_dbtest)
def testFeatureValidity():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)

    # Check State 0
    assert_equals( "I",get_feature_validity(p1_f1.id) )
    assert_equals( "I",get_feature_validity(p1_f2.id) )

    # Apply function
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 1
    assert_equals( "V",get_feature_validity(p1_f1.id) )
    assert_equals( "V",get_feature_validity(p1_f2.id) )

    # Apply function
    p1_f1_t1.validity = "I"

    # Check State 2
    assert_equals( "V",get_feature_validity(p1_f1.id) )
    assert_equals( "V",get_feature_validity(p1_f2.id) )

    # Apply function
    p1_f1_t2.validity = "I"

    # Check State 3
    assert_equals( "I",get_feature_validity(p1_f1.id) )
    assert_equals( "V",get_feature_validity(p1_f2.id) )

    # Apply function
    p1_f2_t2.validity = "I"

    # Check State 4
    assert_equals( "I",get_feature_validity(p1_f1.id) )
    assert_equals( "V",get_feature_validity(p1_f2.id) )

    # Apply function
    p1_f2_t1.validity = "I"

    # Check State 5
    assert_equals( "I",get_feature_validity(p1_f1.id) )
    assert_equals( "I",get_feature_validity(p1_f2.id) )
```

## 10.2. Implementación del modelo

**model__init__.py**

```
from pylons import config
from sqlalchemy import Column, MetaData, Table, types
from sqlalchemy.orm import mapper, relation
from sqlalchemy.orm import scoped_session, sessionmaker
from sqlalchemy import ForeignKey
from sqlalchemy.ext.orderinglist import ordering_list
import datetime

# Global session manager.  Session() returns the session object
# appropriate for the current web request.
Session = scoped_session(sessionmaker(autoflush=True, transactional=True,
                                      bind=config['pylons.g'].sa_engine))

# Global metadata. If you have multiple databases with overlapping table
# names, you'll need a metadata for each database.
metadata = MetaData()

# Define a table.
projects_table = Table("projects", metadata,
    Column("id", types.Integer, primary_key=True),
    Column("name", types.String, nullable=False)
    )

features_table = Table("features", metadata,
    Column("id", types.Integer, primary_key=True),
    Column("name", types.String, nullable=False),
    Column("project_id", types.Integer, ForeignKey('projects.id'), nullable=False),
    Column('position',types.Integer)
    )

tasks_table = Table("tasks", metadata,
    Column("id", types.Integer, primary_key=True),
    Column("name", types.String, nullable=False),
    Column("feature_id", types.Integer, ForeignKey('features.id'), nullable=False),
    Column('position',types.Integer),
    Column('creation_date', types.DateTime, nullable=False,
default=datetime.datetime.now()),
    Column('original_estimation', types.Float, nullable=False),
    Column('validity',types.String, nullable=False)
    )

users_table = Table("users", metadata,
    Column("id", types.Integer, primary_key=True),
    Column("name", types.String, nullable=False)
    )

tracks_table = Table("tracks", metadata,
    Column("id", types.Integer, primary_key=True),
    Column("reported", types.DateTime, nullable=False, default=datetime.datetime.now()),
#cuando fue creado el registro de tracking
    Column("when", types.Date, nullable=False),             # fecha de cuando sucedio lo
registrado
    Column("task_id",types.Integer,ForeignKey('tasks.id')), # en que tarea se trabajo (si es
que se trabajo en algo planificado)
    Column("worked_on", types.String, nullable=False),      # descripcion de en que se
trabajo
    Column("expended_hours", types.Float, nullable=False),  # cuantas horas se trabajo
    Column("whats_next", types.String, nullable=False),     # descripcion de lo que falta
por hacer
    Column("estimated_hours", types.Float, nullable=False) # cuantas horas se estiman que
faltan por trabajar
    )

tracks_users_table = Table("tracks_users", metadata,
    Column("track_id", types.Integer, ForeignKey('tracks.id')),
```

```
    Column("user_id", types.Integer, ForeignKey('users.id'))
    )

# Define ORM classes (often called "mapped classes").
class Project(object):    pass
class Feature(object):    pass
class Task(object):    pass
class User(object):    pass
class Track(object):    pass

# Map each class to its corresponding table.
mapper(Project,projects_table,
       properties={
           'features':relation(Feature,
                               collection_class = ordering_list('position'),
                               order_by = [features_table.c.position],
                               backref='project')
           }
       )
mapper(Feature, features_table,
       properties={
           'tasks':relation(Task,
                            collection_class = ordering_list('position'),
                            order_by = [tasks_table.c.position],
                            backref='feature')
           }
       )
mapper(Task, tasks_table,
       properties={
           'tracks':relation(Track, backref='task')
           }
       )

mapper(Track, tracks_table)

mapper(User, users_table, properties={
    'tracks':relation(Track, secondary=tracks_users_table, backref='users')
           }
       )
```

## tests\test_model_utils.py

```
# -*- coding: UTF-8 -*-
from painlessspike.tests import *
from painlessspike.model import *

import sqlalchemy
from nose.tools import *

# ########################################################################
# funciones utilitarias sobre el modelo
# ########################################################################
def create_user(name):
    new_user = User()
    new_user.name = name
    Session.save(new_user)
    Session.commit()
    return new_user

def create_project(name):
    new_project = Project()
    new_project.name = name
    Session.save(new_project)
    Session.commit()
    return new_project

def create_feature_in_project(feature_name,project):
    new_feature =  Feature()
```

```
    new_feature.name = feature_name

    project.features.append(new_feature)

    Session.save(new_feature)
    Session.commit()
    return new_feature

def add_track_to_task(when, worked_on, expended_hours, estimated_hours, whats_next, user,
task):
    new_track = Track()
    new_track.when = when
    new_track.worked_on = worked_on
    new_track.expended_hours = expended_hours
    new_track.whats_next = whats_next
    new_track.estimated_hours = estimated_hours
    new_track.users.append(user)
    new_track.task = task

    Session.save(new_track)
    Session.commit()

    return new_track

def create_task_in_feature(task_name, task_estimation, feature, task_creation_date,
task_validity):
    new_task = Task()
    new_task.name = task_name
    new_task.creation_date = task_creation_date
    new_task.original_estimation = task_estimation
    new_task.validity = task_validity

    feature.tasks.append(new_task)

    ## TODO: revisar el comportamiento de Session por que sin las 2 siguientes lineas
igualmente pasa los tests
    Session.save(new_task)
    Session.commit()

    return new_task

def get_project(project_name):
    pquery = Session.query(Project)
    return pquery.filter_by(name=project_name).one()

def get_user(user_name):
    uquery = Session.query(User)
    return uquery.filter_by(name=user_name).one()

def get_feature_id_from_task_id(task_id):
    tquery = Session.query(Task)
    theTask = tquery.filter_by(id=task_id).one()
    return theTask.feature_id

def get_task_validity_from_task_id(task_id):
    tquery = Session.query(Task)
    theTask = tquery.filter_by(id=task_id).one()
    return theTask.validity

def get_feature_validity(feature_id):
    tquery = Session.query(Task)
    tasks = tquery.from_statement("SELECT * FROM tasks WHERE feature_id =:fid").params(fid =
feature_id).all()
    validity = "I"
    if tasks:
        for task in tasks:
            if task.validity == "V":
                validity = "V"

    return validity
```

## 10.3. Tests sobre las funcionalidades a implementar

**tests\functional\management\ test_metrics.py**

```
# -*- coding: UTF-8 -*-
from painlessspike.tests import *
from painlessspike.model import *
from painlessspike.tests.test_model_utils import *
from painlessspike.lib.painless_management import *

import sqlalchemy
from nose.tools import *
from lxml import etree

# ###############################################################################
# funciones de incicialización y limpieza del modelo
# ###############################################################################
def setup_dbtest():
    Session.remove()
    metadata.create_all(Session.bind)

def teardown_dbtest():
    metadata.drop_all(Session.bind)

# ###############################################################################
# funciones de testing
# ###############################################################################
def create_fixture_plan():
    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    return p1

@with_setup(setup_dbtest, teardown_dbtest)
def testProjectWithoutTracks():

    # Check State 0
    project = create_fixture_plan()

    # Apply function
    pass

    # Check State 1
    assert_equals( 10, get_original_estimated_hours(project) )
    #assert_equals( 0, get_estimated_hours(project))
    #assert_equals( 0, get_elapsed_hours(project))

@with_setup(setup_dbtest, teardown_dbtest)
def testFactor_Correccion_Estimacion():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    factor = 1.0
    assert_equals( factor, getFactor_Correccion_Estimacion(p1) )
```

```
    # Apply function
    user1 = create_user("John Doe")
    add_track_to_task('2007-11-07', "Many Things", 5, 0, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 5, 0, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 1
    factor = 3.0/10.0
    assert_equals( factor, getFactor_Correccion_Estimacion(p1) )

    # Apply function
    user2 = create_user("Joe Black")
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 2
    factor = 17.0/26.0
    assert_equals( factor, getFactor_Correccion_Estimacion(p1) )

@with_setup(setup_dbtest, teardown_dbtest)
def testVelocidad_Desarrollo():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    velocidad = 0.0
    assert_equals( velocidad, getVelocidad_Desarrollo(p1) )

    # Apply function
    user1 = create_user("John Doe")
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 1
    velocidad = 0.0
    assert_equals( velocidad, getVelocidad_Desarrollo(p1) )

    # Apply function
    user2 = create_user("Joe Black")
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 2
    velocidad = 16.0/18.0
    assert_equals( velocidad, getVelocidad_Desarrollo(p1) )

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 3
    velocidad = 1.0
    assert_equals( velocidad, getVelocidad_Desarrollo(p1) )

@with_setup(setup_dbtest, teardown_dbtest)
def testETA():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")
```

```
    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    ETA = 10.0
    assert_equals( ETA, getETA(p1) )

    # Apply function
    user1 = create_user("John Doe")
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 1
    ETA = 12.0
    assert_equals( ETA, getETA(p1) )

    # Apply function
    user2 = create_user("Joe Black")
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 2
    factor = 14.0/16.0
    ETA = 4.0/factor
    assert_equals( ETA, getETA(p1) )

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 3
    ETA = 0.0
    assert_equals( ETA, getETA(p1) )

@with_setup(setup_dbtest, teardown_dbtest)
def testAvance_Funcionalidad_Estimado_VS_Real():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    AvanceEvsR_F1 = 0.0*100
    AvanceEvsR_F2 = 0.0*100
    assert_equals(AvanceEvsR_F1, getAvance_Funcionalidad_Estimado_VS_Real(p1_f1) )
    assert_equals(AvanceEvsR_F2, getAvance_Funcionalidad_Estimado_VS_Real(p1_f2) )

    # Apply function
    user1 = create_user("John Doe")
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 1
    AvanceEvsR_F1 = (2.0 /(2.0 + 4.0))*100
    AvanceEvsR_F2 = (6.0 /(6.0 + 8.0))*100
    assert_equals(AvanceEvsR_F1, getAvance_Funcionalidad_Estimado_VS_Real(p1_f1) )
    assert_equals(AvanceEvsR_F2, getAvance_Funcionalidad_Estimado_VS_Real(p1_f2) )

    # Apply function
    user2 = create_user("Joe Black")
```

```
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 2
    AvanceEvsR_F1 = (4.0 / (4.0 + 4.0))*100
    AvanceEvsR_F2 = (16.0 / (16.0 + 0.0))*100
    assert_equals(AvanceEvsR_F1, getAvance_Funcionalidad_Estimado_VS_Real(p1_f1) )
    assert_equals(AvanceEvsR_F2, getAvance_Funcionalidad_Estimado_VS_Real(p1_f2) )

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 3
    AvanceEvsR_F1 = (14.0 / (14.0 + 0.0))*100
    AvanceEvsR_F2 = (16.0 / (16.0 + 0.0))*100
    assert_equals(AvanceEvsR_F1, getAvance_Funcionalidad_Estimado_VS_Real(p1_f1) )
    assert_equals(AvanceEvsR_F2, getAvance_Funcionalidad_Estimado_VS_Real(p1_f2) )

@with_setup(setup_dbtest, teardown_dbtest)
def testAvance_Funcionalidad_Estimado_Original_VS_Actual():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2 = create_feature_in_project("P1.F2",p1)
    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    AvanceEOvsA_F1 = (3.0 / 3.0)*100
    AvanceEOvsA_F2 = (7.0 / 7.0)*100
    assert_equals( AvanceEOvsA_F1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f1) )
    assert_equals( AvanceEOvsA_F2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f2) )

    # Apply function
    user1 = create_user("John Doe")
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 1
    AvanceEOvsA_F1 = (3.0 / (2.0 + 4.0))*100
    AvanceEOvsA_F2 = (7.0 / (6.0 + 8.0))*100
    assert_equals( AvanceEOvsA_F1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f1) )
    assert_equals( AvanceEOvsA_F2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f2) )

    # Apply function
    user2 = create_user("Joe Black")
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 2
    AvanceEOvsA_F1 = (3.0 / (4.0 + 4.0))*100
    AvanceEOvsA_F2 = (7.0 / (16.0 + 0.0))*100
    assert_equals( AvanceEOvsA_F1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f1) )
    assert_equals( AvanceEOvsA_F2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f2) )
```

```
    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 3
    AvanceEOvsA_F1 = (3.0 / (14.0 + 0.0))*100
    AvanceEOvsA_F2 = (7.0 / (16.0 + 0.0))*100
    assert_equals( AvanceEOvsA_F1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f1) )
    assert_equals( AvanceEOvsA_F2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual(p1_f2) )

@with_setup(setup_dbtest, teardown_dbtest)
def testTaskStatus():

    #status values:
        # 0 = not started
        # 1 = started
        # 2 = finished

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    # Apply function
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    assert_equals(0, get_task_status(p1_f1_t1,'2007-11-06'))
    assert_equals(0, get_task_status(p1_f1_t2,'2007-11-06'))
    assert_equals(0, get_task_status(p1_f2_t1,'2007-11-06'))
    assert_equals(0, get_task_status(p1_f2_t2,'2007-11-06'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)

    # Check State 1
    assert_equals(1, get_task_status(p1_f1_t1,'2007-11-07'))
    assert_equals(1, get_task_status(p1_f1_t2,'2007-11-07'))
    assert_equals(0, get_task_status(p1_f2_t1,'2007-11-07'))
    assert_equals(0, get_task_status(p1_f2_t2,'2007-11-07'))

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 2
    assert_equals(1, get_task_status(p1_f1_t1,'2007-11-08'))
    assert_equals(1, get_task_status(p1_f1_t2,'2007-11-08'))
    assert_equals(1, get_task_status(p1_f2_t1,'2007-11-08'))
    assert_equals(1, get_task_status(p1_f2_t2,'2007-11-08'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 3
    assert_equals(1, get_task_status(p1_f1_t1,'2007-11-08'))
    assert_equals(1, get_task_status(p1_f1_t2,'2007-11-08'))
    assert_equals(2, get_task_status(p1_f2_t1,'2007-11-08'))
    assert_equals(2, get_task_status(p1_f2_t2,'2007-11-08'))
```

```
    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 4
    assert_equals(1, get_task_status(p1_f1_t1,'2007-11-07'))
    assert_equals(1, get_task_status(p1_f1_t2,'2007-11-07'))
    assert_equals(2, get_task_status(p1_f1_t1,'2007-11-08'))
    assert_equals(2, get_task_status(p1_f1_t2,'2007-11-08'))
    assert_equals(2, get_task_status(p1_f2_t1,'2007-11-08'))
    assert_equals(2, get_task_status(p1_f2_t2,'2007-11-08'))

@with_setup(setup_dbtest, teardown_dbtest)
def testFeatureStatus():

    #status values:
        # 0 = not started
        # 1 = started
        # 2 = finished

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    # Apply function
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    assert_equals(0, get_feature_status(p1_f1,'2007-11-06'))
    assert_equals(0, get_feature_status(p1_f2,'2007-11-06'))


    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)

    # Check State 1
    assert_equals(1, get_feature_status(p1_f1,'2007-11-07'))
    assert_equals(0, get_feature_status(p1_f2,'2007-11-07'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)

    # Check State 2
    assert_equals(1, get_feature_status(p1_f1,'2007-11-07'))
    assert_equals(0, get_feature_status(p1_f2,'2007-11-07'))

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 3
    assert_equals(1, get_feature_status(p1_f1,'2007-11-08'))
    assert_equals(1, get_feature_status(p1_f2,'2007-11-08'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 4
    assert_equals(1, get_feature_status(p1_f1,'2007-11-08'))
    assert_equals(2, get_feature_status(p1_f2,'2007-11-08'))

    # Apply function
```

```
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 5
    assert_equals(1, get_feature_status(p1_f1,'2007-11-07'))
    assert_equals(2, get_feature_status(p1_f1,'2007-11-08'))
    assert_equals(2, get_feature_status(p1_f2,'2007-11-08'))

@with_setup(setup_dbtest, teardown_dbtest)
def testProjectSnapShot():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    # Apply function
    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    # Check State 0
    valid_dates = ['2007-11-06']
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-06'))
    assert_equals(2, get_notstarted_features(p1,'2007-11-06'))
    assert_equals(0, get_started_features(p1,'2007-11-06'))
    assert_equals(0, get_finished_features(p1,'2007-11-06'))


    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)

    # Check State 1
    valid_dates = ['2007-11-06','2007-11-07']
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-07'))
    assert_equals(1, get_notstarted_features(p1,'2007-11-07'))
    assert_equals(1, get_started_features(p1,'2007-11-07'))
    assert_equals(0, get_finished_features(p1,'2007-11-07'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)

    # Check State 2
    valid_dates = ['2007-11-06','2007-11-07']
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-07'))
    assert_equals(1, get_notstarted_features(p1,'2007-11-07'))
    assert_equals(1, get_started_features(p1,'2007-11-07'))
    assert_equals(0, get_finished_features(p1,'2007-11-07'))

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)

    # Check State 3
    valid_dates = ['2007-11-06','2007-11-07','2007-11-08']
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-08'))
    assert_equals(0, get_notstarted_features(p1,'2007-11-08'))
    assert_equals(2, get_started_features(p1,'2007-11-08'))
    assert_equals(0, get_finished_features(p1,'2007-11-08'))

    # Apply function
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    # Check State 4
    valid_dates = ['2007-11-06','2007-11-07','2007-11-08']
```

```
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-08'))
    assert_equals(0, get_notstarted_features(p1,'2007-11-08'))
    assert_equals(1, get_started_features(p1,'2007-11-08'))
    assert_equals(1, get_finished_features(p1,'2007-11-08'))

    # Apply function
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)

    # Check State 5
    valid_dates = ['2007-11-06','2007-11-07','2007-11-08']
    assert_equals(valid_dates, get_valid_dates(p1,'2007-11-08'))
    assert_equals(0, get_notstarted_features(p1,'2007-11-08'))
    assert_equals(0, get_started_features(p1,'2007-11-08'))
    assert_equals(2, get_finished_features(p1,'2007-11-08'))

@with_setup(setup_dbtest, teardown_dbtest)
def testDrawBurnDown():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user2, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user2, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "area"
```

```
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    # <chart_data> -> <row> : Y-Axis Data - Serie 1
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "features left"
    features_left1 = etree.SubElement(row2, "number")
    features_left1.text = "2"
    features_left2 = etree.SubElement(row2, "number")
    features_left2.text = "2"
    features_left3 = etree.SubElement(row2, "number")
    features_left3.text = "1"
    features_left4 = etree.SubElement(row2, "number")
    features_left4.text = "2"
    features_left5 = etree.SubElement(row2, "number")
    features_left5.text = "2"
    features_left6 = etree.SubElement(row2, "number")
    features_left6.text = "1"
    features_left7 = etree.SubElement(row2, "number")
    features_left7.text = "0"
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "features"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_burn_down(p1,'2007-11-12'))

@with_setup(setup_dbtest, teardown_dbtest)
def testCFD():

    p1 = create_project("P1")
```

```
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06','V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06','V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06','V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06','V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09','V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09','V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09','V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09','V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user2, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user2, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "area"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
```

```
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Finished Features
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "finished features"
    features_finished1 = etree.SubElement(row2, "number")
    features_finished1.text = "0"
    features_finished2 = etree.SubElement(row2, "number")
    features_finished2.text = "0"
    features_finished3 = etree.SubElement(row2, "number")
    features_finished3.text = "1"
    features_finished4 = etree.SubElement(row2, "number")
    features_finished4.text = "2"
    features_finished5 = etree.SubElement(row2, "number")
    features_finished5.text = "2"
    features_finished6 = etree.SubElement(row2, "number")
    features_finished6.text = "3"
    features_finished7 = etree.SubElement(row2, "number")
    features_finished7.text = "4"

    # <chart_data> -> <row> : Y-Axis Data - Serie 2: Started Features
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "started features"
    features_started1 = etree.SubElement(row3, "number")
    features_started1.text = "0"
    features_started2 = etree.SubElement(row3, "number")
    features_started2.text = "1"
    features_started3 = etree.SubElement(row3, "number")
    features_started3.text = "2"
    features_started4 = etree.SubElement(row3, "number")
    features_started4.text = "3"
    features_started5 = etree.SubElement(row3, "number")
    features_started5.text = "4"
    features_started6 = etree.SubElement(row3, "number")
    features_started6.text = "4"
    features_started7 = etree.SubElement(row3, "number")
    features_started7.text = "4"

    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Not Started Features
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "not started features"
    features_notstarted1 = etree.SubElement(row4, "number")
    features_notstarted1.text = "2"
    features_notstarted2 = etree.SubElement(row4, "number")
    features_notstarted2.text = "2"
    features_notstarted3 = etree.SubElement(row4, "number")
    features_notstarted3.text = "2"
    features_notstarted4 = etree.SubElement(row4, "number")
    features_notstarted4.text = "4"
    features_notstarted5 = etree.SubElement(row4, "number")
    features_notstarted5.text = "4"
    features_notstarted6 = etree.SubElement(row4, "number")
    features_notstarted6.text = "4"
    features_notstarted7 = etree.SubElement(row4, "number")
    features_notstarted7.text = "4"

    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff')
    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "features"
```

```
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ffff00"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ff0000"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_CFD(p1,'2007-11-12'))

@with_setup(setup_dbtest, teardown_dbtest)
def testTotalUsers():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    users = [1 , 2]
    assert_equals(users,get_project_users(p1,'2007-11-06'))

@with_setup(setup_dbtest, teardown_dbtest)
def testTimeAndBudget():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
```

```
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: % Finished Features
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "% finished features"
    features_finished_percent1 = etree.SubElement(row2, "number")
    features_finished_percent1.text = "0.0"
    features_finished_percent2 = etree.SubElement(row2, "number")
    features_finished_percent2.text = "0.0"
    features_finished_percent3 = etree.SubElement(row2, "number")
    features_finished_percent3.text = "50.0"
    features_finished_percent4 = etree.SubElement(row2, "number")
    features_finished_percent4.text = "50.0"
    features_finished_percent5 = etree.SubElement(row2, "number")
    features_finished_percent5.text = "50.0"
    features_finished_percent6 = etree.SubElement(row2, "number")
    features_finished_percent6.text = "75.0"
    features_finished_percent7 = etree.SubElement(row2, "number")
    features_finished_percent7.text = "100.0"

    # <chart_data> -> <row> : Y-Axis Data - Serie 2: % Used resources
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "% used resources"
    percent_tmp = (0.0 / 14.0)*100
    used_resources_percent1 = etree.SubElement(row3, "number")
    used_resources_percent1.text = str(percent_tmp)
    percent_tmp = (1.0 / 14.0)*100
    used_resources_percent2 = etree.SubElement(row3, "number")
    used_resources_percent2.text = str(percent_tmp)
    percent_tmp = (3.0 / 14.0)*100
    used_resources_percent3 = etree.SubElement(row3, "number")
    used_resources_percent3.text = str(percent_tmp)
```

```
    percent_tmp = (4.0 / 14.0)*100
    used_resources_percent4 = etree.SubElement(row3, "number")
    used_resources_percent4.text = str(percent_tmp)
    percent_tmp = (6.0 / 14.0)*100
    used_resources_percent5 = etree.SubElement(row3, "number")
    used_resources_percent5.text = str(percent_tmp)
    percent_tmp = (8.0 / 14.0)*100
    used_resources_percent6 = etree.SubElement(row3, "number")
    used_resources_percent6.text = str(percent_tmp)
    percent_tmp = (9.0 / 14.0)*100
    used_resources_percent7 = etree.SubElement(row3, "number")
    used_resources_percent7.text = str(percent_tmp)

    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Ideal Line
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "Ideal Line"
    percent_tmp = 0.0*(100.0/6.0)
    ideal_line1 = etree.SubElement(row4, "number")
    ideal_line1.text = str(percent_tmp)
    percent_tmp = 1.0*(100.0/6.0)
    ideal_line2 = etree.SubElement(row4, "number")
    ideal_line2.text = str(percent_tmp)
    percent_tmp = 2.0*(100.0/6.0)
    ideal_line3 = etree.SubElement(row4, "number")
    ideal_line3.text = str(percent_tmp)
    percent_tmp = 3.0*(100.0/6.0)
    ideal_line4 = etree.SubElement(row4, "number")
    ideal_line4.text = str(percent_tmp)
    percent_tmp = 4.0*(100.0/6.0)
    ideal_line5 = etree.SubElement(row4, "number")
    ideal_line5.text = str(percent_tmp)
    percent_tmp = 5.0*(100.0/6.0)
    ideal_line6 = etree.SubElement(row4, "number")
    ideal_line6.text = str(percent_tmp)
    percent_tmp = 6.0*(100.0/6.0)
    ideal_line7 = etree.SubElement(row4, "number")
    ideal_line7.text = str(percent_tmp)

    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', suffix='%', decimals='2')
    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "percent"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "000000"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_time_and_budget(p1,'2007-11-12'))

@with_setup(setup_dbtest, teardown_dbtest)
def testMetricsOnDate():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
```

```
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    date = '2007-11-06'
    factor_correccion = 1.0
    velocidad = 0.0
    eta = (3.0 + 7.0)
    estimado_vs_realF1 = ( 0.0 / (0.0 + 3.0) )*100.0
    estimado_vs_realF2 = ( 0.0 / (0.0 + 7.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (0.0 + 3.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (0.0 + 7.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(0.0, getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f3,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))

    date = '2007-11-07'
    factor_correccion = 1.0
    velocidad = 0.0/2.0
    eta = (4.0 + 7.0)
```

```
    estimado_vs_realF1 = ( 2.0 / (2.0 + 4.0) )*100.0
    estimado_vs_realF2 = ( 0.0 / (0.0 + 7.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (2.0 + 4.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (0.0 + 7.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))

    date = '2007-11-08'
    factor_correccion = 14.0/16.0
    velocidad = 16.0/20.0
    eta = (0.0 + 4.0)/factor_correccion
    estimado_vs_realF1 = ( 4.0 / (4.0 + 4.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (4.0 + 4.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))

    date = '2007-11-09'
    factor_correccion = 22.0/30.0
    velocidad = 30.0/42.0
    eta = (0.0 + 0.0 + 14.0 + 7.0)/factor_correccion
    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 12.0 / (12.0 + 14.0) )*100.0
    estimado_vs_realF4 = ( 0.0 / (0.0 + 7.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (12.0 + 14.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (0.0 + 7.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(estimado_vs_realF3,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f3,date))
    assert_equals(estimado_vs_realF4,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f4,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))
    assert_equals(original_vs_actualF3,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f3,date))
    assert_equals(original_vs_actualF4,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f4,date))

    date = '2007-11-10'
    factor_correccion = 22.0/30.0
    velocidad = 30.0/70.0
```

```
    eta = (0.0 + 0.0 + 18.0 + 14.0)/factor_correccion
    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 28.0 / (28.0 + 18.0) )*100.0
    estimado_vs_realF4 = ( 12.0 / (12.0 + 14.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (28.0 + 18.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (12.0 + 14.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(estimado_vs_realF3,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f3,date))
    assert_equals(estimado_vs_realF4,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f4,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))
    assert_equals(original_vs_actualF3,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f3,date))
    assert_equals(original_vs_actualF4,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f4,date))

    date = '2007-11-11'
    factor_correccion = 68.0/60.0
    velocidad = 60.0/88.0
    eta = (0.0 + 0.0 + 0.0 + 18.0)/factor_correccion
    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 30.0 / (30.0 + 0.0) )*100.0
    estimado_vs_realF4 = ( 28.0 / (28.0 + 18.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (30.0 + 0.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (28.0 + 18.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(estimado_vs_realF3,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f3,date))
    assert_equals(estimado_vs_realF4,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f4,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))
    assert_equals(original_vs_actualF3,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f3,date))
    assert_equals(original_vs_actualF4,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f4,date))

    date = '2007-11-12'
    factor_correccion = 114.0/90.0
    velocidad = 90.0/90.0
    eta = (0.0 + 0.0 + 0.0 + 0.0)/factor_correccion
    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 30.0 / (30.0 + 0.0) )*100.0
    estimado_vs_realF4 = ( 30.0 / (30.0 + 0.0) )*100.0
    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
```

```
    original_vs_actualF3 = ( 3.0 / (30.0 + 0.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (30.0 + 0.0) )*100.0
    assert_equals(factor_correccion, getFactor_Correccion_Estimacion_on_date(p1,date))
    assert_equals(velocidad, getVelocidad_Desarrollo_on_date(p1,date))
    assert_equals(eta, getETA_on_date(p1,date))
    assert_equals(estimado_vs_realF1,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f1,date))
    assert_equals(estimado_vs_realF2,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f2,date))
    assert_equals(estimado_vs_realF3,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f3,date))
    assert_equals(estimado_vs_realF4,
getAvance_Funcionalidad_Estimado_VS_Real_on_date(p1_f4,date))
    assert_equals(original_vs_actualF1,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f1,date))
    assert_equals(original_vs_actualF2,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f2,date))
    assert_equals(original_vs_actualF3,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f3,date))
    assert_equals(original_vs_actualF4,
getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(p1_f4,date))

@with_setup(setup_dbtest, teardown_dbtest)
def testDrawFactorCorreccionHistorial():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
```

```
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    dates = ['2007-11-06', '2007-11-07', '2007-11-08', '2007-11-09', '2007-11-10', '2007-11-
11', '2007-11-12']

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Factor de correccion
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "factor de correccion"
    factor_correccion = 1.0
    factor_correccion1 = etree.SubElement(row2, "number")
    factor_correccion1.text = str(factor_correccion)
    factor_correccion = 1.0
    factor_correccion2 = etree.SubElement(row2, "number")
    factor_correccion2.text = str(factor_correccion)
    factor_correccion = 14.0/16.0
    factor_correccion3 = etree.SubElement(row2, "number")
    factor_correccion3.text = str(factor_correccion)
    factor_correccion = 22.0/30.0
    factor_correccion4 = etree.SubElement(row2, "number")
    factor_correccion4.text = str(factor_correccion)
    factor_correccion = 22.0/30.0
    factor_correccion5 = etree.SubElement(row2, "number")
    factor_correccion5.text = str(factor_correccion)
    factor_correccion = 68.0/60.0
    factor_correccion6 = etree.SubElement(row2, "number")
    factor_correccion6.text = str(factor_correccion)
    factor_correccion = 114.0/90.0
    factor_correccion7 = etree.SubElement(row2, "number")
    factor_correccion7.text = str(factor_correccion)
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "factor de correccion"
```

```
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_factor_correccion_historial(p1,dates))

@with_setup(setup_dbtest, teardown_dbtest)
def testDrawVelocidadDesarrolloHistorial():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
```

```
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    dates = ['2007-11-06', '2007-11-07', '2007-11-08', '2007-11-09', '2007-11-10', '2007-11-
11', '2007-11-12']

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Velocidad de Desarrollo
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "earned value"
    velocidad_desarrollo = 0.0
    velocidad_desarrollo1 = etree.SubElement(row2, "number")
    velocidad_desarrollo1.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 0.0/2.0
    velocidad_desarrollo2 = etree.SubElement(row2, "number")
    velocidad_desarrollo2.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 16.0/20.0
    velocidad_desarrollo3 = etree.SubElement(row2, "number")
    velocidad_desarrollo3.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 30.0/42.0
    velocidad_desarrollo4 = etree.SubElement(row2, "number")
    velocidad_desarrollo4.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 30.0/70.0
    velocidad_desarrollo5 = etree.SubElement(row2, "number")
    velocidad_desarrollo5.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 60.0/88.0
    velocidad_desarrollo6 = etree.SubElement(row2, "number")
    velocidad_desarrollo6.text = str(velocidad_desarrollo)
    velocidad_desarrollo = 90.0/90.0
    velocidad_desarrollo7 = etree.SubElement(row2, "number")
    velocidad_desarrollo7.text = str(velocidad_desarrollo)
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "velocidad de desarrollo"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End
```

```
    root_string = etree.tostring(root)

    assert_equals(root_string, draw_velocidad_desarrollo_historial(p1,dates))

@with_setup(setup_dbtest, teardown_dbtest)
def testDrawETAHistorial():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
```

```
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    dates = ['2007-11-06', '2007-11-07', '2007-11-08', '2007-11-09', '2007-11-10', '2007-11-
11', '2007-11-12']

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: E.T.A.
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "estimated times of arrival"
    factor_correccion = 1.0
    eta = (3.0 + 7.0)
    eta1 = etree.SubElement(row2, "number")
    eta1.text = str(eta)
    factor_correccion = 1.0
    eta = (4.0 + 7.0)
    eta2 = etree.SubElement(row2, "number")
    eta2.text = str(eta)
    factor_correccion = 14.0/16.0
    eta = (0.0 + 4.0)/factor_correccion
    eta3 = etree.SubElement(row2, "number")
    eta3.text = str(eta)
    factor_correccion = 22.0/30.0
    eta = (0.0 + 0.0 + 14.0 + 7.0)/factor_correccion
    eta4 = etree.SubElement(row2, "number")
    eta4.text = str(eta)
    factor_correccion = 22.0/30.0
    eta = (0.0 + 0.0 + 18.0 + 14.0)/factor_correccion
    eta5 = etree.SubElement(row2, "number")
    eta5.text = str(eta)
    factor_correccion = 68.0/60.0
    eta = (0.0 + 0.0 + 0.0 + 18.0)/factor_correccion
    eta6 = etree.SubElement(row2, "number")
    eta6.text = str(eta)
    factor_correccion = 114.0/90.0
    eta = (0.0 + 0.0 + 0.0 + 0.0)/factor_correccion
    eta7 = etree.SubElement(row2, "number")
    eta7.text = str(eta)
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "E.T.A.(hours)"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_eta_historial(p1,dates))
```

```
@with_setup(setup_dbtest, teardown_dbtest)
def testDrawAvanceEstimadoVSRealHistorial():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "bar"
    axis_category = etree.SubElement(root, "axis_category", size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
```

```
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    dates = ['2007-11-06', '2007-11-07', '2007-11-08', '2007-11-09', '2007-11-10', '2007-11-
11', '2007-11-12']

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Avance Estimado VS Real Producto 1
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "feature 1"
    # <chart_data> -> <row> : Y-Axis Data - Serie 2: Avance Estimado VS Real Producto 2
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "feature 2"
    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Avance Estimado VS Real Producto 3
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "feature 3"
    # <chart_data> -> <row> : Y-Axis Data - Serie 4: Avance Estimado VS Real Producto 4
    row5 = etree.SubElement(chart_data, "row")
    string_row5 = etree.SubElement(row5, "string")
    string_row5.text = "feature 4"


    estimado_vs_realF1 = ( 0.0 / (0.0 + 3.0) )*100.0
    estimado_vs_realF2 = ( 0.0 / (0.0 + 7.0) )*100.0
    estimado_vs_realF3 = ( 0.0 )*100.0
    estimado_vs_realF4 = ( 0.0 )*100.0
    estimado_realP1_1 = etree.SubElement(row2, "number")
    estimado_realP1_1.text = str(estimado_vs_realF1)
    estimado_realP2_1 = etree.SubElement(row3, "number")
    estimado_realP2_1.text = str(estimado_vs_realF2)
    estimado_realP3_1 = etree.SubElement(row4, "number")
    estimado_realP3_1.text = str(estimado_vs_realF3)
    estimado_realP4_1 = etree.SubElement(row5, "number")
    estimado_realP4_1.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 2.0 / (2.0 + 4.0) )*100.0
    estimado_vs_realF2 = ( 0.0 / (0.0 + 7.0) )*100.0
    estimado_vs_realF3 = ( 0.0 )*100.0
    estimado_vs_realF4 = ( 0.0 )*100.0
    estimado_realP1_2 = etree.SubElement(row2, "number")
    estimado_realP1_2.text = str(estimado_vs_realF1)
    estimado_realP2_2 = etree.SubElement(row3, "number")
    estimado_realP2_2.text = str(estimado_vs_realF2)
    estimado_realP3_2 = etree.SubElement(row4, "number")
    estimado_realP3_2.text = str(estimado_vs_realF3)
    estimado_realP4_2 = etree.SubElement(row5, "number")
    estimado_realP4_2.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 4.0 / (4.0 + 4.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 0.0 )*100.0
    estimado_vs_realF4 = ( 0.0 )*100.0
    estimado_realP1_3 = etree.SubElement(row2, "number")
    estimado_realP1_3.text = str(estimado_vs_realF1)
    estimado_realP2_3 = etree.SubElement(row3, "number")
    estimado_realP2_3.text = str(estimado_vs_realF2)
    estimado_realP3_3 = etree.SubElement(row4, "number")
    estimado_realP3_3.text = str(estimado_vs_realF3)
    estimado_realP4_3 = etree.SubElement(row5, "number")
    estimado_realP4_3.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
```

```
    estimado_vs_realF3 = ( 12.0 / (12.0 + 14.0) )*100.0
    estimado_vs_realF4 = ( 0.0 / (0.0 + 7.0) )*100.0
    estimado_realP1_4 = etree.SubElement(row2, "number")
    estimado_realP1_4.text = str(estimado_vs_realF1)
    estimado_realP2_4 = etree.SubElement(row3, "number")
    estimado_realP2_4.text = str(estimado_vs_realF2)
    estimado_realP3_4 = etree.SubElement(row4, "number")
    estimado_realP3_4.text = str(estimado_vs_realF3)
    estimado_realP4_4 = etree.SubElement(row5, "number")
    estimado_realP4_4.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 28.0 / (28.0 + 18.0) )*100.0
    estimado_vs_realF4 = ( 12.0 / (12.0 + 14.0) )*100.0
    estimado_realP1_5 = etree.SubElement(row2, "number")
    estimado_realP1_5.text = str(estimado_vs_realF1)
    estimado_realP2_5 = etree.SubElement(row3, "number")
    estimado_realP2_5.text = str(estimado_vs_realF2)
    estimado_realP3_5 = etree.SubElement(row4, "number")
    estimado_realP3_5.text = str(estimado_vs_realF3)
    estimado_realP4_5 = etree.SubElement(row5, "number")
    estimado_realP4_5.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 30.0 / (30.0 + 0.0) )*100.0
    estimado_vs_realF4 = ( 28.0 / (28.0 + 18.0) )*100.0
    estimado_realP1_6 = etree.SubElement(row2, "number")
    estimado_realP1_6.text = str(estimado_vs_realF1)
    estimado_realP2_6 = etree.SubElement(row3, "number")
    estimado_realP2_6.text = str(estimado_vs_realF2)
    estimado_realP3_6 = etree.SubElement(row4, "number")
    estimado_realP3_6.text = str(estimado_vs_realF3)
    estimado_realP4_6 = etree.SubElement(row5, "number")
    estimado_realP4_6.text = str(estimado_vs_realF4)

    estimado_vs_realF1 = ( 14.0 / (14.0 + 0.0) )*100.0
    estimado_vs_realF2 = ( 10.0 / (10.0 + 0.0) )*100.0
    estimado_vs_realF3 = ( 30.0 / (30.0 + 0.0) )*100.0
    estimado_vs_realF4 = ( 30.0 / (30.0 + 0.0) )*100.0
    estimado_realP1_7 = etree.SubElement(row2, "number")
    estimado_realP1_7.text = str(estimado_vs_realF1)
    estimado_realP2_7 = etree.SubElement(row3, "number")
    estimado_realP2_7.text = str(estimado_vs_realF2)
    estimado_realP3_7 = etree.SubElement(row4, "number")
    estimado_realP3_7.text = str(estimado_vs_realF3)
    estimado_realP4_7 = etree.SubElement(row5, "number")
    estimado_realP4_7.text = str(estimado_vs_realF4)
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='outside', size='12',
color='0000ff', decimals='2', suffix="%")

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "dates"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "avance x funcionalidad:  tiempo estimado vs tiempo real"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ffff00"
```

```
    color4 = etree.SubElement(series_color, "color")
    color4.text = "00ffff"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_avance_real_vs_estimado_historial(p1,dates))

@with_setup(setup_dbtest, teardown_dbtest)
def testDrawAvanceOriginalVSActualHistorial():

    p1 = create_project("P1")
    p1_f1 = create_feature_in_project("P1.F1",p1)
    p1_f2 = create_feature_in_project("P1.F2",p1)
    user1 = create_user("John Doe")
    user2 = create_user("Joe Black")

    p1_f1_t1 = create_task_in_feature("P1.F1.T1",1,p1_f1,'2007-11-06',"V")
    p1_f1_t2 = create_task_in_feature("P1.F1.T2",2,p1_f1,'2007-11-06',"V")

    p1_f2_t1 = create_task_in_feature("P1.F2.T1",3,p1_f2,'2007-11-06',"V")
    p1_f2_t2 = create_task_in_feature("P1.F2.T2",4,p1_f2,'2007-11-06',"V")

    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-07', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 3, 4, "Nothing", user1, p1_f2_t2)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t1)
    add_track_to_task('2007-11-08', "Many Things", 1, 2, "Nothing", user1, p1_f1_t2)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t1)
    add_track_to_task('2007-11-08', "Many Things", 5, 0, "Nothing", user2, p1_f2_t2)

    p1_f3 = create_feature_in_project("P1.F3",p1)
    p1_f4 = create_feature_in_project("P1.F4",p1)

    p1_f3_t1 = create_task_in_feature("P1.F3.T1",1,p1_f3,'2007-11-09',"V")
    p1_f3_t2 = create_task_in_feature("P1.F3.T2",2,p1_f3,'2007-11-09',"V")

    p1_f4_t1 = create_task_in_feature("P1.F4.T1",3,p1_f4,'2007-11-09',"V")
    p1_f4_t2 = create_task_in_feature("P1.F4.T2",4,p1_f4,'2007-11-09',"V")

    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t1)
    add_track_to_task('2007-11-09', "Many Things", 5, 0, "Nothing", user2, p1_f1_t2)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t1)
    add_track_to_task('2007-11-09', "Many Things", 6, 7, "Nothing", user2, p1_f3_t2)

    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-10', "Many Things", 8, 9, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-10', "Many Things", 6, 7, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t1)
    add_track_to_task('2007-11-11', "Many Things", 1, 0, "Nothing", user1, p1_f3_t2)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-11', "Many Things", 8, 9, "Nothing", user2, p1_f4_t2)

    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t1)
    add_track_to_task('2007-11-12', "Many Things", 1, 0, "Nothing", user2, p1_f4_t2)

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "bar"
    axis_category = etree.SubElement(root, "axis_category", size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
```

```
    null_root1 = etree.SubElement(row1, "null")
    date1 = etree.SubElement(row1, "string")
    date1.text = '2007-11-06'
    date2 = etree.SubElement(row1, "string")
    date2.text = '2007-11-07'
    date3 = etree.SubElement(row1, "string")
    date3.text = '2007-11-08'
    date4 = etree.SubElement(row1, "string")
    date4.text = '2007-11-09'
    date5 = etree.SubElement(row1, "string")
    date5.text = '2007-11-10'
    date6 = etree.SubElement(row1, "string")
    date6.text = '2007-11-11'
    date7 = etree.SubElement(row1, "string")
    date7.text = '2007-11-12'

    dates = ['2007-11-06', '2007-11-07', '2007-11-08', '2007-11-09', '2007-11-10', '2007-11-
11', '2007-11-12']

    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Avance Estimado VS Real Producto 1
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "feature 1"
    # <chart_data> -> <row> : Y-Axis Data - Serie 2: Avance Estimado VS Real Producto 2
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "feature 2"
    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Avance Estimado VS Real Producto 3
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "feature 3"
    # <chart_data> -> <row> : Y-Axis Data - Serie 4: Avance Estimado VS Real Producto 4
    row5 = etree.SubElement(chart_data, "row")
    string_row5 = etree.SubElement(row5, "string")
    string_row5.text = "feature 4"

    original_vs_actualF1 = ( 3.0 / (0.0 + 3.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (0.0 + 7.0) )*100.0
    original_vs_actualF3 = ( 0.0 )*100.0
    original_vs_actualF4 = ( 0.0 )*100.0
    original_actualP1_1 = etree.SubElement(row2, "number")
    original_actualP1_1.text = str(original_vs_actualF1)
    original_actualP2_1 = etree.SubElement(row3, "number")
    original_actualP2_1.text = str(original_vs_actualF2)
    original_actualP3_1 = etree.SubElement(row4, "number")
    original_actualP3_1.text = str(original_vs_actualF3)
    original_actualP4_1 = etree.SubElement(row5, "number")
    original_actualP4_1.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (2.0 + 4.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (0.0 + 7.0) )*100.0
    original_vs_actualF3 = ( 0.0 )*100.0
    original_vs_actualF4 = ( 0.0 )*100.0
    original_actualP1_2 = etree.SubElement(row2, "number")
    original_actualP1_2.text = str(original_vs_actualF1)
    original_actualP2_2 = etree.SubElement(row3, "number")
    original_actualP2_2.text = str(original_vs_actualF2)
    original_actualP3_2 = etree.SubElement(row4, "number")
    original_actualP3_2.text = str(original_vs_actualF3)
    original_actualP4_2 = etree.SubElement(row5, "number")
    original_actualP4_2.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (4.0 + 4.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 0.0 )*100.0
    original_vs_actualF4 = ( 0.0 )*100.0
    original_actualP1_3 = etree.SubElement(row2, "number")
    original_actualP1_3.text = str(original_vs_actualF1)
    original_actualP2_3 = etree.SubElement(row3, "number")
    original_actualP2_3.text = str(original_vs_actualF2)
    original_actualP3_3 = etree.SubElement(row4, "number")
```

```
    original_actualP3_3.text = str(original_vs_actualF3)
    original_actualP4_3 = etree.SubElement(row5, "number")
    original_actualP4_3.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (12.0 + 14.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (0.0 + 7.0) )*100.0
    original_actualP1_4 = etree.SubElement(row2, "number")
    original_actualP1_4.text = str(original_vs_actualF1)
    original_actualP2_4 = etree.SubElement(row3, "number")
    original_actualP2_4.text = str(original_vs_actualF2)
    original_actualP3_4 = etree.SubElement(row4, "number")
    original_actualP3_4.text = str(original_vs_actualF3)
    original_actualP4_4 = etree.SubElement(row5, "number")
    original_actualP4_4.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (28.0 + 18.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (12.0 + 14.0) )*100.0
    original_actualP1_5 = etree.SubElement(row2, "number")
    original_actualP1_5.text = str(original_vs_actualF1)
    original_actualP2_5 = etree.SubElement(row3, "number")
    original_actualP2_5.text = str(original_vs_actualF2)
    original_actualP3_5 = etree.SubElement(row4, "number")
    original_actualP3_5.text = str(original_vs_actualF3)
    original_actualP4_5 = etree.SubElement(row5, "number")
    original_actualP4_5.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (30.0 + 0.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (28.0 + 18.0) )*100.0
    original_actualP1_6 = etree.SubElement(row2, "number")
    original_actualP1_6.text = str(original_vs_actualF1)
    original_actualP2_6 = etree.SubElement(row3, "number")
    original_actualP2_6.text = str(original_vs_actualF2)
    original_actualP3_6 = etree.SubElement(row4, "number")
    original_actualP3_6.text = str(original_vs_actualF3)
    original_actualP4_6 = etree.SubElement(row5, "number")
    original_actualP4_6.text = str(original_vs_actualF4)

    original_vs_actualF1 = ( 3.0 / (14.0 + 0.0) )*100.0
    original_vs_actualF2 = ( 7.0 / (16.0 + 0.0) )*100.0
    original_vs_actualF3 = ( 3.0 / (30.0 + 0.0) )*100.0
    original_vs_actualF4 = ( 7.0 / (30.0 + 0.0) )*100.0
    original_actualP1_7 = etree.SubElement(row2, "number")
    original_actualP1_7.text = str(original_vs_actualF1)
    original_actualP2_7 = etree.SubElement(row3, "number")
    original_actualP2_7.text = str(original_vs_actualF2)
    original_actualP3_7 = etree.SubElement(row4, "number")
    original_actualP3_7.text = str(original_vs_actualF3)
    original_actualP4_7 = etree.SubElement(row5, "number")
    original_actualP4_7.text = str(original_vs_actualF4)
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='outside', size='12',
color='0000ff', decimals='2', suffix="%")

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "dates"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "avance x funcionalidad: tiempo estimado orig. vs actual"

    series_color = etree.SubElement(root, "series_color")
```

```
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ffff00"
    color4 = etree.SubElement(series_color, "color")
    color4.text = "00ffff"
    # Tree End

    root_string = etree.tostring(root)

    assert_equals(root_string, draw_avance_original_vs_actual_historial(p1,dates))
```

## 10.4. Implementación de las funcionalidades

lib\ painless_management.py

```
from painlessspike.model import *
from painlessspike.tests.test_model_utils import *
from time import strptime
import datetime
import sqlalchemy
from lxml import etree

def get_original_estimated_hours(project):
    total = 0
    for feature in project.features:
        for task in feature.tasks:
            total = total + task.original_estimation
    return total

def getFactor_Correccion_Estimacion(project):

    factor = 0.0
    finished_estimated = 0.0
    finished_ocuppied = 0.0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                track_query1 = Session.query(Track)
                tracks = track_query1.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
                if tracks:
                    if tracks[0].estimated_hours == 0:
                        try:
                            if tracks[1]:
                                all_minus_last_track = 0
                                for track in tracks:
                                    if track.id != tracks[0].id:
                                        all_minus_last_track = all_minus_last_track +
track.expended_hours
                                finished_estimated = finished_estimated +
all_minus_last_track + tracks[1].estimated_hours
                                finished_ocuppied = finished_ocuppied + all_minus_last_track
+ tracks[0].expended_hours
                        except:
                            finished_estimated = finished_estimated +
task.original_estimation
                            finished_ocuppied = finished_ocuppied + tracks[0].expended_hours

    if finished_ocuppied == 0.0:
        return 1.0

    factor = finished_estimated/finished_ocuppied

    if factor == 0.0:
```

```
        return 1.0

    return factor

def getVelocidad_Desarrollo(project):

    velocity = 0.0
    finished_time = 0.0
    project_time = 0.0

    for feature in project.features:
        terminated_feature = 1
        pre_finished_time = 0.0
        for task in feature.tasks:
            track_query1 = Session.query(Track)
            tracks = track_query1.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            pre_project_time = 0.0
            if tracks:
                for track in tracks:
                    pre_project_time = pre_project_time + track.expended_hours

                if tracks[0].estimated_hours == 0:
                    pre_finished_time = pre_finished_time + pre_project_time
                else:
                    terminated_feature = 0

            project_time = project_time + pre_project_time

        if terminated_feature == 1 and get_feature_validity(feature.id) == "V":
            finished_time = finished_time + pre_finished_time

    if project_time == 0.0:
        return 0.0

    velocity =  finished_time/project_time
    return velocity

def getETA(project):

    ETA = 0.0
    estimated_hours = 0.0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                track_query = Session.query(Track)
                last_track = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).first()
                if last_track:
                    estimated_hours = estimated_hours + last_track.estimated_hours
                else:
                    estimated_hours = estimated_hours + task.original_estimation

    factor_correccion_estimacion = getFactor_Correccion_Estimacion(project)

    ETA = estimated_hours/factor_correccion_estimacion

    return ETA

def getAvance_Funcionalidad_Estimado_VS_Real(feature):

    AvanceEvsR = 0.0
    Advance_Total = 0.0
    Advance_and_Estimated_Total = 0.0

    if get_feature_validity(feature.id) == "V":
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
```

```
            pre_Advance_Total = 0.0
            if tracks:
                for track in tracks:
                    pre_Advance_Total = pre_Advance_Total + track.expended_hours
                Advance_Total = Advance_Total + pre_Advance_Total
                Advance_and_Estimated_Total = Advance_and_Estimated_Total +
(pre_Advance_Total + tracks[0].estimated_hours)

    if Advance_and_Estimated_Total == 0.0:
        return AvanceEvsR

    AvanceEvsR = (Advance_Total / Advance_and_Estimated_Total)*100

    return AvanceEvsR

def getAvance_Funcionalidad_Estimado_Original_VS_Actual(feature):

    AvanceEOvsA = 0.0
    Original_Estimation = 0.0
    Actual_Estimation = 0.0

    if get_feature_validity(feature.id) == "V":
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            pre_Actual_Estimation = 0.0
            if tracks:
                for track in tracks:
                    pre_Actual_Estimation = pre_Actual_Estimation + track.expended_hours
                Actual_Estimation = Actual_Estimation + (pre_Actual_Estimation +
tracks[0].estimated_hours)
            else:
                Actual_Estimation = Actual_Estimation + task.original_estimation

            Original_Estimation = Original_Estimation + task.original_estimation

    if  Actual_Estimation == 0.0:
        return AvanceEOvsA

    AvanceEOvsA = (Original_Estimation / Actual_Estimation)*100

    return AvanceEOvsA

def get_task_status(task,date):

    status = 0

    track_query = Session.query(Track)
    tracks = track_query.from_statement("SELECT * FROM tracks WHERE task_id=:task_id ORDER
BY reported,id DESC").params(task_id = task.id).all()
    if tracks:
        for track in tracks:
            if date >= track.when:
                if track.estimated_hours == 0:
                    status = 2
                    break
                else:
                    status = 1
                    break
    else:
        status = 0

    return status

def get_feature_status(feature,date):

    status = 0
    all_complete = 0
    all_tasks_status = 0
```

```
    if get_feature_validity(feature.id) == "V":
        for task in feature.tasks:
            all_complete = all_complete + 2
            all_tasks_status = all_tasks_status + get_task_status(task,date)

    if all_complete > 0:
        if all_complete == all_tasks_status:
            status = 2
        elif all_tasks_status > 0:
            status = 1

    return status

def get_notstarted_features(project,date):

    notstarted_features = 0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            if get_feature_status(feature,date) == 0:
                notstarted_features = notstarted_features + 1

    return notstarted_features

def get_started_features(project,date):

    started_features = 0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            if get_feature_status(feature,date) == 1:
                started_features = started_features + 1

    return started_features

def get_finished_features(project,date):

    finished_features = 0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            if get_feature_status(feature,date) == 2:
                finished_features = finished_features + 1

    return finished_features

def get_valid_dates(project,date):

    valid_dates = []
    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                if task.creation_date <= date and task.creation_date not in valid_dates:
                    valid_dates.append(task.creation_date)
                    valid_dates.sort()
                for track in task.tracks:
                    if track.when <= date and track.when not in valid_dates:
                        valid_dates.append(track.when)
                        valid_dates.sort()

    return valid_dates

def get_total_features(project,date):

    total_features = 0
    features_list = []
    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                if task.validity == "V":
                    if task.creation_date <= date and task.feature_id not in features_list:
```

```
                        features_list.append(task.feature_id)

    total_features = total_features + len(features_list)

    return total_features

def get_project_users(project,date):

    project_users = []
    uquery = Session.query(User)
    users = uquery.from_statement("SELECT id FROM users").all()
    if users:
        for user in users:
            project_users.append(user.id)

    return project_users

def user_has_tracks_on_date(project,date,user):

    has_tracks = 0
    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                if task.validity == "V":
                    for track in task.tracks:
                        if track.when == date:
                            for tuser in track.users:
                                if user == tuser.id:
                                    has_tracks = 1

    return has_tracks

def get_resources_usage(project,dates):

    resources_usage = []
    usage = 0.0
    for date in dates:
        project_users = get_project_users(project,date)
        for user in project_users:
            if user_has_tracks_on_date(project,date,user) == 1:
                usage = usage + 1
        usage_tmp = (usage/(len(project_users)*len(dates)))*100
        resources_usage.append(usage_tmp)

    return resources_usage

def draw_burn_down(project,date):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "area"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "features left"

    dates = get_valid_dates(project,date)
    if dates:
        for date_tmp in dates:
            actual_total_features = get_total_features(project,date_tmp)
```

```
            etree.SubElement(row1, "string").text = date_tmp
            etree.SubElement(row2, "number").text = str(actual_total_features -
get_finished_features(project,date_tmp))
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "features"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    return root_string

def draw_CFD(project,date):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "area"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Finished Features
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "finished features"
    # <chart_data> -> <row> : Y-Axis Data - Serie 2: Started Features
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "started features"
    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Not Started Features
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "not started features"

    dates = get_valid_dates(project,date)
    if dates:
        for date_tmp in dates:
            finished_features = get_finished_features(project,date_tmp)
            started_features = get_started_features(project,date_tmp)
            notstarted_features = get_total_features(project,date_tmp)
            etree.SubElement(row1, "string").text = date_tmp
            etree.SubElement(row2, "number").text = str(finished_features)
            etree.SubElement(row3, "number").text = str(finished_features +
started_features)
            etree.SubElement(row4, "number").text = str(notstarted_features)

    # End of <chart_data>
```

```
    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "features"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ffff00"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ff0000"
    # Tree End

    root_string = etree.tostring(root)

    return root_string

def draw_time_and_budget(project,date):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1: % Finished Features
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "% finished features"
    # <chart_data> -> <row> : Y-Axis Data - Serie 2: % Used resources
    row3 = etree.SubElement(chart_data, "row")
    string_row3 = etree.SubElement(row3, "string")
    string_row3.text = "% used resources"
    # <chart_data> -> <row> : Y-Axis Data - Serie 3: Ideal Line
    row4 = etree.SubElement(chart_data, "row")
    string_row4 = etree.SubElement(row4, "string")
    string_row4.text = "Ideal Line"

    dates = get_valid_dates(project,date)
    resources_usage = get_resources_usage(project,dates)
    if dates:
        counter = 0
        while counter < len(dates):
            finished_features = get_finished_features(project,dates[counter])*1.0
            total_features = get_total_features(project,dates[counter])*1.0
            finished_features_percent = (finished_features/total_features)*100.0
            ideal_line = counter*(100.0/(len(dates)-1))
            etree.SubElement(row1, "string").text = dates[counter]
            etree.SubElement(row2, "number").text = str(finished_features_percent)
            etree.SubElement(row3, "number").text = str(resources_usage[counter])
            etree.SubElement(row4, "number").text = str(ideal_line)
            counter = counter + 1
    # End of <chart_data>
```

```
    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', suffix='%', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "percent"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "000000"
    # Tree End

    root_string = etree.tostring(root)

    return root_string

def getFactor_Correccion_Estimacion_on_date(project,date):

    factor = 0.0
    finished_estimated = 0.0
    finished_ocuppied = 0.0

    for feature in project.features:
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks_tmp = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            tracks = []
            if tracks_tmp:
                counter = 0
                while counter < len(tracks_tmp):
                    if tracks_tmp[counter].when <= date:
                        tracks.append(tracks_tmp[counter])
                    counter = counter + 1
                if tracks:
                    if tracks[0].estimated_hours == 0:
                        try:
                            if tracks[1]:
                                all_minus_last_track = 0
                                for track in tracks:
                                    if track.id != tracks[0].id:
                                        all_minus_last_track = all_minus_last_track +
track.expended_hours
                                finished_estimated = finished_estimated +
all_minus_last_track + tracks[1].estimated_hours
                                finished_ocuppied = finished_ocuppied + all_minus_last_track
+ tracks[0].expended_hours
                        except:
                            finished_estimated = finished_estimated +
task.original_estimation
                            finished_ocuppied = finished_ocuppied + tracks[0].expended_hours

    if finished_ocuppied == 0.0:
        return 1.0

    factor = finished_estimated/finished_ocuppied

    if factor == 0.0:
        return 1.0

    return factor
```

```
def getVelocidad_Desarrollo_on_date(project,date):

    velocity = 0.0
    finished_time = 0.0
    project_time = 0.0

    for feature in project.features:
        terminated_feature = 1
        pre_finished_time = 0.0
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks_tmp = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            pre_project_time = 0.0
            tracks = []
            if tracks_tmp:
                counter = 0
                while counter < len(tracks_tmp):
                    if tracks_tmp[counter].when <= date:
                        tracks.append(tracks_tmp[counter])
                    counter = counter + 1
                if tracks:
                    for track in tracks:
                        pre_project_time = pre_project_time + track.expended_hours

                    if tracks[0].estimated_hours == 0:
                        pre_finished_time = pre_finished_time + pre_project_time
                    else:
                        terminated_feature = 0

            project_time = project_time + pre_project_time

        if terminated_feature == 1 and get_feature_validity(feature.id) == "V":
            finished_time = finished_time + pre_finished_time

    if project_time == 0.0:
        return 0.0

    velocity =  finished_time/project_time
    return velocity

def getETA_on_date(project,date):

    ETA = 0.0
    estimated_hours = 0.0

    for feature in project.features:
        if get_feature_validity(feature.id) == "V":
            for task in feature.tasks:
                track_query = Session.query(Track)
                tracks_tmp = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
                tracks = []
                if tracks_tmp:
                    counter = 0
                    while counter < len(tracks_tmp):
                        if tracks_tmp[counter].when <= date:
                            tracks.append(tracks_tmp[counter])
                        counter = counter + 1
                    if tracks:
                        estimated_hours = estimated_hours + tracks[0].estimated_hours
                    else:
                        if task.creation_date <= date:
                            estimated_hours = estimated_hours + task.original_estimation
                else:
                    if task.creation_date <= date:
                        estimated_hours = estimated_hours + task.original_estimation

    factor_correccion_estimacion = getFactor_Correccion_Estimacion_on_date(project,date)
```

```
    ETA = estimated_hours/factor_correccion_estimacion
    return ETA

def getAvance_Funcionalidad_Estimado_VS_Real_on_date(feature,date):

    AvanceEvsR = 0.0
    Advance_Total = 0.0
    Advance_and_Estimated_Total = 0.0

    if get_feature_validity(feature.id) == "V":
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks_tmp = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            tracks = []
            pre_Advance_Total = 0.0
            if tracks_tmp:
                counter = 0
                while counter < len(tracks_tmp):
                    if tracks_tmp[counter].when <= date:
                        tracks.append(tracks_tmp[counter])
                    counter = counter + 1
                if tracks:
                    for track in tracks:
                        pre_Advance_Total = pre_Advance_Total + track.expended_hours
                    Advance_Total = Advance_Total + pre_Advance_Total
                    Advance_and_Estimated_Total = Advance_and_Estimated_Total +
(pre_Advance_Total + tracks[0].estimated_hours)

    if Advance_and_Estimated_Total == 0.0:
        return AvanceEvsR

    AvanceEvsR = (Advance_Total / Advance_and_Estimated_Total)*100

    return AvanceEvsR

def getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(feature,date):

    AvanceEOvsA = 0.0
    Original_Estimation = 0.0
    Actual_Estimation = 0.0

    if get_feature_validity(feature.id) == "V":
        for task in feature.tasks:
            track_query = Session.query(Track)
            tracks_tmp = track_query.from_statement("SELECT * FROM tracks WHERE task_id
=:task_id ORDER BY reported,id DESC").params(task_id = task.id).all()
            tracks = []
            pre_Actual_Estimation = 0.0
            if tracks_tmp:
                counter = 0
                while counter < len(tracks_tmp):
                    if tracks_tmp[counter].when <= date:
                        tracks.append(tracks_tmp[counter])
                    counter = counter + 1
                if tracks:
                    for track in tracks:
                        pre_Actual_Estimation = pre_Actual_Estimation + track.expended_hours
                    Actual_Estimation = Actual_Estimation + (pre_Actual_Estimation +
tracks[0].estimated_hours)
                else:
                    Actual_Estimation = Actual_Estimation + task.original_estimation
            else:
                if task.creation_date <= date:
                    Actual_Estimation = Actual_Estimation + task.original_estimation

            if task.creation_date <= date:
                Original_Estimation = Original_Estimation + task.original_estimation

    if  Actual_Estimation == 0.0:
        return AvanceEOvsA
```

```
    AvanceEOvsA = (Original_Estimation / Actual_Estimation)*100

    return AvanceEOvsA

def draw_factor_correccion_historial(project,dates):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Factor de correccion
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "factor de correccion"

    if dates:
        counter = 0
        while counter < len(dates):
            etree.SubElement(row1, "string").text = dates[counter]
            etree.SubElement(row2, "number").text =
str(getFactor_Correccion_Estimacion_on_date(project,dates[counter]))
            counter = counter + 1
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "factor de correccion"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)
    return root_string

def draw_velocidad_desarrollo_historial(project,dates):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
```

```
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1: Velocidad de Desarrollo
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "earned value"

    if dates:
        counter = 0
        while counter < len(dates):
            etree.SubElement(row1, "string").text = dates[counter]
            etree.SubElement(row2, "number").text =
str(getVelocidad_Desarrollo_on_date(project,dates[counter]))
            counter = counter + 1
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "velocidad de desarrollo"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    return root_string

def draw_eta_historial(project,dates):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "line"
    axis_category = etree.SubElement(root, "axis_category", orientation="diagonal_up",
size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    # <chart_data> -> <row> : Y-Axis Data - Serie 1: E.T.A.
    row2 = etree.SubElement(chart_data, "row")
    string_row2 = etree.SubElement(row2, "string")
    string_row2.text = "estimated times of arrival"

    if dates:
        counter = 0
        while counter < len(dates):
            etree.SubElement(row1, "string").text = dates[counter]
            etree.SubElement(row2, "number").text =
str(getETA_on_date(project,dates[counter]))
            counter = counter + 1
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='above', size='12',
color='0000ff', decimals='2')

    draw = etree.SubElement(root, "draw")
```

```
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "E.T.A.(hours)"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "dates"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    # Tree End

    root_string = etree.tostring(root)

    return root_string

def draw_avance_real_vs_estimado_historial(project,dates):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "bar"
    axis_category = etree.SubElement(root, "axis_category", size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    if dates:
        counter = 0
        while counter < len(dates):
            etree.SubElement(row1, "string").text = dates[counter]
            counter = counter + 1

    n = 1
    for feature in project.features:
        # <chart_data> -> <row> : Y-Axis Data - Serie n: Avance Estimado VS Real Producto n
        row = etree.SubElement(chart_data, "row")
        etree.SubElement(row, "string").text = "feature " + str(n)
        if dates:
            counter2 = 0
            while counter2 < len(dates):
                etree.SubElement(row, "number").text =
str(getAvance_Funcionalidad_Estimado_VS_Real_on_date(feature,dates[counter2]))
                counter2 = counter2 + 1
        n = n + 1
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='outside', size='12',
color='0000ff', decimals='2', suffix="%")

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "dates"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "avance x funcionalidad:  tiempo estimado vs tiempo real"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
```

```
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ffff00"
    color4 = etree.SubElement(series_color, "color")
    color4.text = "00ffff"
    # Tree End

    root_string = etree.tostring(root)
    return root_string

def draw_avance_original_vs_actual_historial(project,dates):

    # Tree Start
    root = etree.Element("chart")
    chart_type = etree.SubElement(root, "chart_type")
    chart_type.text = "bar"
    axis_category = etree.SubElement(root, "axis_category", size="12")
    axis_value = etree.SubElement(root, "axis_value", size="12", suffix="%")

    # Start of <chart_data>
    chart_data = etree.SubElement(root, "chart_data")

    # <chart_data> -> <row> : X-Axis Data
    row1 = etree.SubElement(chart_data, "row")
    null_root1 = etree.SubElement(row1, "null")
    if dates:
        counter = 0
        while counter < len(dates):
            etree.SubElement(row1, "string").text = dates[counter]
            counter = counter + 1

    n = 1
    for feature in project.features:
        # <chart_data> -> <row> : Y-Axis Data - Serie n: Avance Estimado VS Real Producto n
        row = etree.SubElement(chart_data, "row")
        etree.SubElement(row, "string").text = "feature " + str(n)
        if dates:
            counter2 = 0
            while counter2 < len(dates):
                etree.SubElement(row, "number").text =
str(getAvance_Funcionalidad_Estimado_Original_VS_Actual_on_date(feature,dates[counter2]))
                counter2 = counter2 + 1
        n = n + 1
    # End of <chart_data>

    chart_value = etree.SubElement(root, "chart_value", position='outside', size='12',
color='0000ff', decimals='2', suffix="%")

    draw = etree.SubElement(root, "draw")
    text1 = etree.SubElement(draw, "text", v_align='middle', bold='true', color='ffffff',
h_align='center', width='600', height='50', x='0', y='600', alpha='15', rotation='-90',
font='arial', size='30')
    text1.text = "dates"
    text2 = etree.SubElement(draw, "text", v_align='bottom', bold='true', color='ffffff',
h_align='center', width='800', height='50', x='0', y='550', alpha='15', rotation='0',
font='arial', size='30')
    text2.text = "avance x funcionalidad: tiempo estimado orig. vs actual"

    series_color = etree.SubElement(root, "series_color")
    color1 = etree.SubElement(series_color, "color")
    color1.text = "77bb11"
    color2 = etree.SubElement(series_color, "color")
    color2.text = "ff0000"
    color3 = etree.SubElement(series_color, "color")
    color3.text = "ffff00"
    color4 = etree.SubElement(series_color, "color")
    color4.text = "00ffff"
    # Tree End

    root_string = etree.tostring(root)
    return root_string
```

## 10.5. XML's para la generación de los ejemplos presentados

**avance_estimado_vs_real_historial.xml**

```
<chart>
  <chart_type>bar</chart_type>
  <axis_category size="12"/>
  <axis_value suffix="%" size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>feature 1</string>
      <number>0.0</number>
      <number>33.3333333333</number>
      <number>50.0</number>
      <number>100.0</number>
      <number>100.0</number>
      <number>100.0</number>
      <number>100.0</number>
    </row>
    <row>
      <string>feature 2</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>100.0</number>
      <number>100.0</number>
      <number>100.0</number>
      <number>100.0</number>
      <number>100.0</number>
    </row>
    <row>
      <string>feature 3</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.0</number>
      <number>46.1538461538</number>
      <number>60.8695652174</number>
      <number>100.0</number>
      <number>100.0</number>
    </row>
    <row>
      <string>feature 4</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.0</number>
      <number>46.1538461538</number>
      <number>60.8695652174</number>
      <number>100.0</number>
    </row>
  </chart_data>
  <chart_value color="0000ff" position="outside" decimals="2" suffix="%" size="12"/>
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">dates</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">avance x
funcionalidad: tiempo estimado vs tiempo real</text>
  </draw>
  <series_color>
```

```
    <color>77bb11</color>
    <color>ff0000</color>
    <color>ffff00</color>
    <color>00ffff</color>
  </series_color>
</chart>
```

## avance_original_vs_actual_historial.xml

```
<chart>
  <chart_type>bar</chart_type>
  <axis_category size="12"/>
  <axis_value suffix="%" size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>feature 1</string>
      <number>100.0</number>
      <number>50.0</number>
      <number>37.5</number>
      <number>21.4285714286</number>
      <number>21.4285714286</number>
      <number>21.4285714286</number>
      <number>21.4285714286</number>
    </row>
    <row>
      <string>feature 2</string>
      <number>100.0</number>
      <number>100.0</number>
      <number>43.75</number>
      <number>43.75</number>
      <number>43.75</number>
      <number>43.75</number>
      <number>43.75</number>
    </row>
    <row>
      <string>feature 3</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.0</number>
      <number>11.5384615385</number>
      <number>6.52173913043</number>
      <number>10.0</number>
      <number>10.0</number>
    </row>
    <row>
      <string>feature 4</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.0</number>
      <number>100.0</number>
      <number>26.9230769231</number>
      <number>15.2173913043</number>
      <number>23.3333333333</number>
    </row>
  </chart_data>
  <chart_value color="0000ff" position="outside" decimals="2" suffix="%" size="12"/>
  <draw>
```

```
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">dates</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">avance x
funcionalidad: tiempo estimado orig. vs actual</text>
  </draw>
  <series_color>
    <color>77bb11</color>
    <color>ff0000</color>
    <color>ffff00</color>
    <color>00ffff</color>
  </series_color>
</chart>
```

## burndown.xml

```
<chart>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>finished features</string>
      <number>2</number>
      <number>2</number>
      <number>1</number>
      <number>2</number>
      <number>2</number>
      <number>1</number>
      <number>0</number>
    </row>
  </chart_data>
  <chart_type>area</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value size="12"/>
  <chart_value color="0000ff" position="above" size="12"/>
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">features</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
  </series_color>
</chart>
```

## CFD.xml

```
<chart>
  <chart_type>area</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
```

```
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>finished features</string>
      <number>0</number>
      <number>0</number>
      <number>1</number>
      <number>2</number>
      <number>2</number>
      <number>3</number>
      <number>4</number>
    </row>
    <row>
      <string>started features</string>
      <number>0</number>
      <number>1</number>
      <number>2</number>
      <number>3</number>
      <number>4</number>
      <number>4</number>
      <number>4</number>
    </row>
    <row>
      <string>not started features</string>
      <number>2</number>
      <number>2</number>
      <number>2</number>
      <number>4</number>
      <number>4</number>
      <number>4</number>
      <number>4</number>
    </row>
  </chart_data>
  <chart_value position="above" size="12" color="0000ff" />
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">features</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
    <color>ffff00</color>
    <color>ff0000</color>
  </series_color>
</chart>
```

## eta_historial.xml

```
<chart>
  <chart_type>line</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>estimated times of arrival</string>
      <number>10.0</number>
      <number>11.0</number>
```

```
      <number>4.57142857143</number>
      <number>28.6363636364</number>
      <number>43.6363636364</number>
      <number>15.8823529412</number>
      <number>0.0</number>
    </row>
  </chart_data>
  <chart_value color="0000ff" position="above" decimals="2" size="12"/>
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial"
size="30">E.T.A.(hours)</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
  </series_color>
</chart>
```

## factor_correccion_historial.xml

```
<chart>
  <chart_type>line</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>factor de correccion</string>
      <number>1.0</number>
      <number>1.0</number>
      <number>0.875</number>
      <number>0.733333333333</number>
      <number>0.733333333333</number>
      <number>1.13333333333</number>
      <number>1.26666666667</number>
    </row>
  </chart_data>
  <chart_value color="0000ff" position="above" size="12" decimals="2" />
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">factor de
correccion</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
  </series_color>
</chart>
```

## timeandbudget.xml

```
<chart>
  <chart_type>line</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value suffix="%" size="12"/>
  <chart_data>
```

```
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>% finished features</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>50.0</number>
      <number>50.0</number>
      <number>50.0</number>
      <number>75.0</number>
      <number>100.0</number>
    </row>
    <row>
      <string>% used resources</string>
      <number>0.0</number>
      <number>7.14285714286</number>
      <number>21.4285714286</number>
      <number>28.5714285714</number>
      <number>42.8571428571</number>
      <number>57.1428571429</number>
      <number>64.2857142857</number>
    </row>
    <row>
      <string>Ideal Line</string>
      <number>0.0</number>
      <number>16.6666666667</number>
      <number>33.3333333333</number>
      <number>50.0</number>
      <number>66.6666666667</number>
      <number>83.3333333333</number>
      <number>100.0</number>
    </row>
  </chart_data>
       <chart_value position="above" size="12" color="0000ff" suffix="%" decimals="2"/>
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">percent</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
    <color>ff0000</color>
    <color>000000</color>
  </series_color>
</chart>
```

## velocidad_desarrollo_historial.xml

```
<chart>
  <chart_type>line</chart_type>
  <axis_category orientation="diagonal_up" size="12"/>
  <axis_value size="12"/>
  <chart_data>
    <row>
      <null/>
      <string>2007-11-06</string>
      <string>2007-11-07</string>
      <string>2007-11-08</string>
```

```
      <string>2007-11-09</string>
      <string>2007-11-10</string>
      <string>2007-11-11</string>
      <string>2007-11-12</string>
    </row>
    <row>
      <string>earned value</string>
      <number>0.0</number>
      <number>0.0</number>
      <number>0.8</number>
      <number>0.714285714286</number>
      <number>0.428571428571</number>
      <number>0.681818181818</number>
      <number>1.0</number>
    </row>
  </chart_data>
  <chart_value color="0000ff" position="above" decimals="2" size="12"/>
  <draw>
    <text v_align="middle" bold="true" color="ffffff" h_align="center" height="50"
width="600" alpha="15" y="600" x="0" rotation="-90" font="arial" size="30">velocidad de
desarrollo</text>
    <text v_align="bottom" bold="true" color="ffffff" h_align="center" height="50"
width="800" alpha="15" y="550" x="0" rotation="0" font="arial" size="30">dates</text>
  </draw>
  <series_color>
    <color>77bb11</color>
  </series_color>
</chart>
```

## 10.6. Páginas HTML que grafican los ejemplos presentados en base a los XML's anteriores

**avance_estimado_vs_real_historial.html**

```
<HTML>
<BODY bgcolor="#FFFFFF">

       <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

       codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
              WIDTH="800"
              HEIGHT="600"
              id="charts"
              ALIGN="center">
       <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=avance_estimado_vs_real_historial.x
ml">
       <PARAM NAME=quality VALUE=high>
       <PARAM NAME=bgcolor VALUE=#666666>

       <EMBED
       src="charts.swf?library_path=charts_library&xml_source=avance_estimado_vs_real_histo
rial.xml"
              quality=high bgcolor=#666666
              WIDTH="800"
              HEIGHT="600"
              NAME="charts"
              ALIGN="center"
              swLiveConnect="true"
              TYPE="application/x-shockwave-flash"
              PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
```

```
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## avance_original_vs_actual_historial.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=avance_original_vs_actual_historial
.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED
        src="charts.swf?library_path=charts_library&xml_source=avance_original_vs_actual_his
torial.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## burndown.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=burndown.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED  src="charts.swf?library_path=charts_library&xml_source=burndown.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
                swLiveConnect="true"
```

```
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## CFD.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie VALUE="charts.swf?library_path=charts_library&xml_source=CFD.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED  src="charts.swf?library_path=charts_library&xml_source=CFD.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## eta_historial.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=eta_historial.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED  src="charts.swf?library_path=charts_library&xml_source=eta_historial.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
```

```
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## factor_correccion_historial.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=factor_correccion_historial.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED
        src="charts.swf?library_path=charts_library&xml_source=factor_correccion_historial.x
ml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## timeandbudget.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=timeandbudget.xml">
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED  src="charts.swf?library_path=charts_library&xml_source=timeandbudget.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
```

```
                ALIGN="center"
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```

## velocidad_desarrollo_historial.html

```
<HTML>
<BODY bgcolor="#FFFFFF">

        <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"

        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=6,0,0,0"
                WIDTH="800"
                HEIGHT="600"
                id="charts"
                ALIGN="center">
        <PARAM NAME=movie
VALUE="charts.swf?library_path=charts_library&xml_source=velocidad_desarrollo_historial.xml"
>
        <PARAM NAME=quality VALUE=high>
        <PARAM NAME=bgcolor VALUE=#666666>

        <EMBED
        src="charts.swf?library_path=charts_library&xml_source=velocidad_desarrollo_historia
l.xml"
                quality=high bgcolor=#666666
                WIDTH="800"
                HEIGHT="600"
                NAME="charts"
                ALIGN="center"
                swLiveConnect="true"
                TYPE="application/x-shockwave-flash"
                PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
        </EMBED>
        </OBJECT>

</BODY>
</HTML>
```