# Índice

[Índice 1](#_Toc230478447)

[1.- Diseño de indicadores 2](#_Toc230478448)

[1.1.- Escenario 2](#_Toc230478449)

[1.2.- Objetivo 2](#_Toc230478450)

[1.2.1.- Indicador de funcionalidad completa 2](#_Toc230478451)

[1.2.2.- Análisis preliminar del indicador 3](#_Toc230478452)

[1.2.3.- Agregado de liberaciones al usuario en el gráfico de FC 4](#_Toc230478453)

[1.3.- Resolución 5](#_Toc230478454)

[1.3.1.- Hipótesis y supuestos 5](#_Toc230478455)

[1.3.2.- Indicador de funcionalidad completa 5](#_Toc230478456)

[1.3.3.- Análisis preliminar del indicador 8](#_Toc230478457)

[1.3.4.- Agregado de liberaciones al usuario en el gráfico de FC 9](#_Toc230478458)

[2.- Análisis de Indicadores 11](#_Toc230478459)

[2.1.- Escenario 11](#_Toc230478460)

[2.1.1.- Indicador de funcionalidad completa 12](#_Toc230478461)

[2.1.2.- Indicador de Evolución de la prueba 13](#_Toc230478462)

[2.1.3.- Indicador de Cobertura de la Prueba 14](#_Toc230478463)

[2.2.- Objetivo 15](#_Toc230478464)

[2.3.- Resolución 16](#_Toc230478465)

[2.3.1.- Hipótesis y supuestos 16](#_Toc230478466)

[2.3.2.- Análisis de los indicadores 17](#_Toc230478467)

[2.3.3.- Acciones correctivas 22](#_Toc230478468)

# 1.- Diseño de indicadores

## 1.1.- Escenario

El mismo escenario de WBS para el Proyecto “Sistema UARTN”.

## 1.2.- Objetivo

Se decide implementar un indicador que permita medir objetivamente el avance del Proyecto. Se utilizará un indicador del tipo Funcionalidad Completa. Se pide:

### 1.2.1.- Indicador de funcionalidad completa

Construya el indicador a través de los siguientes pasos:

· Dividir al producto en partes.

· Poner un peso a cada una de esas partes entre 1 y 3 de acuerdo a su complejidad.

· Estimar la fecha de liberación de la funcionalidad por parte del equipo de Desarrollo.

· Estimar la fecha de aprobación de la funcionalidad por parte de QC

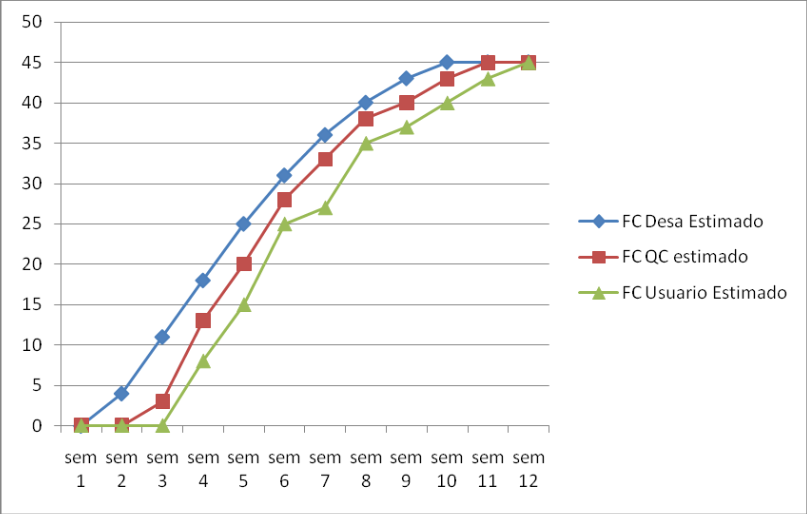
· Graficar el indicador con las fechas presupuestadas en donde:

- El eje horizontal indique fechas

- El eje vertical indique puntos de funcionalidad acumulados. Los puntos de funcionalidad se obtienen de sumar la complejidad estimada a la fecha.

- Existan dos curvas: una de liberación estimada de desarrollo y otra de liberación estimada de QC. Ambas curvas deben llegar a la suma de puntos de funcionalidad.

#### Ejemplo de gráfico de FC



### 1.2.2.- Análisis preliminar del indicador

Analice el indicador con el objetivo de verificar si el calendario está correctamente armado.

· Analice la separación de las curvas y efectúe correcciones

· Analice la pendiente de las curvas y efectúe correcciones

· Analice otros puntos que considere convenientes

· Analice si este indicador le servirá para controlar el Proyecto. Ventajas y desventajas

### 1.2.3.- Agregado de liberaciones al usuario en el gráfico de FC

Agregue las curvas de funcionalidad liberada al usuario y aprobada por el usuario según indica el calendario que armó para el Cliente

· Grafique las curvas nuevas

· Analice

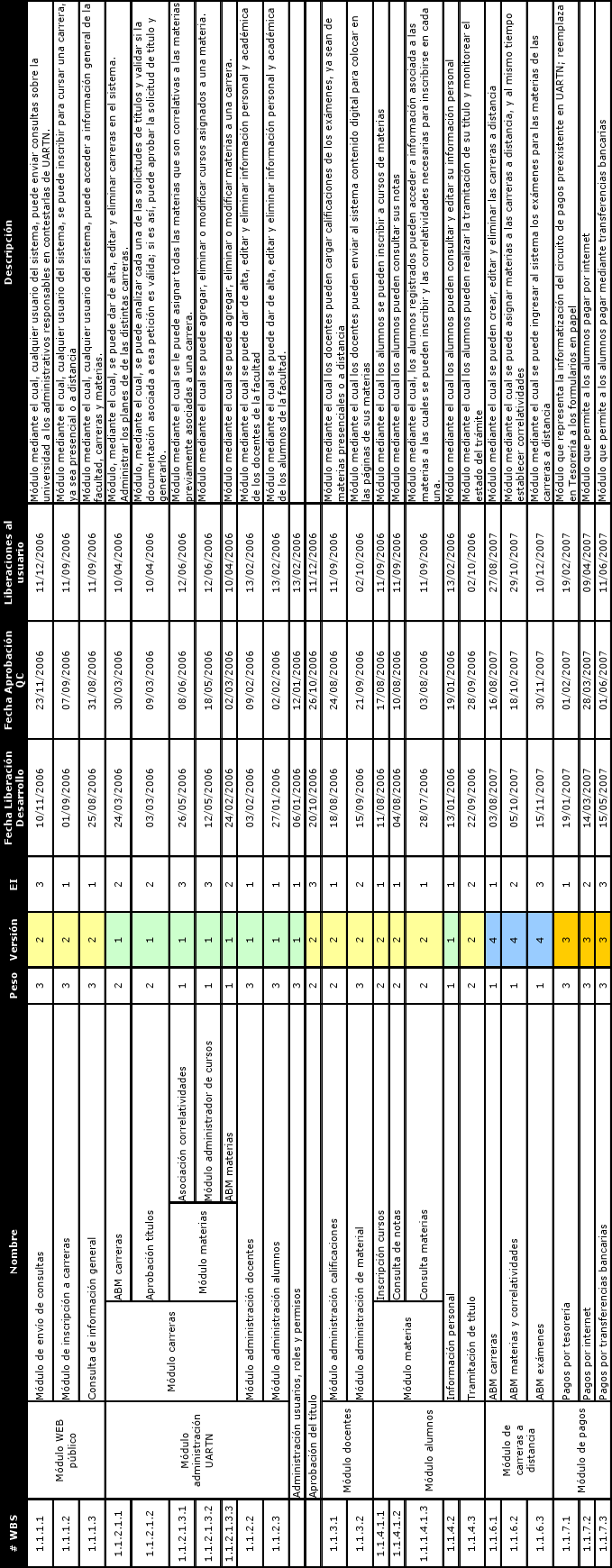
## 1.3.- Resolución

1.3.1.- Hipótesis y supuestos

Se considera para esta actividad todas las versiones del proyecto.

1.3.2.- Indicador de funcionalidad completa

Funcionalidades (ver página siguiente)



Acumulado y fechas

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Peso** | **Peso acumulado** | **Fecha Liberación Desarrollo** | **Semana** | **Fecha Aprobación QC** | **Semana** | **Liberaciones al usuario** | **Semana** |
| 3 | 3 | 06/01/2006 | 1 | 12/01/2006 | 2 | 13/02/2006 |  |
| 1 | 4 | 13/01/2006 | 2 | 19/01/2006 | 3 | 13/02/2006 |  |
| 3 | 7 | 27/01/2006 | 4 | 02/02/2006 | 5 | 13/02/2006 |  |
| 3 | 10 | 03/02/2006 | 5 | 09/02/2006 | 6 | 13/02/2006 | 7 |
| 1 | 11 | 24/02/2006 | 8 | 02/03/2006 | 9 | 10/04/2006 |  |
| 2 | 13 | 03/03/2006 | 9 | 09/03/2006 | 10 | 10/04/2006 |  |
| 2 | 15 | 24/03/2006 | 12 | 30/03/2006 | 13 | 10/04/2006 | 15 |
| 1 | 16 | 12/05/2006 | 19 | 18/05/2006 | 20 | 12/06/2006 |  |
| 1 | 17 | 26/05/2006 | 21 | 08/06/2006 | 23 | 12/06/2006 | 24 |
| 2 | 19 | 28/07/2006 | 30 | 03/08/2006 | 31 | 11/09/2006 |  |
| 2 | 21 | 04/08/2006 | 31 | 10/08/2006 | 32 | 11/09/2006 |  |
| 2 | 23 | 11/08/2006 | 32 | 17/08/2006 | 33 | 11/09/2006 |  |
| 2 | 25 | 18/08/2006 | 33 | 24/08/2006 | 34 | 11/09/2006 |  |
| 3 | 28 | 25/08/2006 | 34 | 31/08/2006 | 35 | 11/09/2006 |  |
| 3 | 31 | 01/09/2006 | 35 | 07/09/2006 | 36 | 11/09/2006 | 37 |
| 3 | 34 | 15/09/2006 | 37 | 21/09/2006 | 38 | 02/10/2006 |  |
| 2 | 36 | 22/09/2006 | 38 | 28/09/2006 | 39 | 02/10/2006 | 40 |
| 2 | 38 | 20/10/2006 | 42 | 26/10/2006 | 43 | 11/12/2006 |  |
| 3 | 41 | 10/11/2006 | 45 | 23/11/2006 | 47 | 11/12/2006 | 50 |
| 3 | 44 | 19/01/2007 | 55 | 01/02/2007 | 57 | 19/02/2007 | 60 |
| 3 | 47 | 14/03/2007 | 63 | 28/03/2007 | 65 | 09/04/2007 | 67 |
| 3 | 50 | 15/05/2007 | 72 | 01/06/2007 | 74 | 11/06/2007 | 76 |
| 1 | 51 | 03/08/2007 | 83 | 16/08/2007 | 85 | 27/08/2007 | 87 |
| 1 | 52 | 05/10/2007 | 92 | 18/10/2007 | 94 | 29/10/2007 | 96 |
| 1 | 53 | 15/11/2007 | 98 | 30/11/2007 | 100 | 10/12/2007 | 102 |

## 

Gráfico



Eje X: cantidad de semanas transcurridas

Eje Y: puntos de funcionalidad acumulados

1.3.3.- Análisis preliminar del indicador

Como se puede apreciar en el gráfico, las curvas de “Fecha Liberación Desarrollo” y “Fecha Aprobación QC” figuran muy próximas una de otra. Esto quiere decir que punto de funcionalidad a punto de funcionalidad, las fechas de liberación de desarrollo están muy próximas a las fechas de aprobación de QC, lo que claramente implica que no se espera encontrar en la etapa de QC grandes problemas o complejidades al momento de analizar pensando en aprobar que demoren mucho la liberación. Esto también se traduce en menos demora en la puesta en producción definitiva de la versión, lo cual en principio es bueno, ya que permite que el cliente se encuentre con un retorno de la inversión lo antes posible.

Respecto a las pendientes, se puede observar que cerca de la semana 30 se ve un incremento de la pendiente, lo que quiere decir que en la semana 30 se da inicio a una serie de pocas semanas en que se debe acumular con mucha mayor velocidad puntos de funcionalidad, lo cual en principio no necesariamente es malo, pero ciertamente llama la atención. No se considera que sea un reflejo de que el calendario está mal armado, ya que se supone que el equipo de desarrollo en ese momento podrá hacer frente a ese desafío, ya que las etapas previas de planificación y análisis no arrojaron ni mostraron ninguna anomalía.

Este indicador sirve para controlar el proyecto en términos del desempeño de QC y del área de desarrollo, es decir, permite medir claramente cuánta distancia hay entre que se libera una funcionalidad y que QC la aprueba; alguna demora detectada en esta aprobación puede darse tanto a impericia en QC para terminar en tiempo como a impericia del área de desarrollo por liberar funcionalidad que impide que se la controle de una manera medianamente aceptable. No dá de manera directa un indicio sobre cuánto falta para terminar el proyecto, siendo esto una desventaja. Por otro lado no queda claro en qué etapa del proyecto es válido, y si bien una funcionalidad aprobada por QC no debería presentar errores críticos, no queda claro si QC en este caso también controla que la funcionalidad realmente esté completa.

1.3.4.- Agregado de liberaciones al usuario en el gráfico de FC

Gráfico



Eje X: cantidad de semanas transcurridas

Eje Y: puntos de funcionalidad acumulados

Como en el caso anterior, se puede observar que no se prevee que haya mucha demora entre las fechas de aprobación de QC en los distintos puntos de funcionalidad acumulados y la liberación al usuario, lo cual nuevamente implica que no se considera que ocurrirán problemas graves en las etapas contempladas, que obliguen a demorar la puesta en producción todavía más.

E igualmente como en el caso anterior, se ve que cerca de la semana 30 incrementa la velocidad con que se libera al usuario la funcionalidad. Se considera que el mencionado evento será manejado correctamente.

Este caso es distinto del anterior, ya que si bien las primeras dos curvas consideradas brindan la misma información que en el caso anterior, la tercera curva brinda información sobre cuándo se liberan los productos al usuario, lo cual permite mínimamente llevar una cuenta de con qué velocidad se pone funcionalidad frente al usuario, si bien esto no es de por sí un cálculo directo de lo que falta para completar el proyecto.2.- Análisis de Indicadores

## 2.1.- Escenario

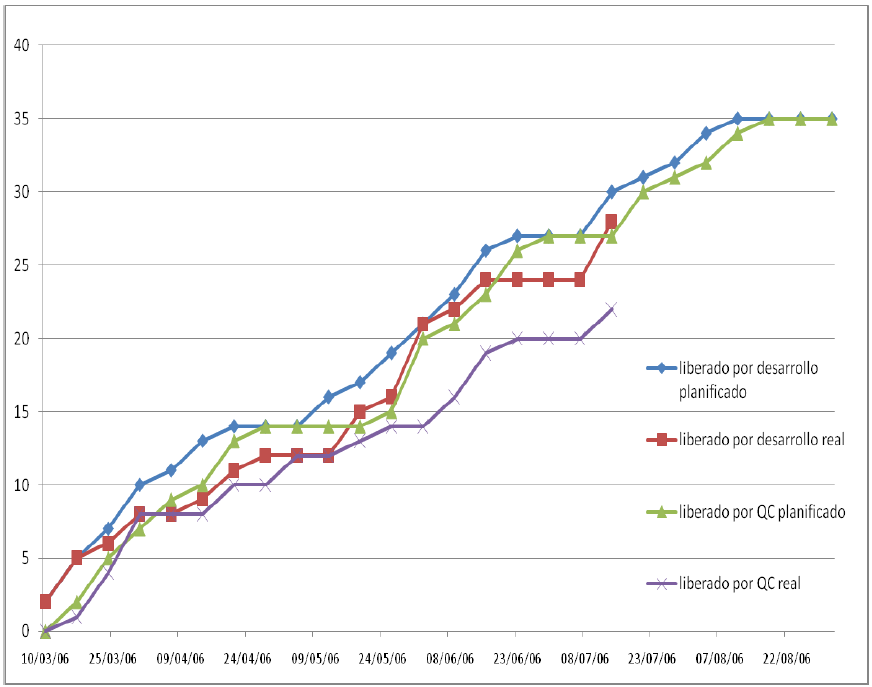
El mismo escenario de WBS para el proyecto “Sistema UARTN”.

La etapa de construcción de esta primera versión se inició el 1 de marzo de 2006. Previamente hubo una etapa de análisis y diseño global de 2 meses aproximadamente.

Hoy es 14 de julio y usted, Jefe del Proyecto, está analizando los indicadores del Proyecto con el objetivo de entender cómo se encuentra posicionado para los próximos hitos.

* 11/08/2006: Será la tercera liberación para aceptación de usuarios de UARTN.
* 15/09/2006: En esta fecha se liberará la primera versión para instalar en UARTN. Esta fecha es inamovible, se trata de un compromiso adquirido con la universidad.

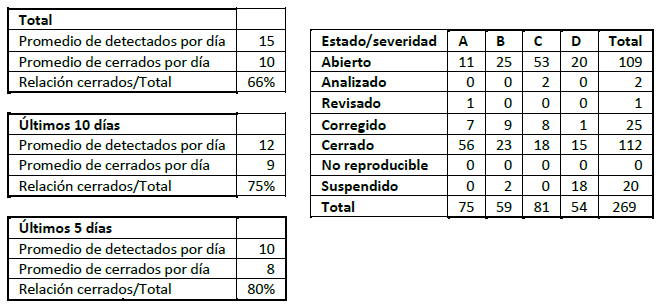
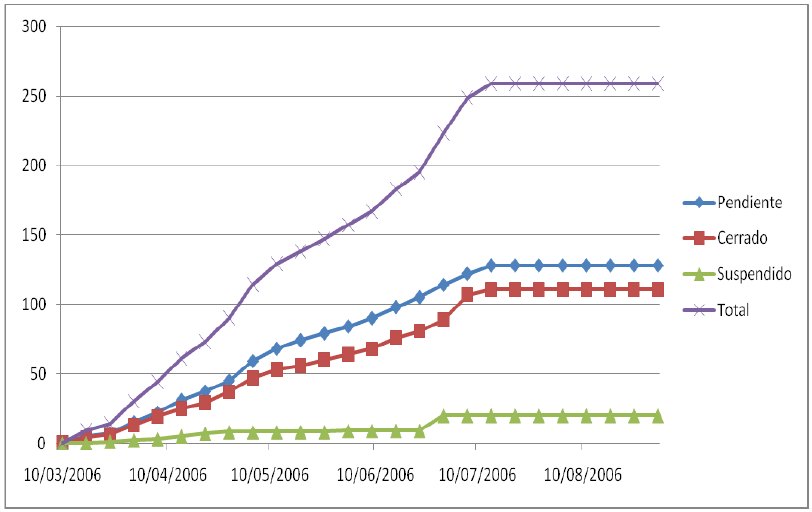
### 2.1.1.- Indicador de funcionalidad completa



El indicador de FC muestra 4 curvas:

* **Código Completo – Puntos estimados acumulados:** Liberación planificada de funcionalidad por el equipo de desarrollo.
* **Código Completo – Puntos reales acumulados:** Liberación real de funcionalidad por el equipo de desarrollo.
* **Funcionalidad Liberada – Puntos estimados acumulados:** Liberación planificada de funcionalidad por el equipo de QC.
* **Funcionalidad Liberada – Puntos reales acumulados:** Liberación real de funcionalidad por el equipo de QC.

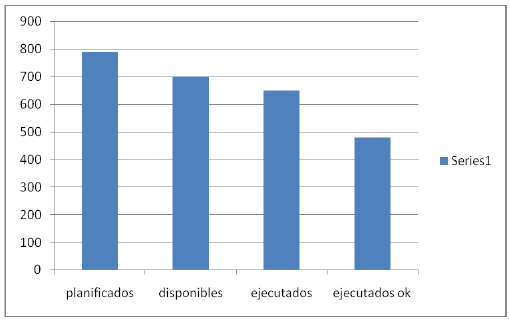
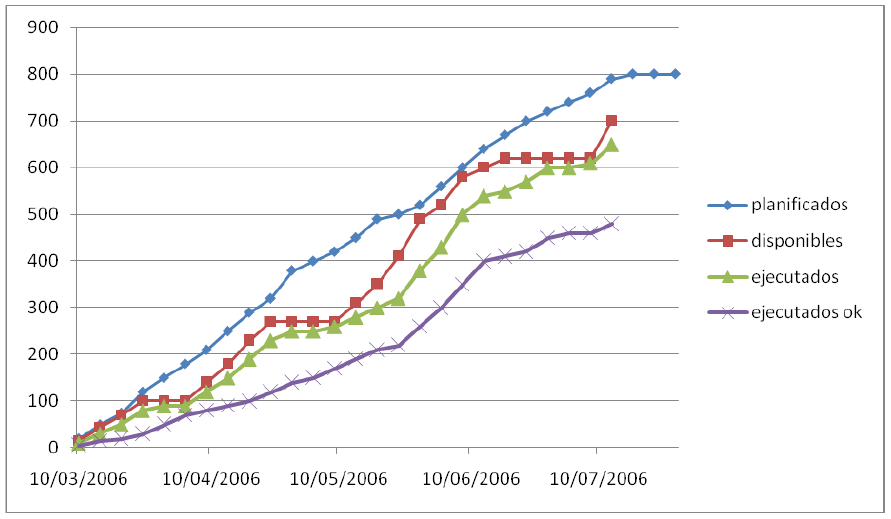
### 2.1.2.- Indicador de Evolución de la prueba



El indicador de Evolución de la Prueba muestra 4 curvas:

* **Total:** cantidad total de defectos acumulada.
* **Cerrados:** Cantidad acumulada de defectos cerrados por el equipo de QC.
* **Pendientes:** Cantidad acumulada de defectos pendientes (defectos abiertos sin corregir, corregidos por revisar por QC, analizados por desarrollo, etc).
* **Suspendidos:** Defectos suspendidos que quedarán para la próxima versión.

### 2.1.3.- Indicador de Cobertura de la Prueba



El indicador de Cobertura de la Prueba muestra 4 valores:

* **Planificados:** Cantidad de casos de prueba planificados a ejecutar.
* **Disponibles:** Cantidad de casos de prueba disponibles. Son los casos de prueba que pueden ejecutarse ya que el equipo de desarrollo liberó la funcionalidad a la que hacen referencia.
* **Cumplidos:** Cantidad de casos de prueba ejecutados por el equipo de QC.
* **Ejecutados OK:** Cantidad de casos de prueba ejecutados en los que no se detectaron defectos.
* **Totales:** Cantidad de casos totales, incluye los planificados más los que aún no se han escrito pero se estima se escribirán. No figura el valor en el gráfico, pero se estiman alrededor de 1.100.

## 2.2.- Objetivo

Se pide analizar los indicadores en relación a los próximos objetivos del Proyecto, identificados por los dos hitos pendientes. El análisis debe incluir:

* Análisis detallado de cada Indicador, conteniendo:
  + ¿Qué se lee?
  + ¿Cuáles son las posibles causas?
  + ¿Cuáles son los posibles impactos?
* Acciones correctivas
  + Identifique las acciones correctivas a ejecutar

## 2.3.- Resolución

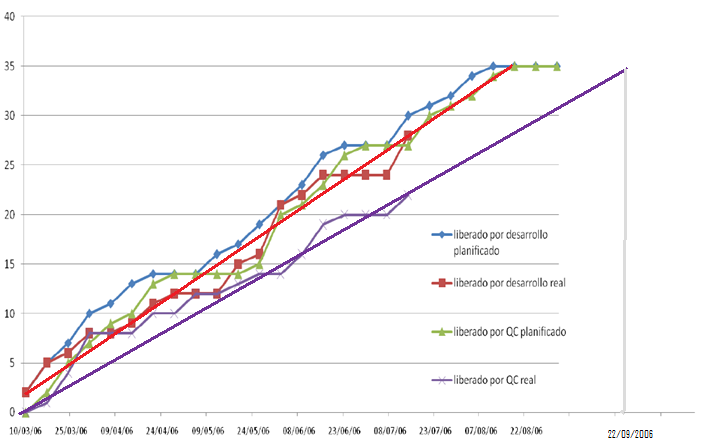
### 2.3.1.- Hipótesis y supuestos

* No se hacen análisis de FC y Errores por módulos/grupos porque no hay información sobre módulos/grupos, ni se diferencian las funcionalidades y errores por módulos/grupos.

### 2.3.2.- Análisis de los indicadores

#### 2.3.2.1.- Funcionalidad Completa

#### ¿Qué se lee?



El indicador muestra que el equipo de desarrollo tiene un ritmo de liberación de código similar al planificado.

Si bien tiene pequeños desvíos, se ve que si se traza una recta con la pendiente relativa a la velocidad de liberación, se puede estimar que van a terminar de acuerdo a lo planificado.

Con respecto al equipo de QC se puede ver que a partir del 24/05/06 se apartaron significativamente de lo planificado.

Inicialmente tienen un atraso de 15 días en la liberación de funcionalidad y se ve que este se empieza a hacer cada vez mayor, porque si se traza una recta con la pendiente relativa a la velocidad de liberación, se ve que va a liberar todo un mes más tarde que lo planificado.

#### ¿Cuáles son las posibles causas?

Una causa posible es que los desarrolladores, a pesar de seguir adelante con el desarrollo dejen muchos defectos abiertos y no se dediquen a cerrarlos, por lo que el equipo de QC no pueda liberar ninguna funcionalidad.

Otra causa podría ser que el equipo de QC se haya atrasado en la ejecución o diseño de los test, esta información podrá obtenerse de los indicadores que faltan analizar.

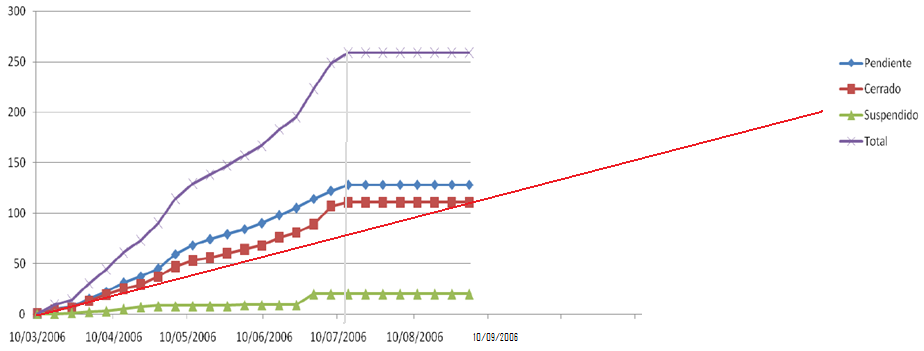
#### ¿Cuáles son los posibles impactos?

Como se ve en el gráfico que contiene las curvas de proyección, es posible que si en la fecha de aceptación se muestra únicamente el código liberado por QC, ya se le tenga que informar al cliente sobre el posible atraso y disminución de las funcionalidades que se van entregar en la siguiente versión del producto.

Como no se puede modificar la fecha la siguiente versión probablemente no contenga todas las funcionalidades planificadas.

#### 2.3.2.2.- Evolución de la prueba

#### ¿Qué se lee?



A través del indicador se puede ver que en comparación con la cantidad de defectos abiertos, el ritmo del equipo de QC es muy lento. Habiendo 157 errores por cerrar, abren 10 por día y cierran 8 por día, las cuales son cifras muy pequeñas (un orden de magnitud menor) considerando que se encuentra muy próximo el cierre de la versión.

También se puede ver que dentro de 157 errores por cerrar solo 25 están corregidos, por lo tanto el equipo de desarrollo tampoco está corrigiendo los defectos abiertos por lo que el equipo de QC tampoco podría mejorar mucho la situación.

El equipo de QC está estancado hace aproximadamente dos meses, no abre, cierra ni suspende una cantidad considerable de bugs.

#### ¿Cuáles son las posibles causas?

El equipo de desarrollo podría no estar ocupándose de corregir errores para poder cumplir en fecha la liberación de código.

El equipo de QC podría o ser muy chico o ineficiente, porque tienen trabajo atrasado (25 errores corregidos sin procesar).

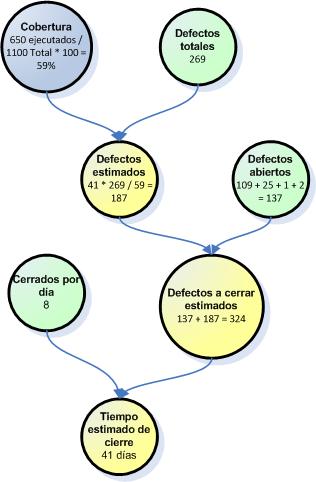
#### ¿Cuáles son los posibles impactos?

Dada la cercanía de la fecha de cierre de la versión es probable que se tenga que suspender una cantidad considerable de bugs.

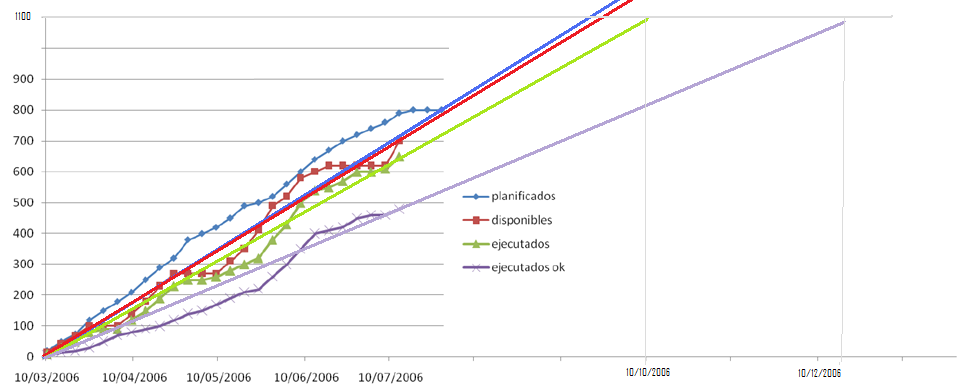
Modificaciones en el equipo de QC, dado el bajo rendimiento actual.

#### 2.3.2.3.- Cobertura de la prueba

#### ¿Qué se lee?



Si se considera que un mes tiene 20 días de trabajo, faltarían dos meses para cerrar todos los errores, por lo tanto al ritmo al que va el equipo de QC podría hacer todo el trabajo necesario para llegar a la fecha de cierre de la versión (en dos meses).



Pero con respecto a los casos de prueba diseñados se ve que el equipo de QC no da abasto, ni siquiera podría tener el total de casos de prueba diseñados para la fecha de cierre de versión.

Por lo tanto deberían ocupar parte del tiempo en acelerar esta tarea y se atrasarían en el cierre de bugs, y el tiempo estimado para cerrarlos indicaba que tendrían que estar full time cerrando bugs.

Sin embargo se ve que el mayor problema es que el ritmo de ejecución ok de casos de prueba es muy bajo.

#### ¿Cuáles son las posibles causas?

El equipo de desarrollo genera muchos errores, y por más de que el indicador de FC aparente un buen rendimiento de este sector, es posible que estemos ante el síndrome del 90%.

El equipo de QC es muy pequeño o ineficiente.

#### ¿Cuáles son los posibles impactos?

Entrega de una versión con muchos errores suspendidos o menor cantidad de funcionalidades.

Que el retraso en corrección de errores por parte del equipo de desarrollo oculte el retraso del equipo de testing que tampoco da abasto.

### 2.3.3.- Acciones correctivas

* Tomar medidas para que los desarrolladores liberen código más confiable, con menos bugs y problemas para revisar.
* Tomar medidas para que el equipo de QC pueda concebir de manera más rápida los diseños de las pruebas y las estrategias de ejecución de pruebas.
* Tomar medidas para incrementar el nivel de calidad del equipo de desarrollo y del equipo de QC, ya sea con cursos, capacitación o ayuda de especialistas en la materia.
* Analizar si los miembros del equipo de QC son pocos en relación con los miembros del equipo de desarrollo; si es así, considerar tomar más gente para QC.