Índice

[1.- Enunciado 2](#_Toc230445171)

[1.1.- Escenario 2](#_Toc230445172)

[1.1.1.- Indicador de funcionalidad completa 3](#_Toc230445173)

[1.1.2.- Indicador de Evolución de la prueba 4](#_Toc230445174)

[1.1.3.- Indicador de Cobertura de la Prueba 5](#_Toc230445175)

[1.2.- Objetivo 6](#_Toc230445176)

[2.- Resolución 7](#_Toc230445177)

[2.1.- Hipótesis y supuestos 7](#_Toc230445178)

[2.2.- Análisis de los indicadores 8](#_Toc230445179)

[2.2.1.- Funcionalidad Completa 8](#_Toc230445180)

[2.2.2.- Evolución de la prueba 9](#_Toc230445181)

[2.2.3.- Cobertura de la prueba 10](#_Toc230445182)

[2.3.- Acciones correctivas 11](#_Toc230445183)

# 1.- Enunciado

## 1.1.- Escenario

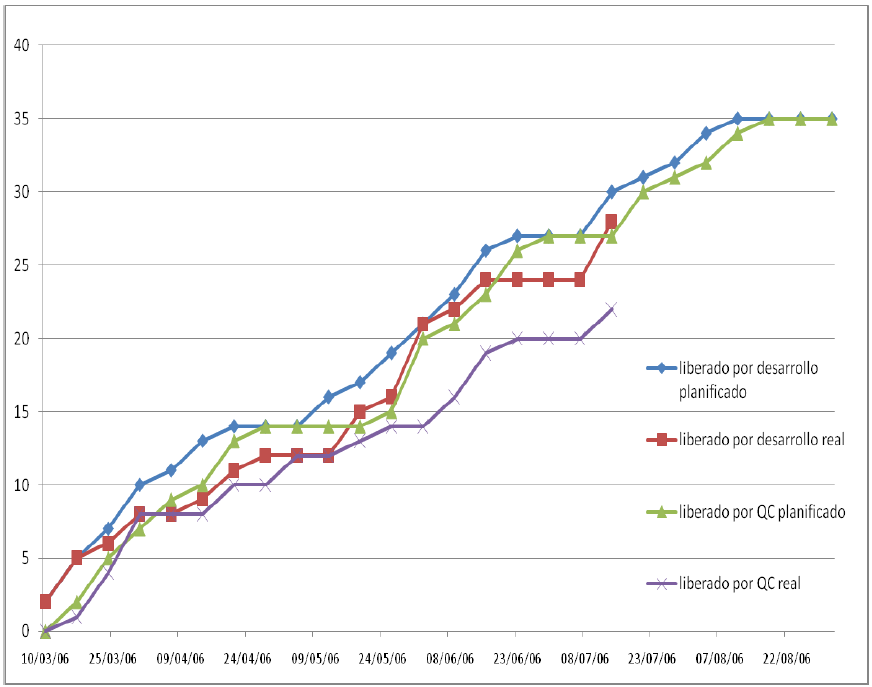
El mismo escenario de WBS para el proyecto “Sistema UARTN”.

La etapa de construcción de esta primer 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.

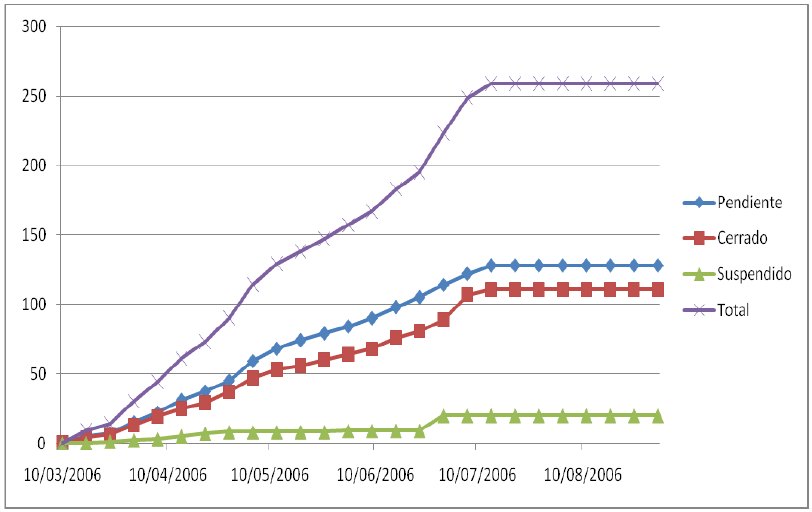
### 1.1.1.- Indicador de funcionalidad completa

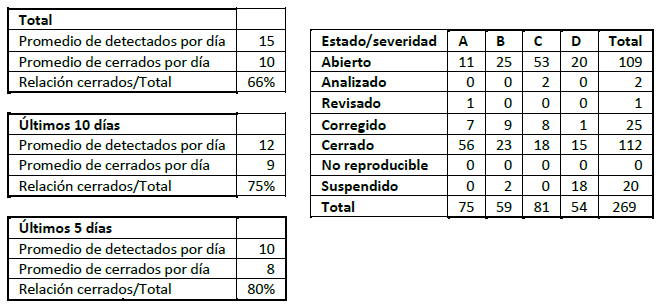


El indicador de FC muestra 4 curvas:

* **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.

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

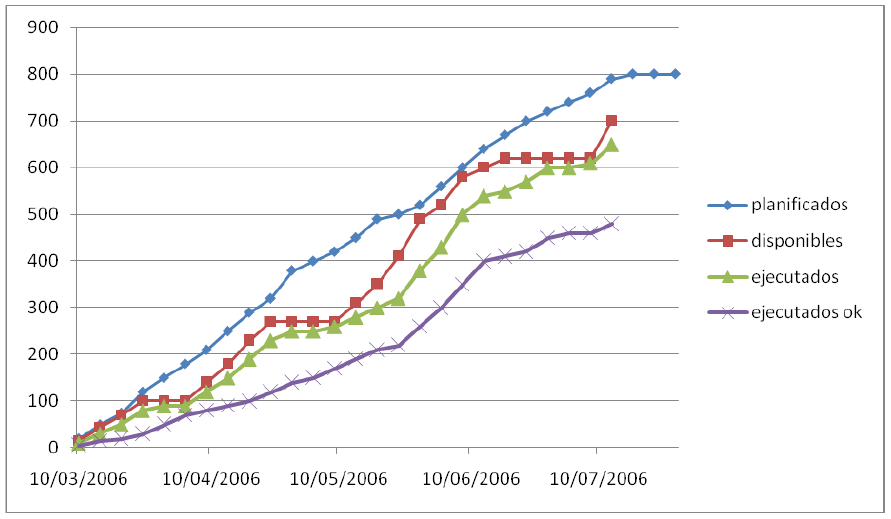


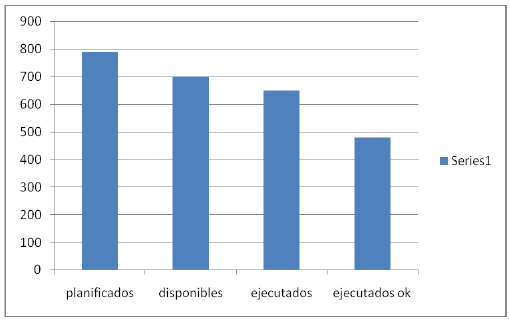


El indicador de Evolución de la Prueba muestra 4 curvas:

* **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.

### 1.1.3.- 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.

## 1.2.- 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.- Resolución

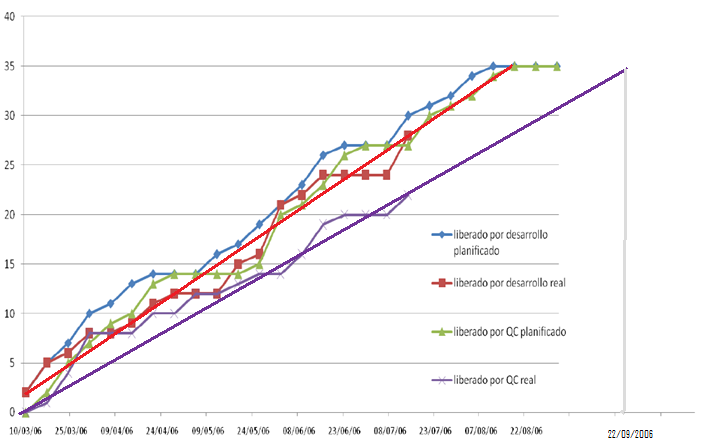
## 2.1.- Hipótesis y supuestos

* 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.

## 2.2.- Análisis de los indicadores

### 2.2.1.- 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.

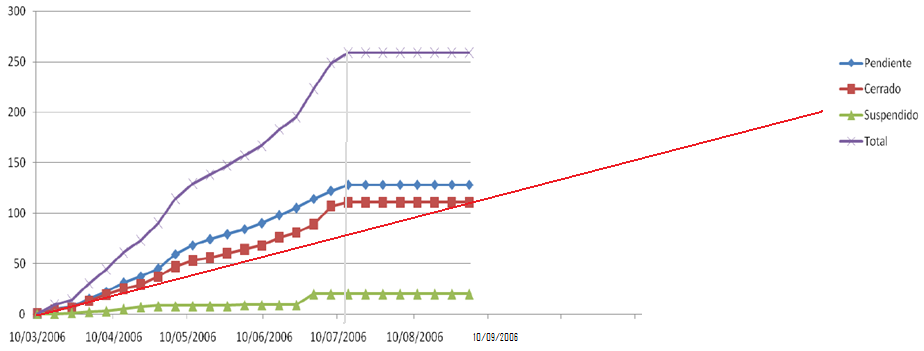
#### ¿Cuáles son los posibles impactos?

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.

### 2.2.2.- 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).

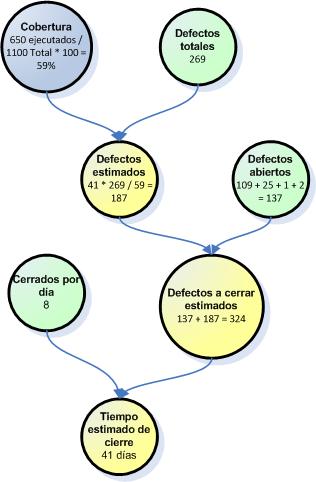
#### ¿Cuáles son los posibles impactos?

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

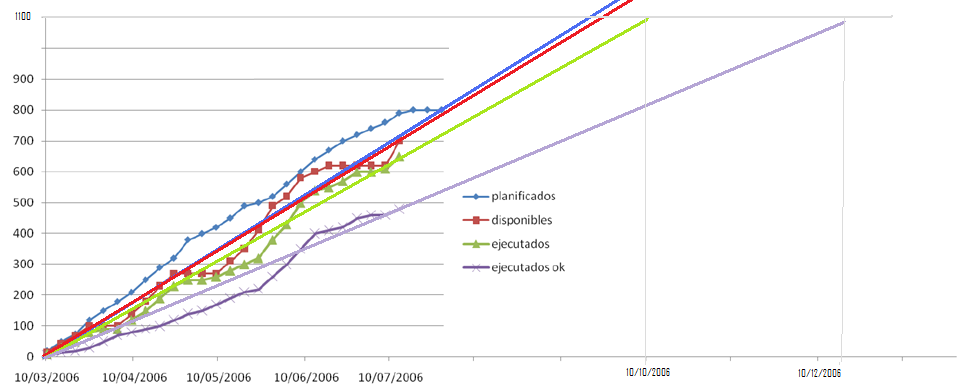
Modificaciones en el equipo de QC, dado el bajo rendimiento actual.

### 2.2.3.- 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.- Acciones correctivas