

### 5.13. GESTIÓN DINÁMICA DE UN PROYECTO

Existen muchas actividades, desde escribir un libro a construir un buque, que presentan unas características muy similares. Existe en primer lugar un objetivo y unas previsiones de lo que costará conseguir ese objetivo, seguido a continuación de una fase en la que se emplea una cantidad de trabajo y recursos, y por último, con mayor o menor éxito se logra el objetivo deseado.

No obstante, muchos proyectos acaban, en relación con los objetivos iniciales, durando más de lo previsto, costando más de lo esperado y ofreciendo una calidad mucho menor de lo deseado.

Vamos a ver a continuación un modelo para ayudar a comprender los aspectos relacionados con la consecución de un proyecto. Para hacer el proyecto fácil de explicar, vamos a trabajar en un proyecto genérico como diseñar un edificio, aunque el modelo es aplicable a otras muchas situaciones, como desarrollar software, diseñar un nuevo producto, etc.



Los modelos pueden utilizarse tanto para investigar proyectos de construcción, donde la disponibilidad del material es importante, hasta investigaciones científicas en las que la experimentación juega un papel esencial.

En el proceso de conceptualizar y crear el modelo vamos a seguir un procedimiento progresivo. Vamos a empezar con la estructura más simple del problema, para irlo mejorando en sucesivas versiones del modelo. Esta es una técnica útil ya que nos proporciona etapas en las que el modelo nos reproduce bien algunos aspectos de la realidad, a la vez que nos muestra aspectos donde debe ser mejorado. Iremos simulando en cada una de las etapas, y viendo el efecto de los cambios introducidos en relación con la etapa anterior.

En el proceso de creación del modelo utilizaremos el ordenador para que nos muestre con claridad los efectos de los cambios que nosotros estamos haciendo en la estructura del modelo. No obstante, aunque el ordenador es muy eficiente en este trabajo, es importante que nosotros sepamos interpretar las simulaciones que vayamos a realizar y para ello es conveniente que antes de cada simulación tengamos unas expectativas claras de lo que esperamos que ocurra, de forma que los resultados del modelo nos sirvan para confirmar o no nuestras expectativas.

## Modelo 1. Hacer un trabajo

La característica más fundamental de un proyecto es que hay un trabajo que hacer, y que este se va realizando progresivamente. Vamos a introducir esto en un modelo.

Total de tareas previstas



- (1) FINAL TIME = 24  
Units: Month
- (2) INITIAL TIME = 0  
Units: Month
- (3) SAVEPER =
- (4) Tareas finalizadas= trabajo  
Initial value: 0  
Units: m2
- (5) Tareas pendientes= -trabajo  
Initial Value: Total de tareas previstas
- (6) TIME STEP = 0.0625 Ver nota  
Units: Month
- (7) Total de tareas previstas= 1000  
Units: m2
- (8) trabajo= 100  
Units: m2/mes



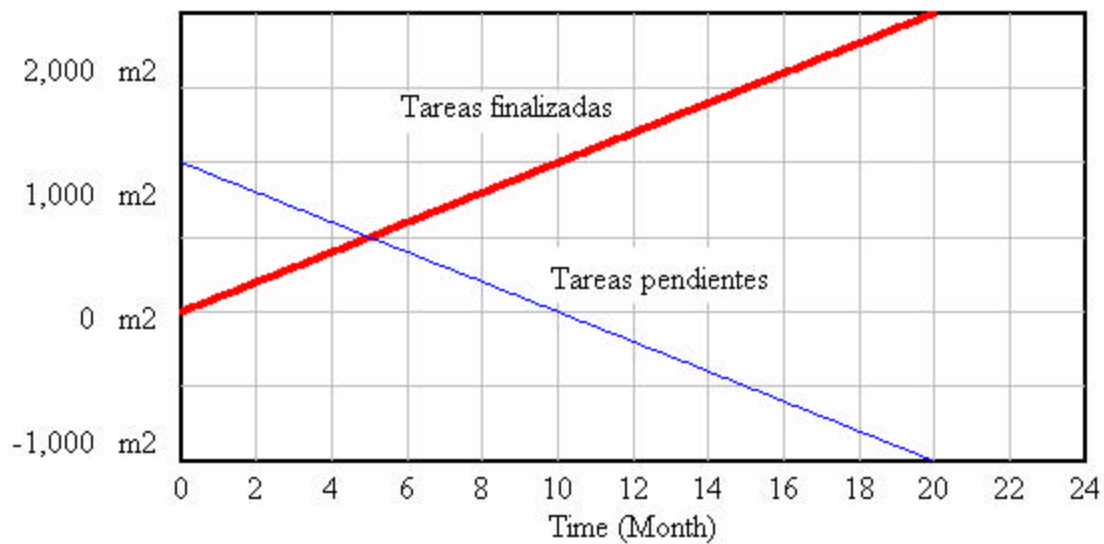
Nota:

(5) Usaremos la tecla de Choose Initial Variable para seleccionar “Total de tareas previstas”

(6) Si TIME STEP=1 haríamos 24 pasos en la simulación (Final Time – Initial Time). Queremos que se haga el cálculo cada 2 días ... 12 meses x (2/365) = 0.065 y escogemos la opción más próxima.

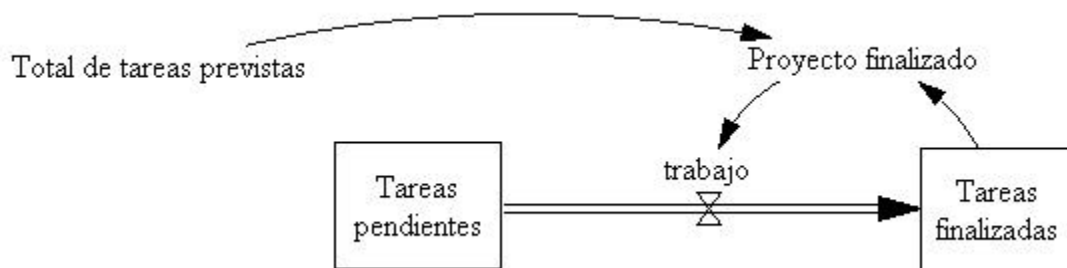
Nota 2:

(-) Si aparece un aviso del tipo “Tareas finalizadas is not used” es porque el software detecta que esta variable no influye en ninguna otra, y nos avisa ya que esto no es normal.



### Modelo 2. Parar al acabar.

Es necesario introducir en el modelo un mecanismo que nos detenga el proceso cuando el proyecto se acabe.



(01) FINAL TIME = 24

(02) INITIAL TIME = 0

(03) Proyecto finalizado=IF THEN ELSE(Tareas finalizadas>=Total de tareas previstas,1,0)

(04) SAVEPER =

(05) Tareas finalizadas= trabajo  
Initial value: 0  
Units: m2

(06) Tareas pendientes= -trabajo,  
Initial value: Total de tareas previstas  
Units: m2

(07) TIME STEP = 0.0625

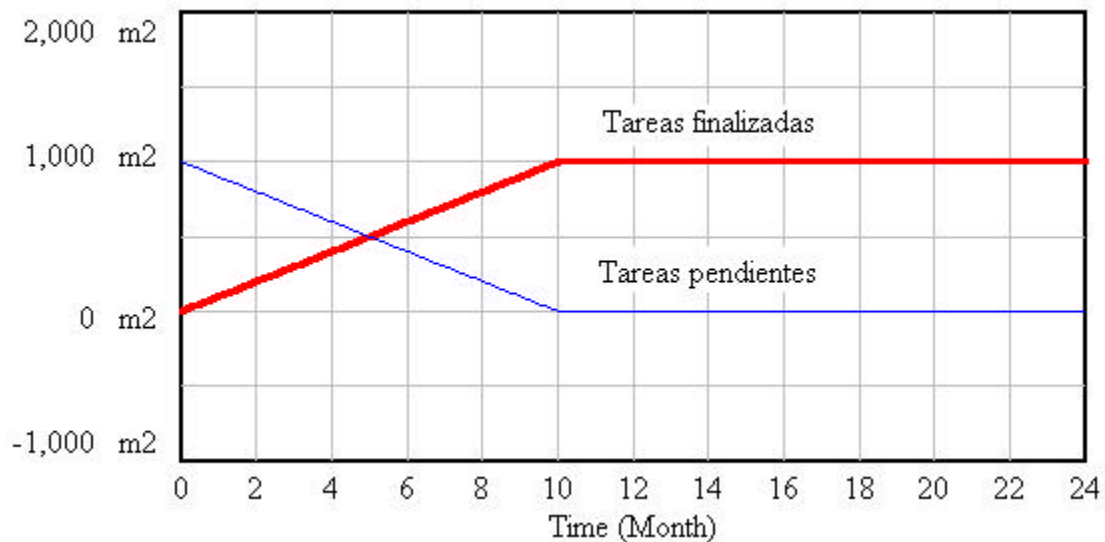
(08) Total de tareas previstas= 1000

Units: m2

Total de tareas previstas para el proyecto. Las vamos a considerar todas iguales, ya que el propósito del modelo es trabajar con parámetros no previsibles como son los errores.

(09) trabajo=IF THEN ELSE(Proyecto finalizado,0,100)

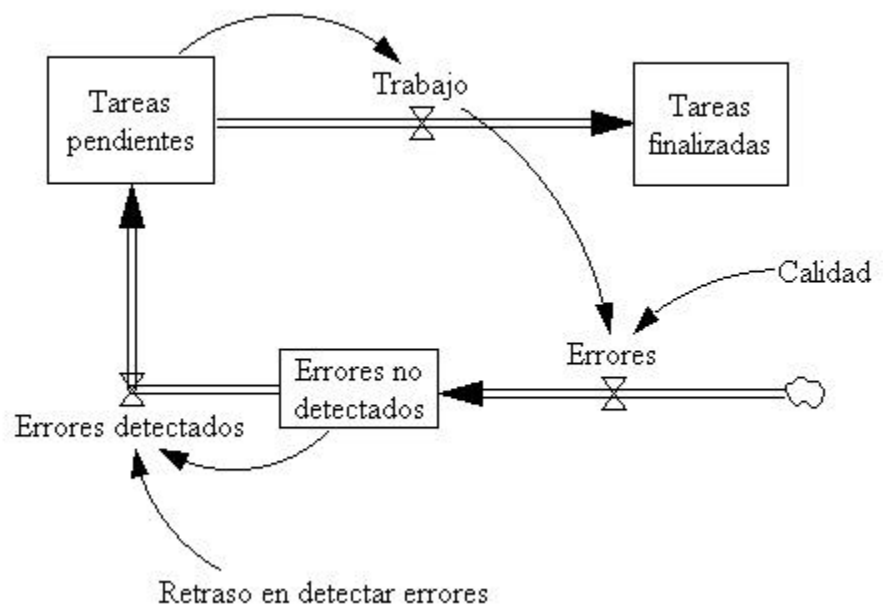
Units: m2/mes



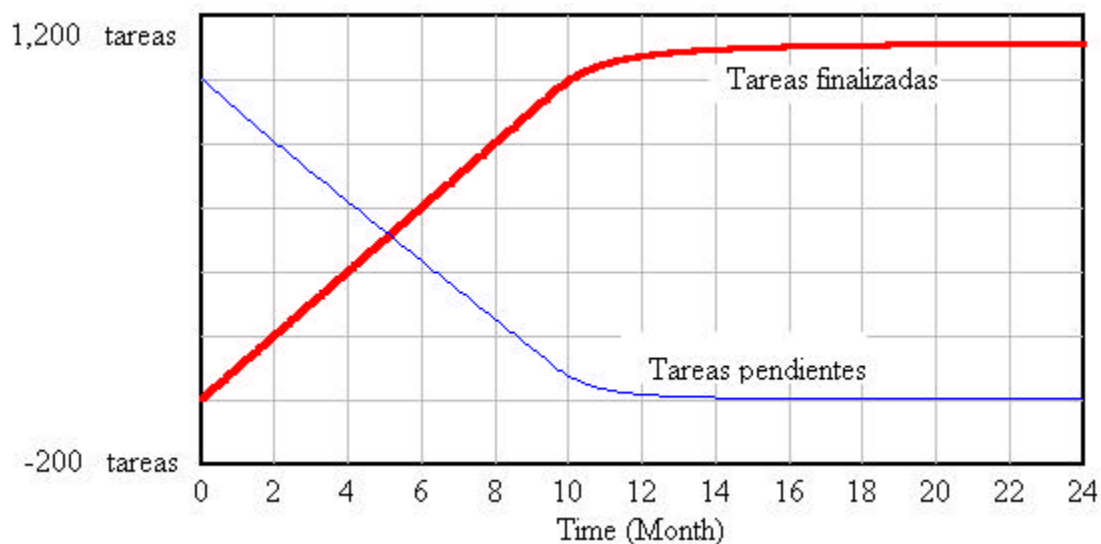
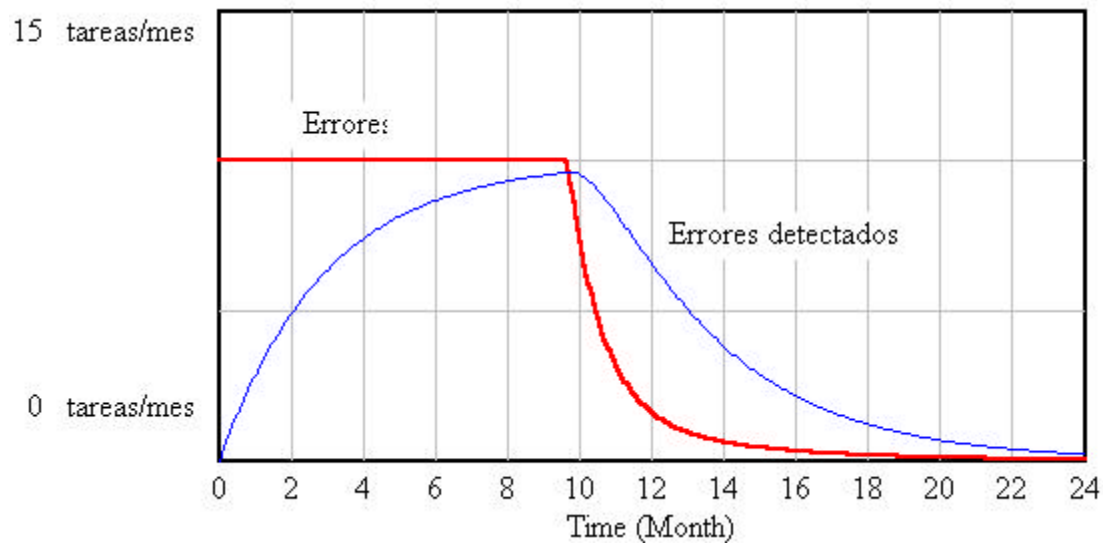
### Modelo 3. Incluir los errores

En todo proyecto se producen errores, trabajos que hay que volver a hacer, y estos deben ser incluidos en el modelo.

Total de tareas previstas



- (01)  $Calidad = 0.9$   
Calidad de los trabajos dados como finalizados
- (02)  $Errores = Trabajo * (1 - Calidad)$   
Units: tareas/mes  
Los errores serán función de la cantidad de trabajo hecho por el porcentaje de trabajo que no cumple las Normas.
- (03)  $Errores\ detectados = Errores\ no\ detectados / Retraso\ en\ detectar\ errores$   
Units: tareas/mes  
Tomaremos una fracción constante de los errores pendientes de detectar.
- (04)  $Errores\ no\ detectados = Errores - Errores\ detectados$   
Initial value: 0  
Units: tareas
- (05)  $FINAL\ TIME = 24$
- (06)  $INITIAL\ TIME = 0$
- (07)  $Retraso\ en\ detectar\ errores = 3$   
Units: mes
- (08)  $SAVEPER =$
- (09)  $Tareas\ finalizadas = Trabajo$   
Initial value: 0  
Units: tareas
- (10)  $Tareas\ pendientes = -Trabajo + Errores\ detectados$   
Initial value: Total de tareas previstas  
Units: tareas
- (11)  $TIME\ STEP = 0.0625$
- (12)  $Total\ de\ tareas\ previstas = 1000$   
Units: tareas  
Total de tareas previstas para el proyecto. Las vamos a considerar todas iguales, ya que el propósito del modelo es trabajar con parámetros no previsibles como son los errores en el proyecto.
- (13)  $Trabajo = MIN(100, Tareas\ pendientes)$   
Units: tareas/mes  
Como máximo se hacen 100 tareas al mes

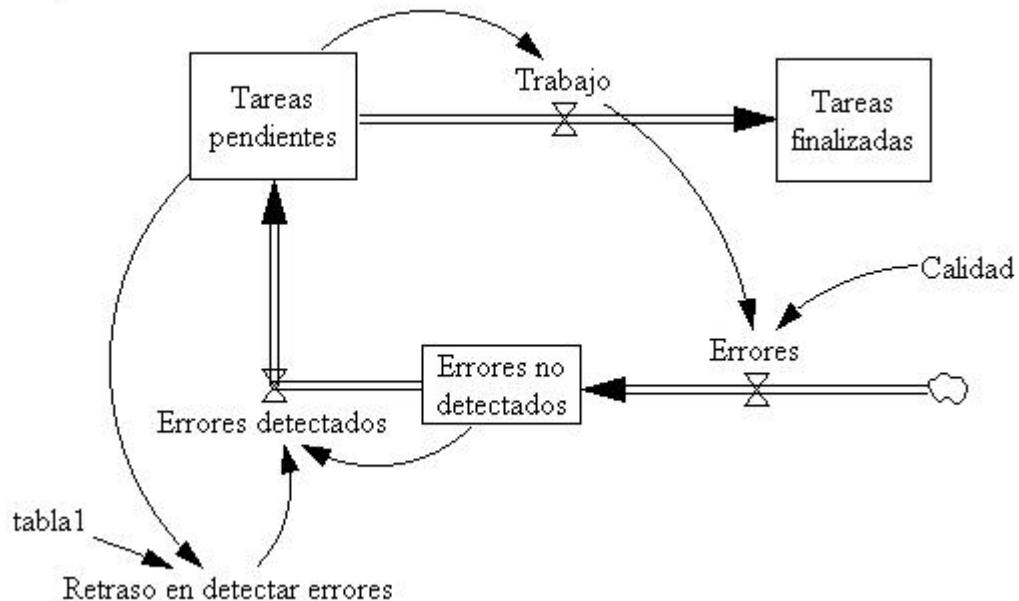


Del modelo 2 al modelo 3 hay un "giro", de forma que dejamos de utilizar la variable "proyecto finalizado" para controlar que no se hagan mas tareas de la previstas. La razón es que observamos que en realidad en muchos proyectos se hacen muchas mas tareas de las inicialmente previstas ya que hay que repetir aquellas tareas que se hicieron mal, o con errores. Por lo tanto esta variable no puede ser un limite para la cantidad de trabajo realizado.

#### Modelo 4. Descubrir los errores

Los errores que se producen en un proyecto no se descubren inmediatamente, sino cuando van a hacerse otros trabajos posteriores.

Total de tareas previstas



- (01)  $Calidad = 0.9$   
Calidad de los trabajos dados como finalizados
- (02)  $Errores = Trabajo * (1 - Calidad)$   
Units: tareas/mes  
Los errores serán función de la cantidad de trabajo hecho por el porcentaje de trabajo que no cumple las Normas.
- (03)  $Errores\ detectados = Errores\ no\ detectados / Retraso\ en\ detectar\ errores$   
Units: tareas/mes  
Tomaremos una fracción constante de los errores pendientes de detectar.
- (04)  $Errores\ no\ detectados = \frac{Errores - Errores\ detectados}{0}$   
Units: tareas
- (05)  $FINAL\ TIME = 24$
- (06)  $INITIAL\ TIME = 0$
- (07)  $Retraso\ en\ detectar\ errores = tabla1(Tareas\ pendientes)$   
Units: mes  
Se reconsidera la constante (3) que habíamos puesto en el modelo 3, y se hace que sea función de la cantidad de tareas pendientes que hay.

(08)  $SAVEPER =$

(09) tabla1

(0,0.5),(200,1),(400,3),(600,6),(800,9),(1000,12)

En esta tabla recogemos la idea lógica de que cuando se está al principio del proyecto, y queda mucho trabajo por hacer se tarda más tiempo en detectar los errores que cuando ya se está en las últimas fases del proyecto, donde cualquier error es detectado con más rapidez.

(10) Tareas finalizadas= Trabajo  
0

Units: tareas

(11) Tareas pendientes=  $\frac{\text{Trabajo} + \text{Errores detectados}}{\text{Total de tareas previstas}}$

Units: tareas

(12)  $TIME STEP = 0.0625$

(13) Total de tareas previstas= 1000

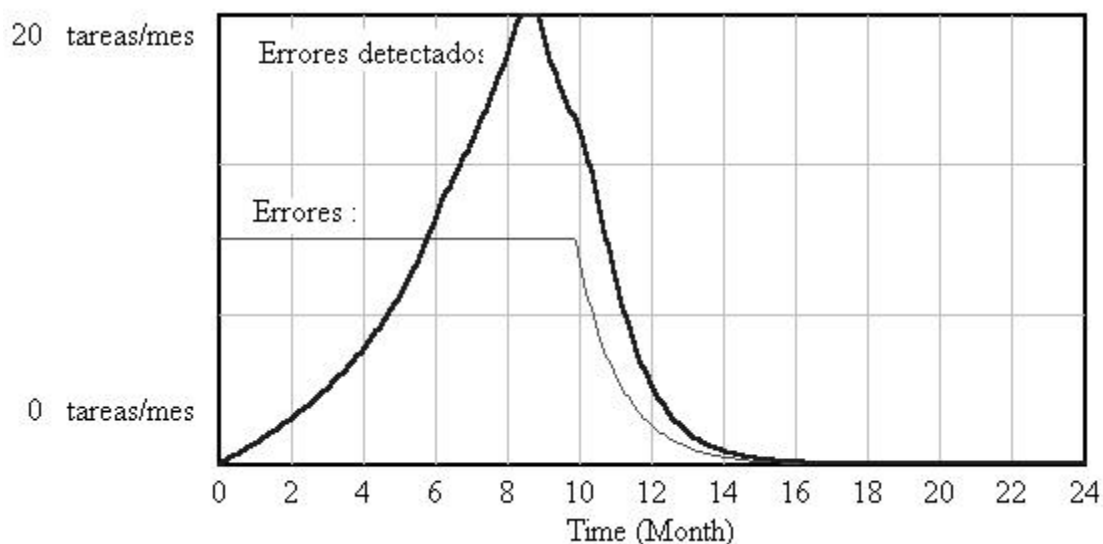
Units: tareas

Total de tareas previstas para el proyecto. Las vamos a considerar todas iguales, ya que el propósito del modelo es trabajar con parametros no previsibles como son los errores en el proyecto.

(14) Trabajo=MIN(100,Tareas pendientes)

Units: tareas/mes

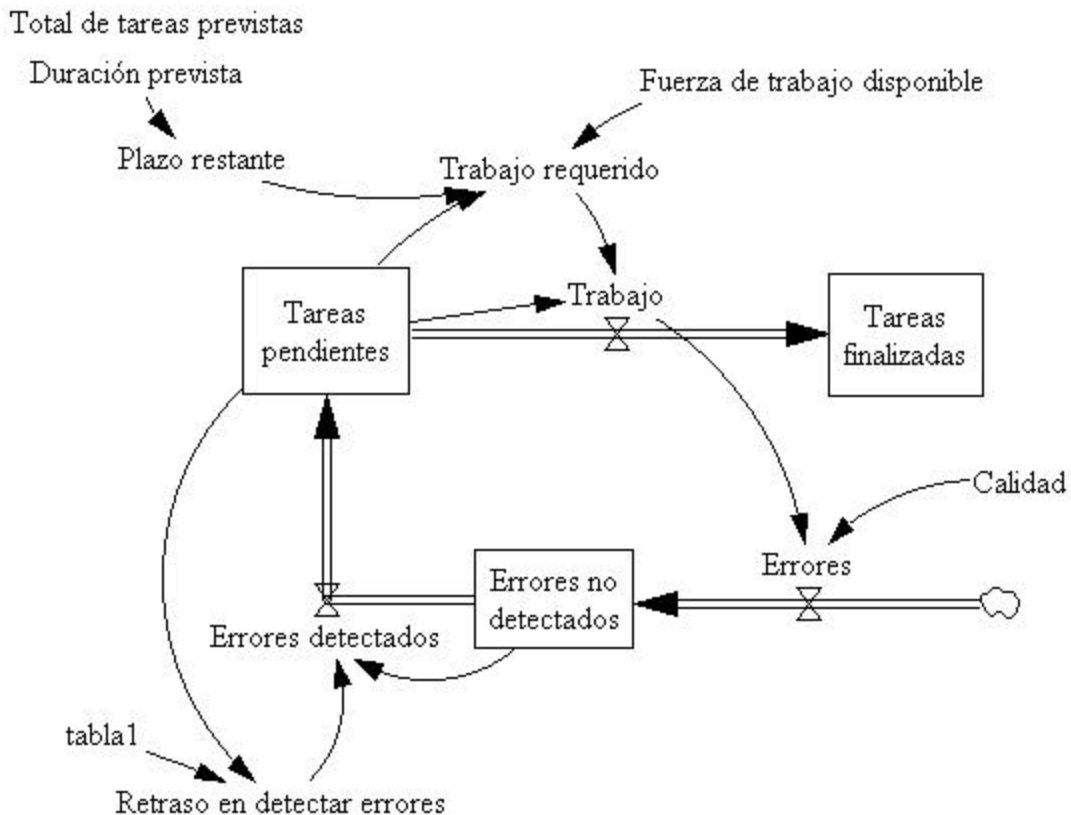
Como máximo se hacen 100 tareas al mes





### Modelo 5. La presión de la fecha de entrega.

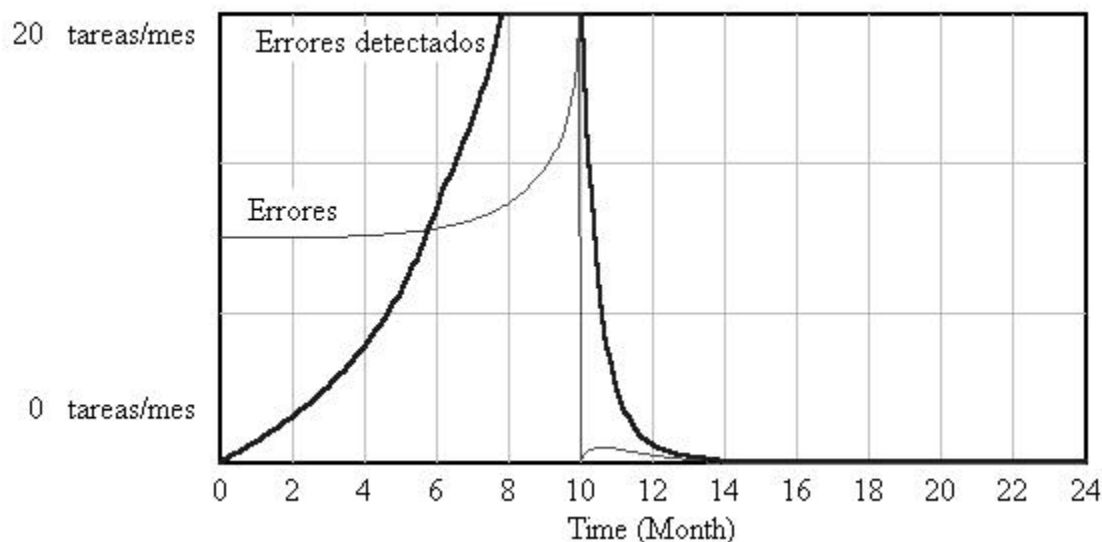
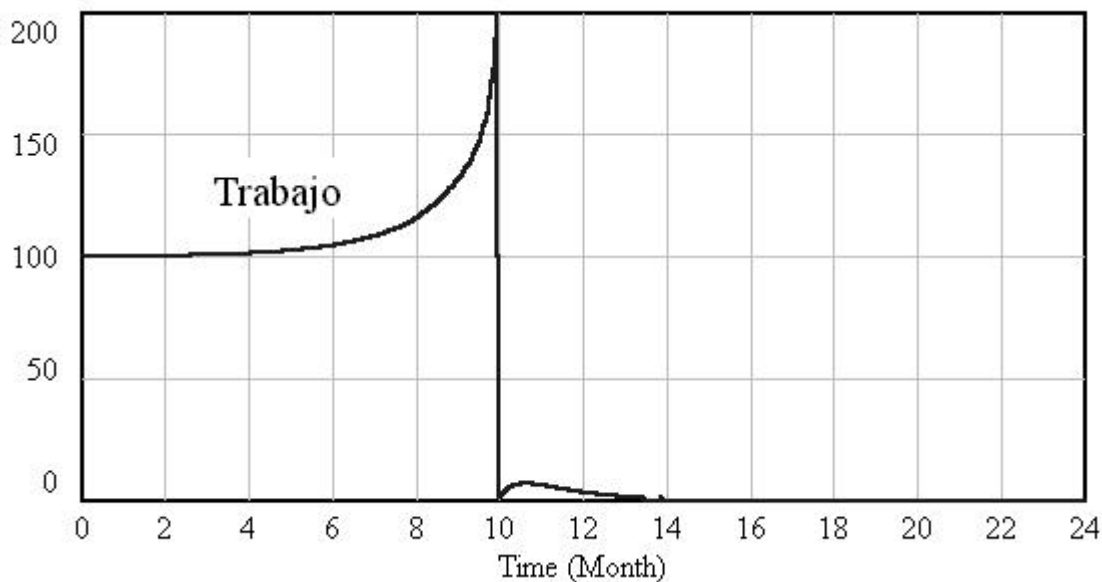
En todo proyecto existe una fecha de finalización y entrega, que va a modificar el ritmo de trabajo y la cantidad de los recursos empleados.



- (01)  $\text{Calidad} = 0.9$   
Calidad de los trabajos dados como finalizados
- (02)  $\text{Duración prevista} = 10$   
Units: meses  
Hay 1000 tareas a un ritmo previsto de 100 al mes
- (03)  $\text{Errores} = \text{Trabajo} \times (1 - \text{Calidad})$   
Units: tareas/mes  
Los errores serán función de la cantidad de trabajo hecho por el porcentaje de trabajo que no cumple las Normas.
- (04)  $\text{Errores detectados} = \frac{\text{Errores no detectados}}{\text{Retraso en detectar errores}}$   
Units: tareas/mes  
Tomaremos una fracción constante de los errores pendientes de detectar.

- (05) Errores no detectados= Errores-Errores detectados  
Initial value: 0  
Units: tareas
- (06) FINAL TIME = 24
- (07) Fuerza de trabajo disponible= 250  
Units: tareas/mes  
Corresponde a la capacidad de producción máxima
- (08) INITIAL TIME = 0
- (09) Plazo restante= MAX(Duración prevista-RAMP(1,0,100),0)  
Units: meses  
Es una forma sencilla de contar el plazo residual en cada momento, empezando por 10 meses hasta llegar a 0.
- (10) Retraso en detectar errores= tabla1(Tareas pendientes)  
Units: mes  
Se reconsidera la constante (3) que habíamos puesto en el modelo 3, y se hace que sea función de la cantidad de tareas pendientes que hay.
- (11) SAVEPER =
- (12) tabla1  
(0,0.5),(200,1),(400,3),(600,6),(800,9),(1000,12)  
En esta tabla recogemos la idea lógica de que cuando se está al principio del proyecto, y queda mucho trabajo por hacer se tarda más tiempo en detectar los errores que cuando ya se está en las últimas fases del proyecto, donde cualquier error es detectado con más rapidez.
- (13) Tareas finalizadas= Trabajo  
Initial value: 0  
Units: tareas
- (14) Tareas pendientes= -Trabajo+Errores detectados  
Total de tareas previstas  
Units: tareas
- (15) TIME STEP = 0.0625
- (16) Total de tareas previstas= 1000  
Units: tareas  
Total de tareas previstas para el proyecto. Las vamos a considerar todas iguales, ya que el propósito del modelo es trabajar con parametros no previsibles como son los errores en el proyecto.

- (17) Trabajo=IF THEN ELSE(Tareas pendientes<1,0,Trabajo requerido)  
 Units: tareas/mes  
 Si las tareas pendientes son menor que 1 (0) no se hace ninguna, en caso contrario se hace el Trabajo requerido
- (18) Trabajo requerido=MIN(Fuerza de trabajo disponible,XIDZ(Tareas pendientes,Plazo restante,Tareas pendientes))  
 Units: tareas/mes  
 La cantidad de trabajo requerido (llamado) será como máximo la Fuerza de trabajo disponible. Se calcula como el cociente entre las Tareas pendientes (tareas) y el Plazo restante (meses). La función XIDZ nos ofrece este cociente excepto cuando el Plazo restante es cero, en cuyo supuesto se toma 1 como Plazo.



En esta última versión del modelo ya podemos apreciar con claridad los fenómenos que deseamos estudiar en los proyectos como son la acumulación de errores en las fases finales del proyecto, o la distribución de las tareas del proyecto, que se van incrementando progresivamente a pesar de nuestra programación de cargas de trabajo, y que se prolongan incluso más allá de la fecha de entrega del proyecto.