



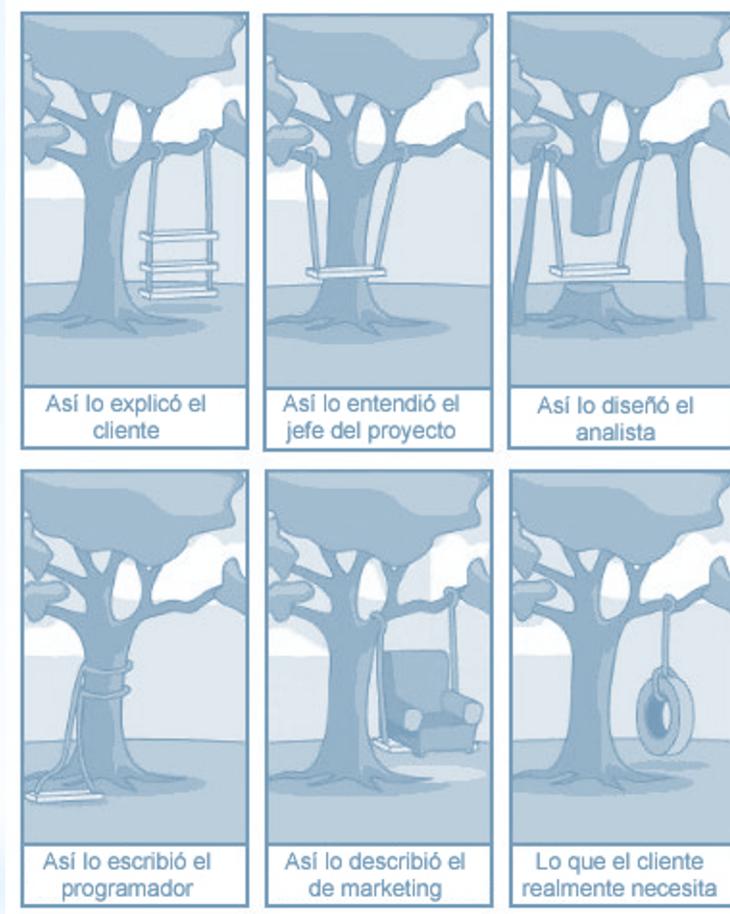
Universidad Nacional Autónoma de México
Facultad de Ingeniería
Computación para Ingenieros
Tema 7
DESARROLLO DE UN PROYECTO BÁSICO DE SOFTWARE

7. Desarrollo de un proyecto básico de software

Objetivo: Elaborar un proyecto básico de software aplicando la teoría del diseño de programas y herramientas de desarrollo.

7. Desarrollo de un proyecto básico de software

- 7.1 Fases en el desarrollo de proyectos de software.
- 7.2 Herramientas de desarrollo en entornos visuales.
- 7.3 Elaboración de un proyecto básico de software.



7.1 Fases en el desarrollo de proyectos de software.

7.1 Fases en el desarrollo de proyectos de software.

Un proyecto de software se define como el proceso de administración para crear un sistema o software a partir de un conjunto de especificaciones o requerimientos.

Un proyecto de software inicia con las especificaciones del sistema que necesita el usuario.

Ya que se poseen los requerimientos se tiene que realizar el estimado de tiempo y personal necesario para cumplir con el objetivo en el tiempo estimado y con las restricciones impuestas por el usuario (presupuesto, equipo, espacio, etc.).

Un administración eficiente de un proyecto de software debe poseer las siguientes características:

- **Objetivos claros y definidos.**
- **Actividades planificadas y controladas.**
- **Menor número de recursos.**
- **Mínimo de tiempo.**
- **Fecha de inicio y fin calendarizados.**
- **Obtener un resultado único.**

Metodología de gestión de proyectos de software

Una metodología de gestión de proyectos cubre todos los aspectos que un líder de proyectos necesita administrar: desarrollo de software, selección de paquetes, mudanza de su departamento de proyectos, etc.

La Guía del PMBOK® (Project Management Body of Knowledge) cubre nueve áreas de la gestión de proyectos:

- Gestión del Costo
- Gestión del Riesgo
- Gestión del Alcance
- Gestión de los Recursos
- Gestión de las Comunicaciones
- Gestión de la Calidad
- Gestión del Tiempo
- Gestión de las Adquisiciones
- Gestión de la Integración

Los alcances de una metodología están dados por los siguientes elementos:

Descomposición	División del proyecto total en diferentes fases (divide y vencerás).
Panorama	Propósito, objetivos, entregables, así como los periodos de tiempo/entrega para cada fase.
Actividades	Identificar las tareas principales.
Entradas y salidas	Identificar las entradas o pre-requisitos para cada actividad así como sus salidas o entregables.
Instrucciones	Definir los pasos a seguir en cada actividad.

Participantes	Definir el equipo de trabajo por cada actividad.
Calidad	Definir los parámetros de calidad de cada fase.
Tiempos	Definir los tiempos de entrega para cada actividad.
Gobernabilidad	Establecer las autoridades aplicables para cada parte del equipo.
Materiales de soporte	Referencias, apoyos, guías, etc., que permitan realizar las actividades dadas.

Fases de la ingeniería de software

La ingeniería de software se define como el uso y establecimiento de principios de ingeniería sólidos, a fin de obtener un software que sea económicamente fiable y funcione eficientemente.

La ingeniería de software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:

- **Planeación y estimación del proyecto.**
- **Análisis de requerimientos del sistema y software**
- **Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico**
- **Codificación**
- **Pruebas y mantenimiento (validación y verificación).**

Los métodos para llevar a cabo una ingeniería de software de calidad a menudo se presentan en un lenguaje orientado especial o en una notación gráfica e introducen un conjunto de criterios para la calidad de software.

Ciclo de vida del software

La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida de un software como:

Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.

El ciclo de vida del software está constituido por las fases de análisis, diseño, implementación e instalación.

La calidad de software que se genera se mide en dos grandes rubros:

- **Factores de calidad externos:** Correcto, sólido o robusto, confiable, extensible, reutilizable, compatible, eficiente, portable, fácil de usar.
- **Factores de calidad internos:** Modular y legible.



Análisis y especificaciones de requisitos

Es el proceso para averiguar qué es lo que requiere el cliente del sistema de software (análisis de requisitos). Esta etapa permite definir los requisitos de forma clara y concisa (especificación de requisitos).

El equipo desarrollador de software se enfrenta al principio de un proyecto con un documento escrito por el cliente, en el cual expresa, en términos de la aplicación, qué se requiere del sistema. Este documento se conoce como declaración de requisitos.

La especificación del sistema debe contener una declaración de las funciones del sistema así como las restricciones con las cuales tendrá que trabajar el equipo desarrollador.

También debe contener información adicional como los detalles del hardware que se usará y la capacitación que tendrá que proporcionar el productor, así como una especificación de las herramientas de software especiales que se usarán en el proyecto.

Diseño del sistema

Se refiere al manejo de distintas técnicas y principios con el propósito de definir a detalle un producto para poder realizarlo físicamente (*cómo*).

El objetivo del diseño del sistema es producir una arquitectura que satisfaga las funciones de la especificación del sistema y al mismo tiempo respete las restricciones del documento.

En el diseño del sistema, el diseñador tiene que especificar el procesamiento que debe ocurrir en un componente individual y definir la interfaz entre los componentes.

El diseño del sistema debe realizar todas las funciones que se detallan en la especificación del sistema y, al mismo tiempo, satisfacer todas las restricciones.

El diseño representa el más alto nivel de abstracción y se puede seguir hasta requisitos más específicos funcionales, de datos o de comportamiento, es decir, permite generar una representación técnica del software a desarrollar.

La notación que se usa para describir el procesamiento que ocurre en cada unidad de programa se conoce como lenguaje de diseño de programas. Este lenguaje consiste en recursos junto con descripciones del procesamiento en lenguaje natural.

El diseño debe ser una guía entendible tanto para los desarrolladores del código, como para los que se encargan de realizar pruebas, así como para los encargados del mantenimiento del software.

Por lo tanto, debe proporcionar una idea completa de la funcionalidad y comportamiento del software desde el punto de vista de la implementación.

Implementación

Una vez realizada la etapa de diseño de la solución es posible automatizarla, es decir, codificar el diseño en algún lenguaje de programación.

La codificación representa la sistematización de la solución para hacer el proceso más eficiente y fácil para el usuario final.

Este proceso es el menos difícil pues sólo comprende la traducción de las unidades de programa, expresadas en lenguaje de diseño de programas, a código de programa.

Esto casi siempre implica una traducción directa de los enunciados del lenguaje de diseño del programa a enunciados en lenguaje de programación.

Pruebas y mantenimiento

La última etapa del ciclo del software está definida por las pruebas del software generado y el mantenimiento del mismo.

En términos de ingeniería de software un sistema es correcto cuando refleja por completo los requisitos del usuario que se detallaron en la especificación del sistema.

El proceso de revisar que un sistema sea correcto se describe mediante el término colectivo de validación y verificación: dos actividades separadas que aseguran que el sistema que se entrega cumple con los requisitos del usuario.

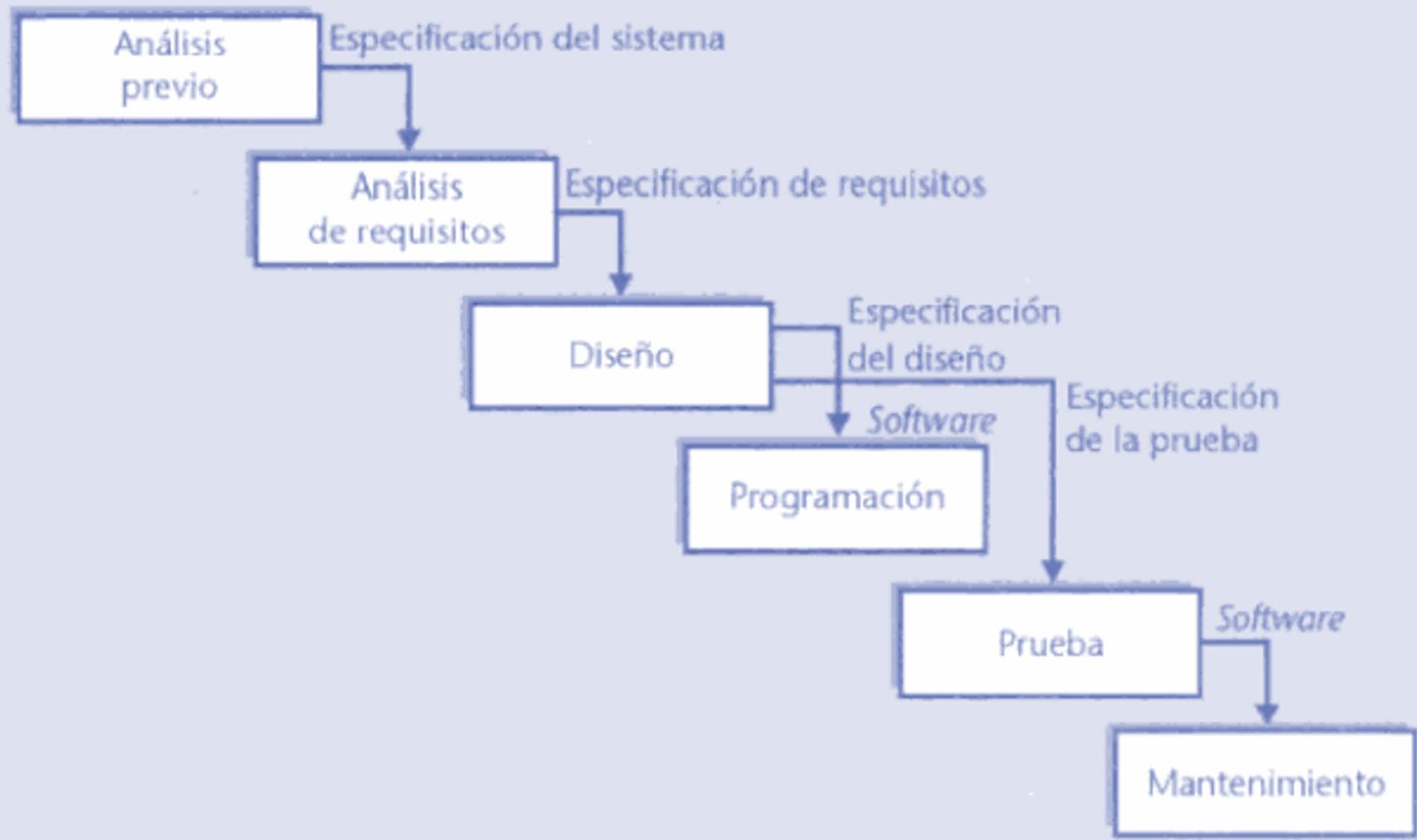
La verificación es el proceso de revisar que la salida de una fase corresponda con la entrada a esa fase. En la fase de diseño del sistema, esto consiste en asegurar que el diseño del sistema sea un reflejo correcto de la especificación de sistema.

La validación es el proceso de asegurar que el sistema en desarrollo cumpla con los requisitos del usuario. Un ejemplo de validación lo representa las pruebas de aceptación: el proceso de ejecución de un sistema con datos de prueba para revisar que cumpla con los requisitos de los usuarios.

El software es intrínsecamente flexible y, por ende, está en constante cambio.

Así mismo, los requisitos o necesidades del usuario cambian según cambian las circunstancias del negocio y, por tanto, el software también desarrollarse y cambiar.

Esto da lugar a la necesidad de mantenimiento del software y se refiere al hecho de corregir errores, generar mejoras o desarrollar nuevas funcionalidades.





How the customer explained it



How the Project Leader understood it



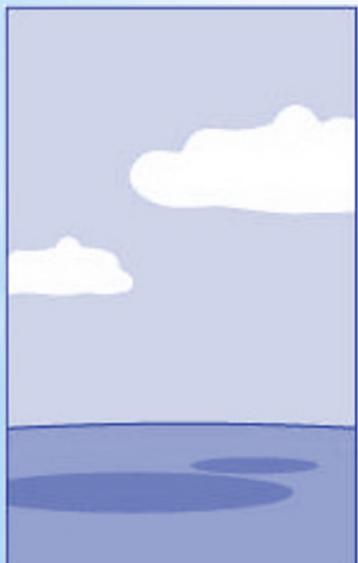
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed



7.2 Herramientas de desarrollo en entornos visuales.

7.2 Herramientas de desarrollo en entornos visuales.

La administración de proyectos es la disciplina para gestionar proyectos exitosamente, la cual puede y debe aplicarse durante el ciclo de vida de cualquier proyecto.

Una metodología de administración de proyectos es distinto a una metodología de desarrollo de software:

Metodología de administración de proyectos	Metodología de desarrollo de aplicaciones
Divide las fases del proyecto y el plan a seguir en cada una.	Establece las fases del desarrollo y las actividades que involucra
Define roles y responsabilidades de cada integrante del equipo.	Define roles y responsabilidades que corresponden a cada fase.
Administra el presupuesto del proyecto.	Define las medidas a emplearse para contabilizar el desarrollo.

El proceso de administración de proyectos recibe como entradas o es afectado por:

- Necesidades y requerimientos del proyecto a desarrollar (alcance).
- Límites establecidos en tiempo, costo, calidad, desempeño requerido, aspectos legales, etc.
- Mecanismos para lograrlo el proyecto: personas, técnicas, herramientas, equipo y organización.

Fases de la administración de Proyectos

Es posible distinguir 5 componentes de un proyecto (4 etapas más el control) en el desarrollo de un proyecto (Project management):

- 1.- Iniciación de proyecto.
- 2.- Planificación de proyecto.
- 3.- Producción de proyecto o ejecución.
- 4.- Supervisión y control del Proyecto.
- 5.- Finalización de proyecto o cierre.

IDE para la administración de un proyecto

Empezar a desarrollar un proyecto es difícil, por lo que es recomendable el uso de herramientas informáticas que faciliten este trabajo. Los programas de administración de proyectos ofrecen una ayuda importante para realizar un buen seguimiento del proyecto y que éste se cumpla en tiempo y forma.

Tras la organización de ideas se puede controlar las actividades en un programa de gestión que permita un seguimiento del proyecto mientras se produce su desarrollo.

Existe muchos y muy diversos IDE's de administración de proyectos, a continuación se presentan algunos.

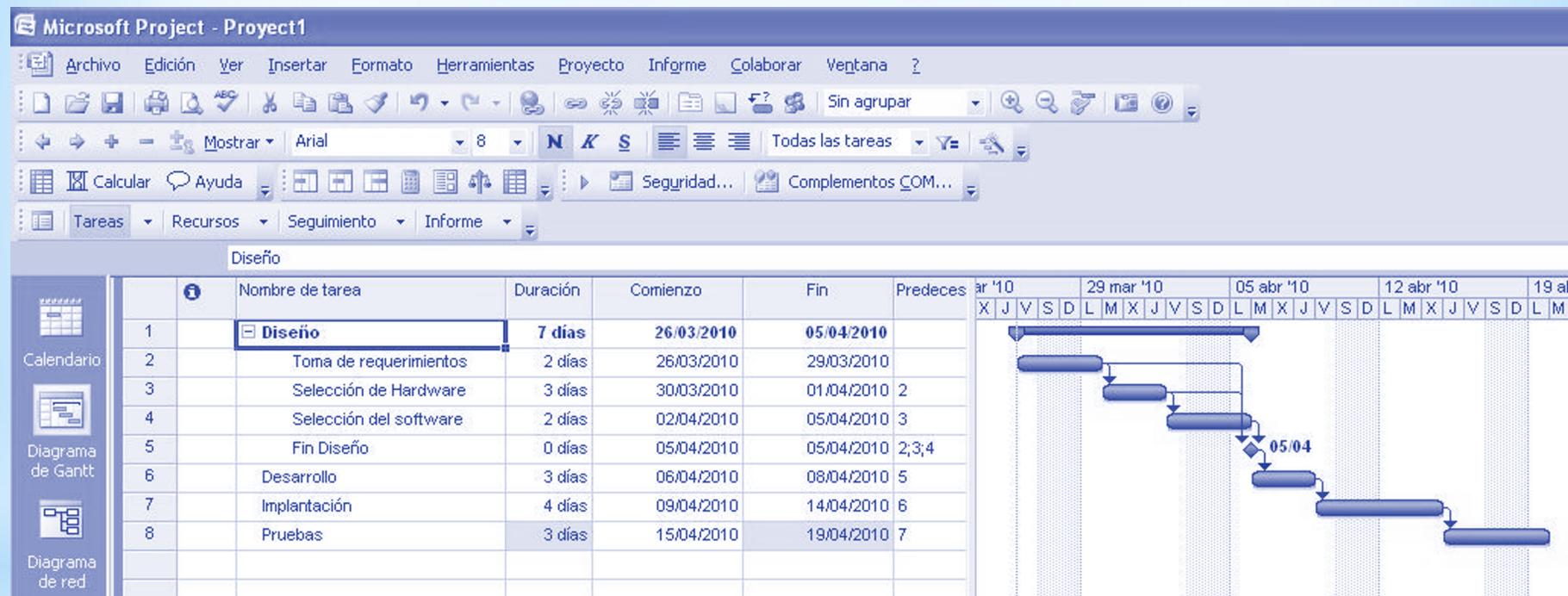
Microsoft Project

Permite la administración de tareas, recursos y el calendario del proyecto. Puede distribuir para cada tarea el personal y el material necesario y adecuarlo a la misma en función de la previsión que se realice.

Si una tarea se amplía o reduce posteriormente, se podrá, a su vez, disminuir o aumentar las necesidades de recursos.

Una vez que describas todas las actividades programadas se puede realizar el seguimiento y hacer modificaciones cuando sea necesario.

Microsoft Project

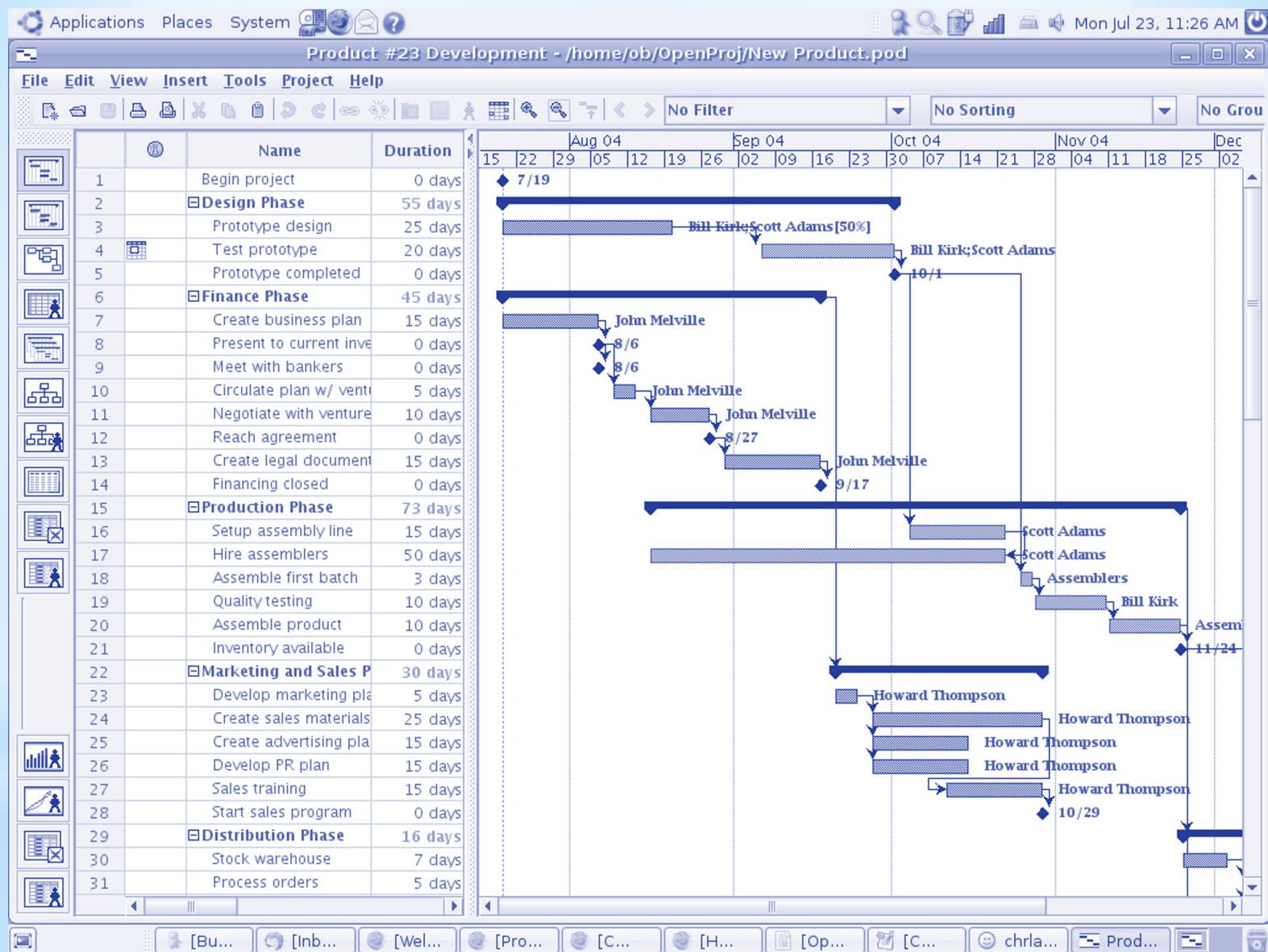


OpenProj

Programa de código libre, puede trabajar con archivos de Microsoft Project y permite gestionar el calendario de trabajo, las tareas, los recursos y la línea de base de forma similar al producto de Microsoft.

Tiene la herramienta projects-on-demand que permite el control de varios proyectos de forma simultánea y a través de Internet, aunque en esta modalidad habrá que abonar una cuota mensual.

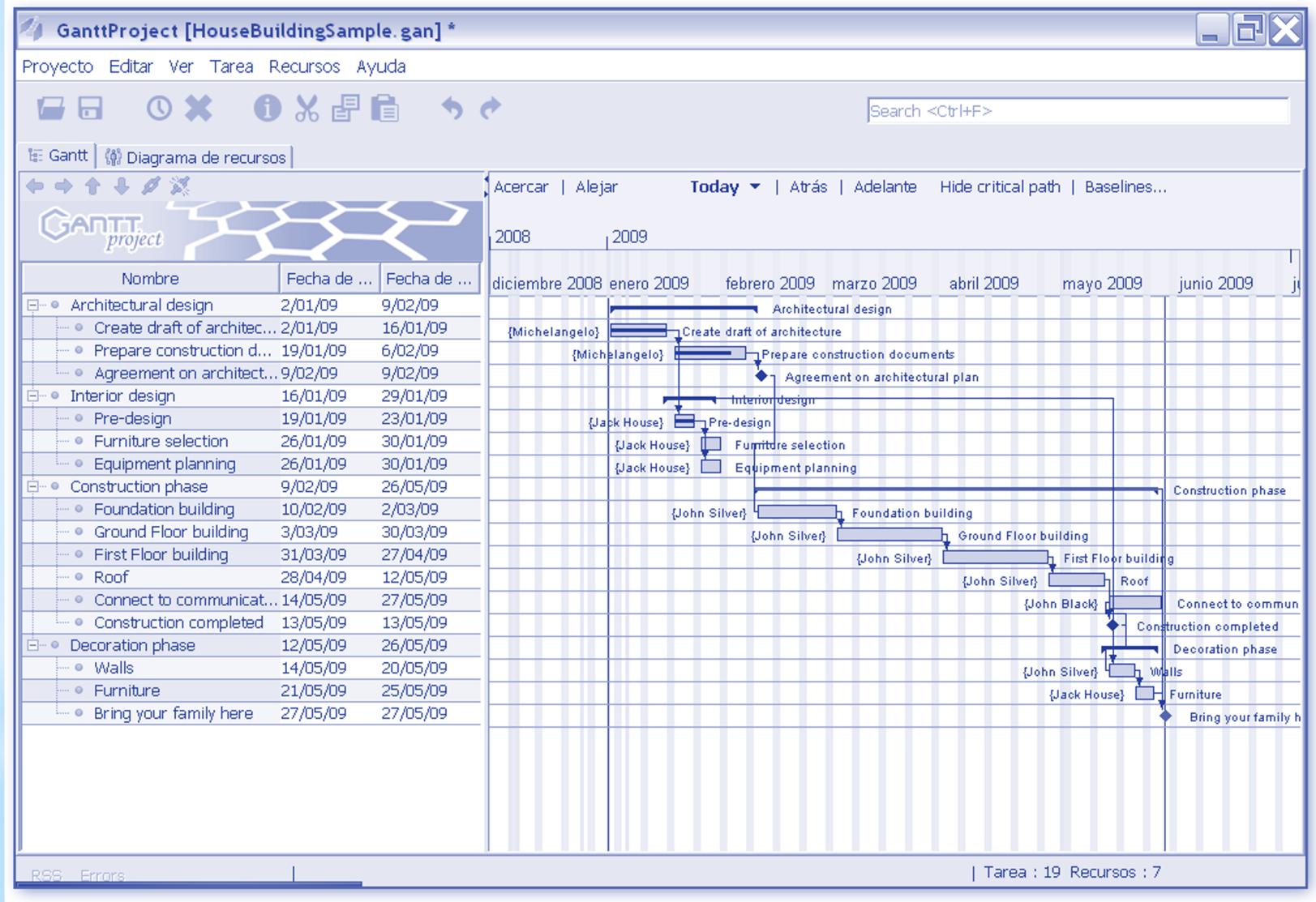
Open Proj



GanttProject

Gestor de proyectos gratuito de código libre que facilita el desarrollo y control del proyecto. El programa está disponible para sistemas operativos Windows, Mac y Linux. Para poder utilizarlo es necesario tener instalado Java en el equipo.

GranttProject



ClockingIt

Es un gestor de proyectos que se usa desde Internet, mediante un registro en su página web se puede acceder a las múltiples opciones que ofrece, como crear varios usuarios para la administración, la creación de clientes y de tareas.

Clocking IT

ClockingIT

Erleend Simonsen / ClockingIT
22:29 04/01/2009 (2 Online)
Timer Window

Overview Browse Timeline Schedule Reports W&D Chat Forums Files Projects New Task ↻

Users Clients Preferences Log Out

Task Worked Estimate Due Date

#121 Blog	0m	1h	
#60 Review Instructions & Help	3h5m	6h	
#120 Tutorial	0m	1d	
* 1.0 [Application]	2053m	1w485834m	18/07/2008
#153 Time Estimate report	0m	1d	11/07/2008
#222 Change task time estimate to remaining	0m	2d	
#171 Schedule page	2053m	1w	
#60 Allow workshop comments from Timer W	0m	4m	
#605 Milestone log entries	0m	2h	
#603 Update milestones on date change	0m	2h	
#109 Week start preferences	0m	1h	
#116 Handle resolutions lower than 1200x180	0m	1h	
#104 Chat Widget	0m	1h	
* 1.1 [Application]	3862m	7w6203h	13/09/2008
#177 Handle sheets with user_id NULL	0m	1d	
#104 New project by copying old project	0m	2d	
#101 Rework Filtering / Views	0m	2w	
#276 Task start date	0m	4h	
#103 Add not-tagged-with to views	0m	4h	
#133 Company News	0m	2h	
#456 Custom Fields	3h52m	1w	
#599 Custom Task Types	0m	1w	
#400 Custom Workflow	0m	1w	
#609 Default notification targets for client / P	0m	4h	
#197 Default User for tasks creation	0m	1h	
#108 Edit / Delete tags	0m	1d	
#602 Email Notification on project completion	0m	2h	
#606 Form widget for external pages	0m	0m	
#172 Graph of created vs closed per time ran	0m	4h	
#601 Group by creator	0m	0m	
#609 List previously run reports for download	0m	4h	
#112 Overdue	0m	1h	
#107 Per Project Wilks (Spacers)	0m	2d	
#150 Play sound on chat messages.	0m	4h	
#626 Project and milestone time quota	0m	0m	
#605 Reminder sent a configurable amount of	0m	2h	
#605 Reminders	0m	1h	
#603 RSS feeds for Views	0m	0m	
#620 Save reports input for later re-running	0m	4h	
#112 Set Active Milestone	0m	1h	
#121 Show Current Week	0m	1h	
#121 Show Status	0m	1h	

Clocking IT v0.99.2
Feedback? Suggestions? Ideas? Bugs? Admin

VIEWS New

- Open Tasks
- My Open Tasks
- My In Progress Tasks
- Unassigned Tasks
- Closed Tasks
- Defects
- New Features

NOTES New

- IPTS / Application (about 1 year ago)
- Ideas / Application (about 1 year ago)

Ellen Simonsen

Online Messages

22:26 Ellen S.
Hey dad!
22:27 Erleend S.
Hoyal Busy, busy.
Smile for the camera!
22:27 Ellen S.
"smiles"

Ellen S. 2 Online (2)



7.3 Elaboración de un proyecto básico de software.

7.3 Elaboración de un proyecto básico de software.

El objetivo específico de este subtema es elaborar un proyecto de software aplicando la metodología aprendida en este curso.

7. Desarrollo de un proyecto básico de software

Objetivo: Elaborar un proyecto básico de software aplicando la teoría del diseño de programas y herramientas de desarrollo.

- 7.1 Fases en el desarrollo de proyectos de software.
- 7.2 Herramientas de desarrollo en entornos visuales.
- 7.3 Elaboración de un proyecto básico de software.