

1. HERRAMIENTAS DE AYUDA A LA CREACIÓN DE PROGRAMAS:

- 1.1. Compilación separada (make)
- 1.2. **Entornos de desarrollo**
- 1.3. Bibliotecas de programas: estáticas y dinámicas
- 1.4. Herramientas de documentación
- 1.5. Análisis de rendimiento de programas (profiling)
- 1.6. Control de versiones (CVS)



El proceso de creación de programas

● Fases:

- ✓ Especificación de requisitos, análisis del problema → **PAPEL**
- ✓ Escribir programa → **EDITOR** (vi, emacs, MSWord, NotePad,...)
- ✓ Compilar programa → **COMPILADOR** (gcc, g++, VisualC++,...)
- ✓ Depurar programa → **DEPURADOR** (gdb,...)
- ✓ Documentar programa → **EDITOR, HERRAMIENTAS “ad hoc”**

- Se requieren múltiples programas que sirven de herramientas para construir nuevos programas.



Entornos de Desarrollo Integrado (IDE)

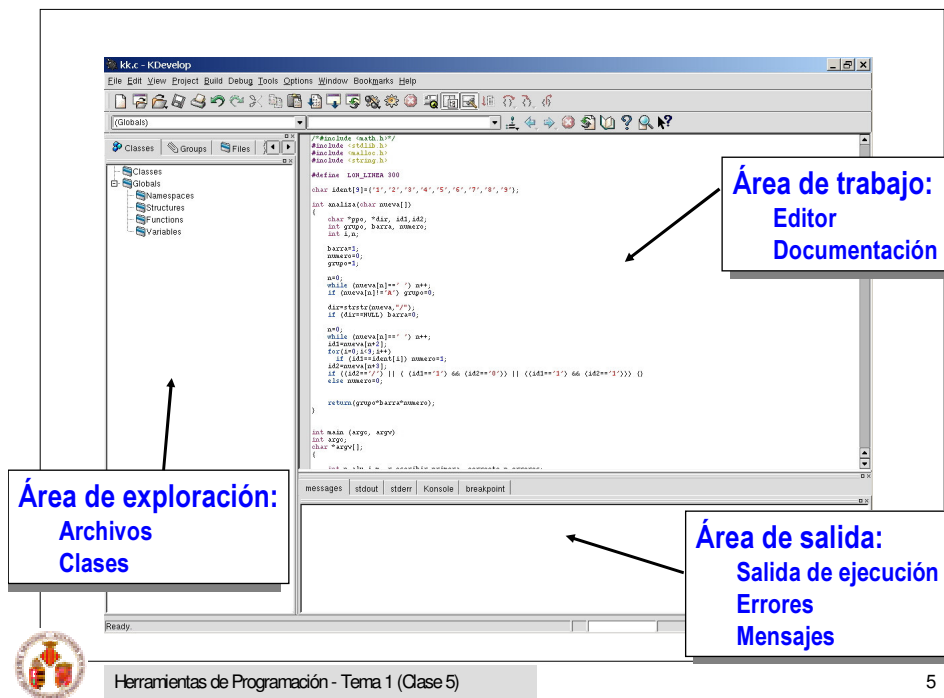
- El objetivo es combinar todas las herramientas necesarias para el desarrollo de programas en un único entorno de trabajo basado en GUI (Interfaz Gráfico de Usuario).
- Un IDE no es un compilador, no es un depurador, no es un editor, no es un explorador de archivos. Es todo a la vez.
 - ✓ Combina todos estos programas utilizando, en muchos casos, programas de uso habitual fuera del entorno (p.ej., *g++*, *gdb*, *emacs*) como órdenes en línea.
 - ✓ Algunas (o todas) de las aplicaciones que incorpora pueden ser exclusivas del IDE.



Entornos de Desarrollo Integrado (IDE) (2)

- Un IDE facilita el desarrollo de programas, incorporando herramientas para la gestión de proyectos que permiten:
 - ✓ Gestión de archivos
 - ✓ Gestión del proceso de compilación (*make*)
- Ejemplo de un IDE para Linux → **kdevelop**





Kdevelop

● Estructura de 3 áreas:

- ✓ Trabajo
- ✓ Salida
- ✓ Exploración

● Orientado a “Proyectos” Kde:

- ✓ Cargar proyecto “xaro”, ver estructura.
- ✓ Compilar
- ✓ Construir



- Repasar un poco los menús

- Construir un proyecto:

 - ✓ Leer el paso a paso

- Añadir clase “punto”

