

 Universidad
de Zaragoza

Laboratorio de Programación

3º Ingeniería en Informática

```
<meta name="generator" content="TP-EQUIV" />
<meta charset="iso-8859-1" />
<meta name="headline" content="Introduction" />
<meta name="section" content="Diseño Gráfico" />
<meta name="description" content="Some enterprises
need an immediate urgent re-engineering, let's take over." />
<!-- GEN Maps -->
<map name="barras" />
<area id="AAA" coords="7,9,167,32" />
<!-- iaaa index.html/1/hi.html" shape="RECT" />
</map>
```



<http://diis.unizar.es/>

Departamento de Informática e Ingeniería de Sistemas



Índice de la presentación

- ❑ Objetivos y organización
- ❑ Tipo de Trabajo a desarrollar
- ❑ Fases del Trabajo
- ❑ Sistema de Evaluación
- ❑ Recursos didácticos
- ❑ ¿Qué funciona bien? ¿Qué funciona mal?
- ❑ Reflexiones de futuro





Objetivos y Organización

- **Objetivos:**
 - ↳ **Análisis, diseño e implementación en grupo de un sistema software de tamaño medio**
 - ↳ **Integración de los conocimientos** adquiridos en las asignaturas previas de programación y **sensibilidad** ante la necesidad de nuevos conocimientos

- **Organización:**
 - ↳ Los estudiantes se dividirán en grupos de **tres personas**. A cada grupo se le asignará como tutor uno de los profesores responsables de la asignatura.
 - ↳ Cada grupo mantendrá una **reunión semanal** con su tutor en la que los estudiantes presentarán y discutirán ideas y propuestas y tomarán decisiones sobre cómo realizar el trabajo.





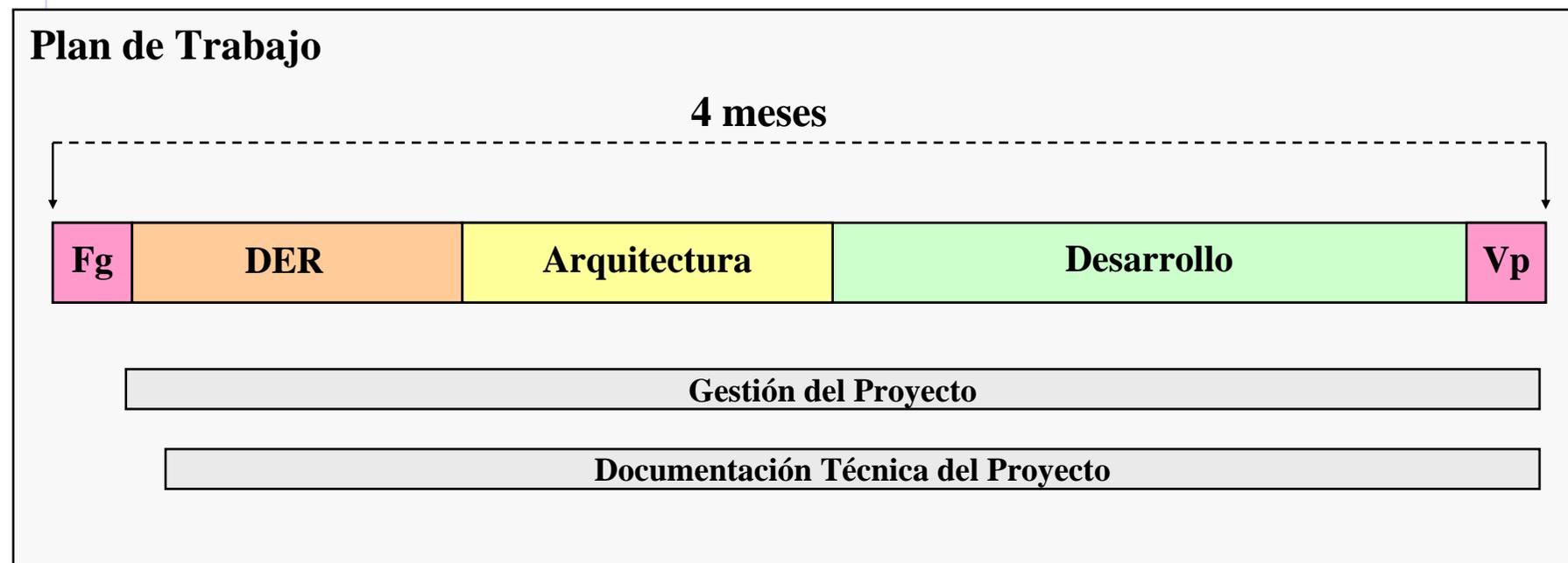
Trabajo a realizar

- ❑ Desarrollo del sistema informático de una **“CASA DE APUESTAS”**
 - ↳ Algoritmia, bases de datos, recuperación de información, etc..
 - ↳ No interfaz
 - ↳ No Web
- ❑ Lenguaje de programación: C
- ❑ Base de datos: ORACLE
- ❑ Plataforma:
 - ↳ HENDRIX
 - ↳ L0.01



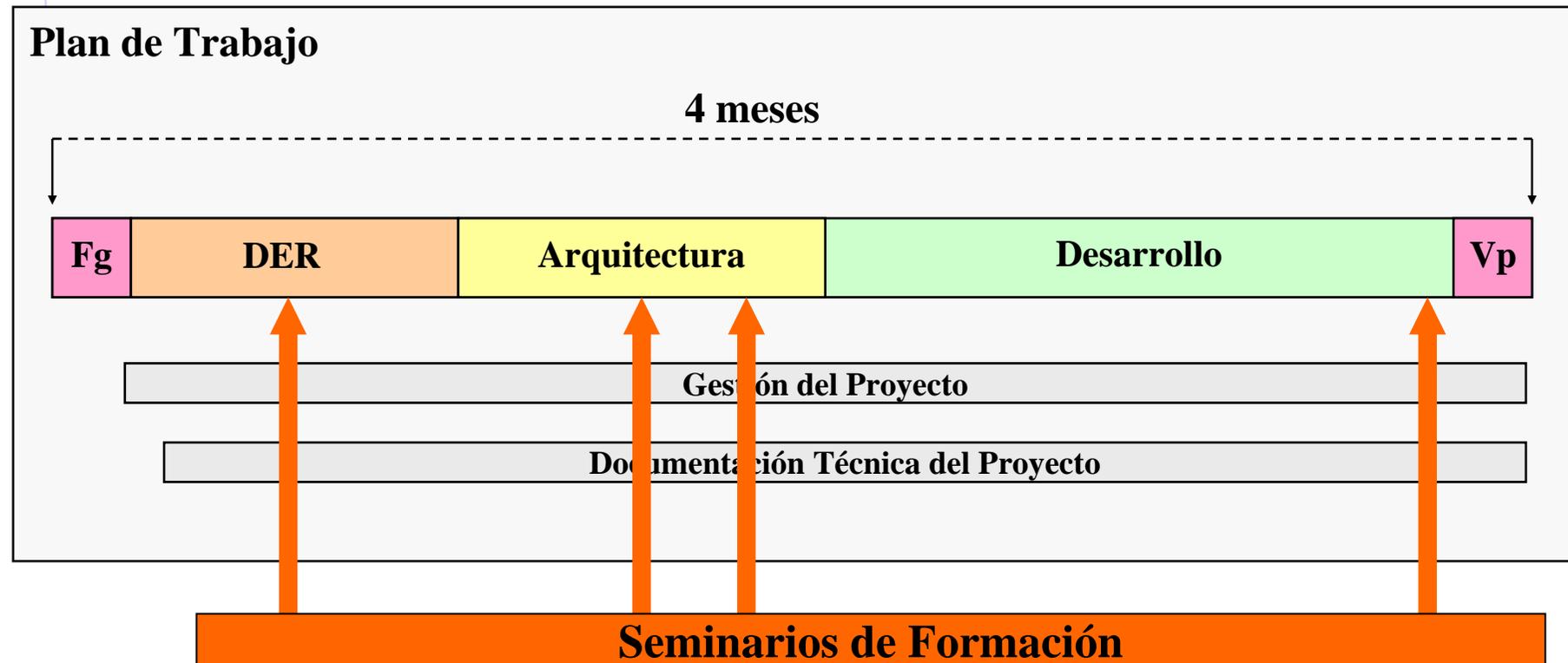


Visión general de las fases del trabajo





Visión general de las fases del trabajo





Evaluación (Fases generales)

- **Requisito Previo:** Entrega de la documentación técnica del proyecto, incluye la memoria y el código fuente generado
 - ⌚ ¿Cumple los criterios calidad mínimos? → Si cumple, No cumple o Podría ser mejorada en un tiempo razonable
- **Presentación del Producto:**
 - ⌚ Cada presentación durará 15 minutos + preguntas de los profesores y de los propios compañeros.
 - ⌚ Los alumnos valorarán los resultados de sus compañeros en función de los suyos propios, tanto de la presentación como del sistema software
 - ⌚ Los objetividad de los alumnos al valorar a sus propios compañeros será considerada por los profesores (evaluación crítica)





Evaluación (Elementos a premiar)

- Desde un punto de vista más técnico:
 - ↳ Originalidad y alcance de la propuesta
 - ↳ La metodología aplicada en la realización del trabajo
 - ↳ La calidad de las soluciones (nivel de diseño e implementación)
 - ↳ Documentación técnica del proyecto
 - ↳ El estudio de alternativas
 - ↳ La eficiencia de las soluciones finales
- Otros aspectos:
 - ↳ La “profesionalidad” del equipo
 - ↳ La participación e implicación en el trabajo diario
 - ↳ La transmisión de ideas (comunicación oral)
 - ↳ La capacidad de vender su trabajo y aportaciones





Evaluación (III)

- Así, de manera más precisa, se valorará:
 - ↳ 10% Cumplimiento de los plazos y de los objetivos recogidos en el documento de especificación de requisitos
 - ↳ 15% Seguimiento + Metodología aplicada (tutor)
 - ↳ 20% Documentación técnica del proyecto (fase previa)
 - ↳ 35% la calidad del programa (modularidad, legibilidad, robustez y eficiencia, estructuras de datos y algoritmos utilizados...)
 - ↳ 10% presentación del trabajo (15 minutos + preguntas de los profesores)
 - ↳ 10% evaluación a los compañeros y de los compañeros



Material Didáctico

Laboratorio de Programación

Primer cuatrimestre (Otoño) de Ingeniería en Informática. Centro Politécnico Superior. Universidad de Zaragoza

Fechas de interés

- 20 de septiembre 2007 - Presentación de la Asignatura
- 27 de septiembre del 2007 (14:00 h.) - Fecha límite para la formación de los grupos de trabajo (listados en aula A.06)
- 2 de octubre del 2007 - Publicación de la asignación de tutores a grupos de trabajo
- 17 de octubre del 2007 (14:00 h.) - Fecha límite para la entrega del documento de Especificación de Requisitos (DER)

Material relacionado con la asignatura

- Presentación (curso 07-08) ([Descargar](#))
- [Secuencias de escape](#) para posicionamiento del cursor, borrado de pantalla, etc.
- [Compiladores](#) disponibles en Hendrix

Otro material útil del curso 2006-07 (perfectamente reutilizable)

- Pautas orientativas para elaborar la memoria técnica ([Descargar](#))
- ¿Qué debo contar durante la [presentación de la asignatura?](#)

Seminarios

Cómo preparar una presentación

- Transparencias de la presentación ([Descargar](#))

Profesores

- Pedro Álvarez (Despacho 2.16, alvaper@unizar.es)
- Fran Ruiz (Despacho 3.01, franjr@unizar.es)
- Emilio Sobreviela (Despacho 2.14, emilio@ivo.cps.unizar.es)
- Rubén Béjar (Despacho 2.23, rbejar@unizar.es)
- Carmen Pellicer (Despacho 0.01, carmen.pellicer@unizar.es)
- Gregorio de Miguel (Pendiente de despacho y correo en Unizar)
- José María Falcó (Pendiente de despacho y correo en Unizar)

Información importante acerca de ANSI

Introducción

- He recibido una serie de preguntas acerca de como se pueden lograr distintos tipos de caracteres en la pantalla, ya sea en un terminal VT o en un PC.
- Aquí va algo de información acerca de ello. La mayoría esta sacada de postings al grupo alt.ascii-art, del manual de hexcalibur, del bfont y del programa colorit para Ms-Dos.
- Para ver una animación vt o algun texto con caracteres de control ansi en si, se usa el comando TYPE en Ms-Dos, y CAT en Unix.

Secuencias de escape

- Una secuencia de escape consiste del caracter `chr(27)`, seguido de una serie de caracteres, y se usa como un medio de comunicación entre el programa y el terminal o pantalla. El caracter 27 indica que lo que sigue es una secuencia de control, y debe ser interpretada y no transcrita al usuario. Usualmente se representa a este caracter como `esc` o `^f`.

Como crear una secuencia de escape

- Bajo Ms-Dos, la *única* solución que conozco es recurrir a un editor hexadecimal, como Hexed o HEXcalibur, para crear este caracter. Luego, con un editor de texto cualquiera (incluso el de Ms-Dos) se copia este caracter usando el clipboard (copy-paste o copiar-pegar)
- En Unix, lo primero es salir de SCREEN en caso de que se este usando este programa. Luego, se entra al vi y se escribe:

```
i Control-V Control-f
```

Información sobre compiladores disponibles

Compilador de Ada en Hendrix

- `/usr/local/gcc/bin/gcc (/usr/local/gcc/bin/gcc -c -x ada)`
- Por defecto las cuentas hacen uso del gcc versión 2.95.2 que se encuentra en el path `/usr/local/lib/gcc-lib/sparc-sun-solaris2.8/2.95.2/`, pero para compilar correctamente en Ada hay que usar el que se encuentra en `/usr/local/lib/gcc-lib/sparc-sun-solaris2.7/2.8.1`

Precompiladores en Hendrix

- Pro *C/C++
- Modada

Documentación básica de los precompiladores

- Pro *C/C++ : en `/usr/local/oracle/app/oracle/product/8.1.6/precomp/doc/proc/readme.doc`
- Modada : en `/usr/local/oracle/app/oracle/product/8.1.6/precomp/doc/mod/readme.txt`

Ejemplos con los precompiladores

- Pro *C/C++ : en `/usr/local/oracle/app/oracle/product/8.1.6/precomp/demo/proc`
- Modada : en `/usr/local/oracle/app/oracle/product/8.1.6/precomp/demo/modada`

Preguntas frecuentes

- [PREGUNTA] "Hemos intentado compilar Ada en Hendrix según la información encontrada en la página web de la asignatura. Sin embargo, no entendemos muy bien que es lo que hay que escribir exactamente para programar un .adb e incluso para precompilar código Ada o C donde haya SQL embebido [...]"
- [RESPUESTA] gnatmake no funciona correctamente en Hendrix, por la versión de compilador, es por eso que tenéis que usar `/usr/local/gcc/bin/gcc -c -x ada`. Con esto lo que consigues son los ficheros .ali y .o (Ada Library Information y objeto). Para obtener el ejecutable debes seguir los mismos pasos que hace el gnatmake: (1) `gcc -c -x ada 'file.adb'`; (2) `gnatbind -x 'file.ali'`; y (3) `gnatlink 'file.ali'`, o si prefieres una vez que tengas el *ali y el *o ejecuta `gnatmake 'file'.o` y funcionará. [...]



“Nuestra” visión ...

- ¿Qué funciona bien?
 - ↳ Nivel de implicación de los propios alumnos
 - ↳ Desarrollo de nuevas “competencias” profesionales
 - Comunicación oral y escrita, análisis, trabajo en grupo, planificación y organización, razonamiento crítico, autoaprendizaje, etc.
 - ↳ Perspectiva global de los conocimientos
 - ↳ Nivel de madurez adquirido por los alumnos

- ¿Qué debería ser mejorable?
 - ↳ Procesos y pautas metodológicas comunes, pero ...
 - ↳ El “feedback” a los alumnos
 - ↳ Valoración de los trabajos por parte de los profesores (!somos hasta 11 profesores!)
 - ↳ Necesitamos EVOLUCIONAR ...





¿Hacia dónde evolucionamos?

- **En principio, es un objetivo básico de este proyecto docente**
- No obstante, apuntamos algunas ideas:
 - ↳ Aplicar la idea de “Equipo de Grupos” al desarrollo de sistemas software más complejos (integración de sistemas)
 - Oppss!!!! Esto requiere mucho trabajo ☹
 - ↳ Experimentar con tecnologías “desconocidas” a priori
 - Oppss!!!! Esto requiere mucho trabajo ☹
 - ↳ Buscar contextos de problemas más próximos al mundo de la empresa (primera experiencia)
 - ↳ ...

