

## **Actividades UF1\_2:**

### **1a Part:**

1. Elabora los apuntes de esta Unidad y guárdalos en GitHub en un repositorio llamado 'Apuntes\_UF1\_2'.

[https://github.com/kevincaballerosanchez/Apuntes\\_UF1\\_2](https://github.com/kevincaballerosanchez/Apuntes_UF1_2)

2. ¿Para qué sirve un compilador? ¿Qué tipo de archivo obtenemos tras compilar?

Un compilador sirve para traducir tu código y así desde este generar el lenguaje ensamblador primero y luego traducir el lenguaje ensamblador al lenguaje máquina.

Al compilar un programa generamos uno o varios archivos objeto. Habitualmente los programas que compilemos harán uso de algunas funciones de biblioteca; en ese caso, el/los archivo/s objeto no son aún un fichero ejecutable, puesto que le falta añadir el código de esas funciones (lo cual hará el enlazador).

3. ¿Para qué sirve un enlazador? ¿Qué tipo de archivo obtenemos tras enlazar?

Un enlazador o linker es un programa que toma los objetos generados en los primeros pasos del proceso de compilación, la información de todos los recursos necesarios (bibliotecas), quita aquellos recursos que no necesita, y enlaza el código objeto con su(s) biblioteca(s) con lo que finalmente produce un fichero ejecutable.

4. ¿Para qué sirve un interprete? ¿Obtenemos algún archivo tras interpretar?

Un intérprete, al igual que un compilador, sirve para ejecutar código fuente de un lenguaje de alto nivel o un lenguaje de scripting pero sin tener que obtener código máquina como resultado final. Unos ejemplos de lenguajes interpretados son el lenguaje PHP, Python o el shell de comandos de Linux. Normalmente no obtenemos ningún archivo ya que el código se va ejecutando a medida que se necesita.

5. ¿Qué extensión tienen los archivos de código objeto?

COFF, también llamado Common Object File Format

6. Explica cada uno de los siguientes conceptos e indica la relación entre ellos.

- Código fuente: El código fuente de un programa informático (o software) es un conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un programa.
- Código objeto: Se llama código objeto al código que resulta de la compilación del código fuente.
- Código binario: El código binario es el sistema de codificación usado para la representación de textos, o procesadores de instrucciones de computadora, utilizando el sistema binario

El código fuente es el que compilaremos para obtener el código objeto, el cual después enlazaremos para que el sistema pueda ejecutarlo y leerlo en código binario.

7. ¿Qué tipo de código es el `bytecode` generado por el compilador de Java?

El `bytecode` es un código intermedio más abstracto que el código máquina, un archivo binario que contiene un programa ejecutable similar a un módulo objeto, que es un archivo binario producido por el compilador cuyo contenido es el código objeto o código máquina.

## 2a Part:

8. Bibliotecas. Define qué se entiende por biblioteca o librería y los tipos que existen. ¿Qué tipo es el más utilizado actualmente? ¿Por qué?

Una biblioteca o librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación. Existen las librerías estáticas y dinámicas, siendo las mas utilizadas las dinámicas, ya que los programas grandes necesitan todas las librerías a parte del ejecutable, lo podemos ver cuando les instalamos y abrimos la carpeta de la instalación.

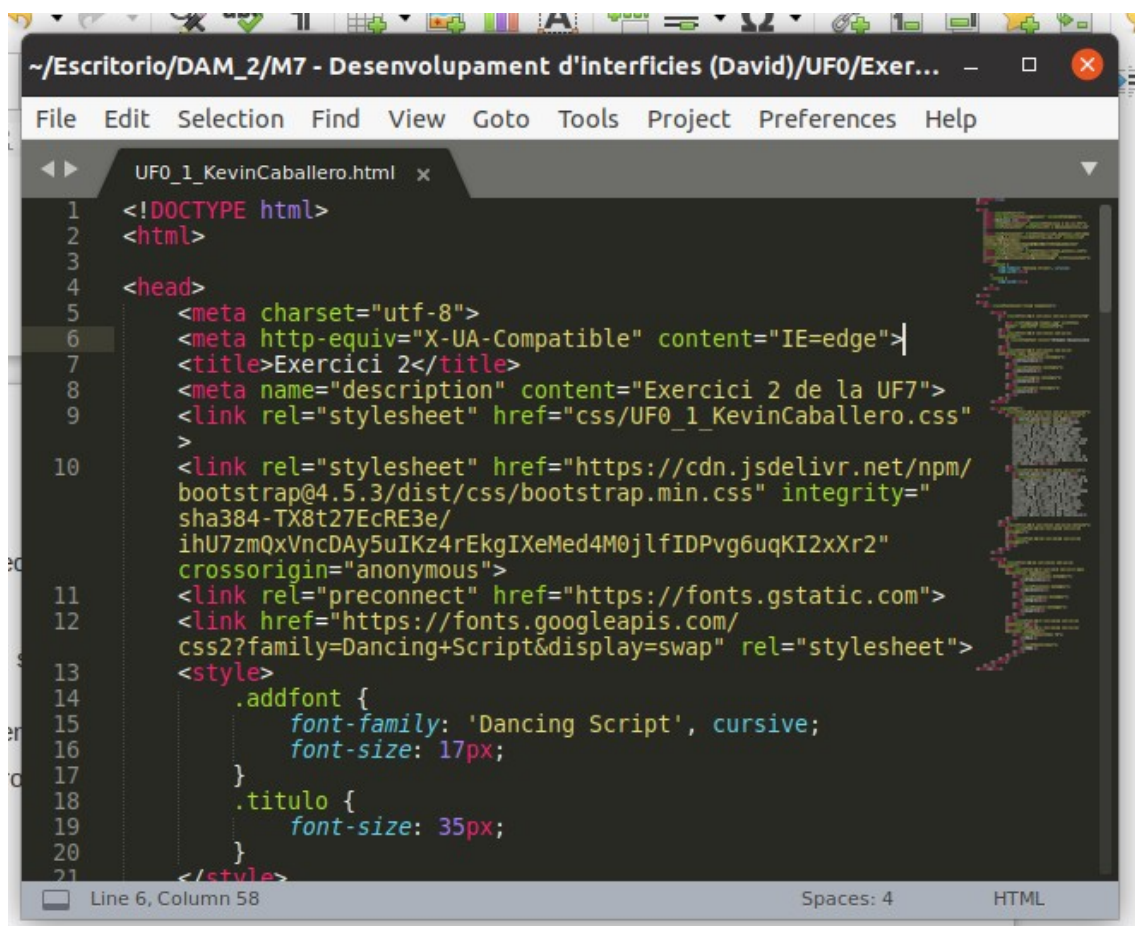
9. Crea una biblioteca dinámica en Java que proporcione las funciones para sumar, restar, multiplicar y dividir 2 números enteros. Crea un programa que haga uso de ella y comprueba que se ejecuta correctamente.

10. Bibliotecas. Busca información y explica las ventajas y desventajas de usar bibliotecas estáticas i dinámicas.

- Un programa compilado con librerías estáticas es más grande, ya que se hace copia de todo lo que necesita.
- Un programa compilado con librerías estáticas se puede llevar a otro ordenador sin necesidad de llevarse las librerías.
- Un programa compilado con librerías estáticas es, en principio, más rápido en ejecución. Cuando llama a una función de la librería, la tiene en su código y no tiene que ir a leer el fichero de la librería dinámica para encontrar la función y ejecutarla.

- Si cambiamos una librería estática, a los ejecutables no les afecta. Si cambiamos una dinámica, los ejecutables se ven afectados. Esto es una ventaja si hemos cambiado la librería para corregir un error (se corrige automáticamente en todos los ejecutables), pero es un inconveniente si tocar eso nos hace cambiar los ejecutables (por ejemplo, hemos añadido un parámetro más a una función de la librería, los ejecutables ya hechos dejan de funcionar).

11. Instala el editor sublime text. Saca una captura de pantalla una vez instalado.



```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="utf-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <title>Exercici 2</title>
8   <meta name="description" content="Exercici 2 de la UF7">
9   <link rel="stylesheet" href="css/UF0_1_KevinCaballero.css"
10
11   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/
bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="
sha384-TX8t27EcRE3e/
ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
12   <link rel="preconnect" href="https://fonts.gstatic.com">
13   <link href="https://fonts.googleapis.com/
css2?family=Dancing+Script&display=swap" rel="stylesheet">
14   <style>
15     .addfont {
16       font-family: 'Dancing Script', cursive;
17       font-size: 17px;
18     }
19     .titulo {
20       font-size: 35px;
21     }
22   </style>
```

12. ¿Para qué sirve el complemento Emmet? Haz uso de él al escribir un archivo html de prueba.

Es ni más ni menos que un plugin de diseño que se adapta a los editores de texto mas populares y nos permite escribir nuestro código HTML de forma abreviada, es cierto que hoy en dia existen los *snippets* que nos permiten generar pequeños fragmentos de código que para nosotros es reusable, pero Emmet va mas allá, no tienes que generarte tus fragmentos de codigo o snippets y luego recordar como los fuiste llamando y que hacia cada uno (ojo que los snippets tambien son un puntazo y en algunos casos conviene usarlos), con Emmet tienes ya todo predefinido, aprende a escribir el codigo de forma abreviada y Emmet ya genera el resto, olvidate de andar creando plantillas de codigo y luego exportandolas si cambias de equipo o de editor, con Emmet instalas el plugin y a correr.

13. Instala el entorno integrado Netbeans.

14. Crea un proyecto en Java en Netbeans. Realiza su compilación y ejecución.

