

Programación - 2526

Área personal / Mis cursos / FORMACIÓN PROFESIONAL 25/26 / CIFP CARLOS III / CGS Desarrollo de Aplicaciones Web 25/26 / Primero(DAW) / Programación - 2526 / PRUEBAS CALIFICABLES / Tarea RA2 PROG03

Navegación

▼ Área personal

[Página principal del sitio](#)

► Educación a distancia para la Formación Profesional

▼ Mis cursos

► FORMACIÓN PROFESIONAL 25/26

▼ CIFP CARLOS III

► CGS Desarrollo de Aplicaciones Web 25/26

► Primero(DAW)

► [Bases de Datos - 2526](#)

► [Digitalización](#)

► [Aplicada a los Sectores](#)

► [Productivos...](#)

► [Entornos de Desarrollo - 2526](#)

► [Inglés Profesional para DAW - 2526](#)

► [Lenguajes de marcas y sistemas de gestión de información - 2526](#)

► [Programación - 2526](#)

► [Participantes](#)

► [Competencias](#)

► [Calificaciones](#)

► [Programación](#)

► [COMUNICACIÓN](#)

► [TEMAS](#)

► [RA1](#)

► [RA2](#)

► [*RA3](#)

▼ PRUEBAS CALIFICABLES

► [Cuestionario RA1 PROG01](#)

► [Cuestionario RA1 PROG02](#)

► [Tarea RA1 PROG01](#)

► [Tarea RA1 PROG02](#)

► [Cuestionario RA2](#)

► [Tarea RA2 PROG03](#)

► [Sistemas informáticos \(PRL\) - 2526](#)

► [Tutoría 1º DAW - 2526](#)

► [Punto de Encuentro Alumnado CIFP CARLOS III - 2526](#)

Tarea RA2 PROG03

Apertura: martes, 4 de noviembre de 2025, 10:00

Cierre: jueves, 20 de noviembre de 2025, 23:55

Detalles de la tarea de esta unidad.

Enunciado.

En esta unidad has visto las características fundamentales de la programación orientada a objetos y estudiado los conceptos más importantes de este modelo de programación. Has aprendido a crear y manipular objetos, qué son los métodos y cómo se definen los parámetros. También has podido conocer cómo se estructura la Biblioteca de Clases de Java, viendo algunas clases importantes, como por ejemplo las que nos permiten realizar la programación de la consola.

Ejercicio 1

- Construye un nuevo proyecto Java denominado **Prog03_Ejerc1**. En el proyecto debe aparecer un paquete, que no puede ser el paquete por defecto, ponle el nombre que creas oportuno. Dentro de dicho paquete:
- Crea una clase denominada **Fecha**. Esta clase no debe contener método **main**.
- Declara en el fichero de la clase un tipo enumerado, denominado **enumMes**, para los meses del año.
- La clase debe contener un atributo para el día, otro para mes (del tipo enumerado declarado) y un tercero para el año.
- Implementa un constructor que inicialice el mes al valor recibido por parámetro y los demás atributos a 0. Observa su cabecera en el siguiente código:

```
Fecha (enumMes mes){
```

- Declara otro constructor que inicialice todos los atributos de la clase. Su cabecera podría ser la siguiente:

```
Fecha (int dia, enumMes mes, int anio){
```

- Implementa los métodos que permitan acceder y modificar cada uno de los atributos de la clase. Los nombres de dichos métodos serán: **getXXX ()** para obtener el valor del atributo **XXX** y **setXXX (v)** para actualizar el atributo **XXX** con el valor **v**. Puedes observar la cabecera de algún método en el siguiente código:

```
public int getDia() {
```

```
}
```

```
public void setDia(int dia) {
```

```
}
```

- Implementa un método que devuelva **true** si el valor contenido en la fecha es verano y **false** en caso contrario. Observa su cabecera en el siguiente código:

```
public boolean isSummer(){
```

No es necesario utilizar estructuras condicionales pues aún no las hemos trabajado. Se puede implementar este método utilizando operadores de comparación.

- Implementa un método que devuelva una cadena con la fecha en formato largo, por ejemplo, 15 de julio de 2020. Observa su cabecera:

```
public String toString (){
```

Ya tenemos nuestra clase **Fecha** implementada. Ahora vamos a probar su funcionalidad desde otra clase, denominada **Principal**, que contendrá el método **main**. Esta clase la debes crear en el mismo paquete que la clase **Fecha**. Dentro de dicha clase:

- Instancia un objeto de la clase **Fecha** denominado **objFecha1** con el primer constructor.
- Actualiza los atributos dia y año para dicho objeto.
- Muestra la fecha por pantalla en formato largo.
- Muestra un mensaje por pantalla indicando si la fecha es verano (puedes utilizar el operador de comparación ternario. Punto 5.3 de la Unidad 2).
- Instancia otro objeto de la clase **Fecha** denominado **objFecha2** con el segundo constructor.
- Muestra el año de esta fecha por pantalla.
- Muestra la fecha en formato largo por pantalla.
- Muestra un mensaje por pantalla indicando si la fecha es verano o no.

Observa un ejemplo de ejecución:

```
Primera fecha, inicializada con el primer constructor
```

```
La fecha es: 20 de febrero del 2000
```

```
No es verano
```

```
Segunda fecha, inicializada con el segundo constructor
```

```
La fecha 2 contiene el año 2015
```

```
La fecha es: 15 de julio del 2015
```

```
Es verano
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejercicio 2

- Crea un proyecto Java denominado **Prog03_Ejerc2**.
- Dentro del proyecto, crea un paquete denominado **com.prog03.figuras**.
- Dentro de dicho paquete, crea una clase denominada **Rectangulo** que:
 - Declare atributos para la base y la altura de un rectángulo.
 - Declare un constructor vacío que inicialice los atributos a 0.
 - Declare un constructor que inicialice base y altura.
 - Declare los siguientes métodos:
 - Métodos para actualizar y obtener el valor de cada atributo.
 - float getArea()**: devuelve el área del rectángulo.
 - String toString()**: devuelve una cadena conteniendo su área y su altura.
 - boolean isCuadrado()**: devuelve un booleano indicando si el rectángulo es cuadrado.
- Crea otro paquete con el nombre **com.prog03.Principal**.
- Dentro de dicho paquete crea una clase denominada **principal** que contenga el método **main**.
- En el método **main**, instancia al menos dos objetos de la clase **Rectangulo** y comprueba su funcionamiento.

Criterios de puntuación. Total 10 puntos.

- Ejercicio 1: 4 puntos.
- Ejercicio 2: 6 puntos.
- Total: 10 puntos.

Recursos necesarios para realizar la Tarea.

- Ordenador personal.
- JDK.
- IntelliJ.

Consejos y recomendaciones.

Para una mejor organización de la tarea, crea una carpeta para cada ejercicio llamada EjercicioXX, donde la XX es el número del ejercicio. Sitúa dentro de cada carpeta los ficheros de cada proyecto y cuando todos los ejercicios estén completos comprime esas carpetas en un sólo archivo para subirlo a la plataforma.

Añade al código fuente los comentarios que consideres oportuno.

Indicaciones de entrega.

Se deben entregar cada uno de los proyectos IntelliJ creados.

Una vez realizada la tarea elaborarás un único fichero comprimido (zip o rar) que contenga los dos ficheros correspondientes a los proyectos. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre.SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la tercera unidad del MP de PROG**, debería nombrar esta tarea como...

sanchez_manas_begona_PROG03_Tarea

[Agregar entrega](#)

Estado de la entrega

Número del intento	Este es el intento 1.
Estado de la entrega	Todavía no se han realizado envíos
Estado de la calificación	Sin calificar
Tiempo restante	3 días 14 horas restante
Última modificación	-
Comentarios de la entrega	Comentarios (0)

[Cuestionario RA2](#)

Ir a...

[Contactar con el soporte del sitio](#)

Usted se ha identificado como **LAURA PAREDES ALCARAZ** (Cerrar sesión)

2526 30019702 12242003 02 00485

Resumen de retención de datos

Descargar la app para dispositivos móviles