



CICLO FORMATIVO DE GRADO SUPERIOR

IFC303 - DESARROLLO DE APLICACIONES WEB

MÓDULO: PROGRAMACIÓN

MODALIDAD: A DISTANCIA

Módulo
PROGRAMACIÓN. (0485)

Profesor: Eduardo Carreira López

Centro Educativo

<i>Código</i>	<i>Centro</i>	<i>Año académico</i>
33028167	CIFP Sectores Industriales y Servicios	2023/2024

Ciclo formativo

<i>Familia profesional</i>	<i>Ciclo formativo</i>	<i>Código</i>	<i>Curso</i>
Informática y Comunicaciones	Desarrollo de Aplicaciones WEB	IFC303	1º

LEGISLACIÓN APLICABLE	IFC303 Desarrollo de Aplicaciones WEB <ul style="list-style-type: none"> • Ley Orgánica 2/2006, de 3 de mayo, de Educación. • Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la anterior. • Real Decreto 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo • Ley Orgánica 3/2022, de 31 de marzo, de ordenación e integración de la Formación Profesional. • DAW - Real Decreto 686/2010 del Ministerio de Educación, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas. • DAW - Decreto 184/2012 de la Consejería de Educación, Cultura y Deporte del Principado de Asturias, de 8 de agosto, por el que se establece el currículo del ciclo formativo de Grado Superior de Formación Profesional en Desarrollo de Aplicaciones Web.
Nº de créditos	ECTS: 14
Horas/curso	224 horas
Horas/semana	7 horas, distribuidas 3 + 2 + 2
Especialidad del profesorado	PROFESOR DE ENSEÑANZA SECUNDARIA – 107 - Informática

ÍNDICE

1. RESULTADOS DE APRENDIZAJE QUE DEFINEN EL MÓDULO	4
2. CRITERIOS DE EVALUACIÓN MATIZADOS Y RELACIÓN CON LOS RA	5
3. CONTENIDOS	8
4. CRITERIOS DE CALIFICACIÓN	12
4.1. Evaluación ORDINARIA	
4.2. Evaluación final EXTRAORDINARIA	
4.3. Evaluación Alumnado que promociona a 2º con el módulo PENDIENTE	
5. SESIONES DE FORMACIÓN TELEMÁTICAS Y TUTORÍAS INDIVIDUALES	13
6. PLANIFICACIÓN SESIONES TUTORÍA COLECTIVA	14
7. PLANIFICACIÓN DE ACTIVIDADES DE EVALUACIÓN	16

1. RESULTADOS DE APRENDIZAJE QUE DEFINEN EL MÓDULO

Resultados de Aprendizaje	
RA1	Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
RA2	Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
RA3	Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
RA4	Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
RA5	Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
RA6	Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.
RA7	Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
RA8	Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
RA9	Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.

2. CRITERIOS DE EVALUACIÓN EN RELACIÓN CON RESULTADOS DE APRENDIZAJE

Resultados de Aprendizaje	Criterios de Evaluación
RA1 Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	a) Se han identificado los bloques que componen la estructura de un programa informático. b) Se han creado proyectos de desarrollo de aplicaciones c) Se han utilizado entornos integrados de desarrollo. d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno. e) Se ha modificado el código para crear y utilizar variables. f) Se han creado y utilizado constantes y literales. g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje. h) Se ha comprobado el funcionamiento de las conversiones de tipo i) Se han introducido comentarios en el código.
RA2 Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.	a) Se han identificado los fundamentos de la programación orientada a objetos. b) Se han escrito programas simples. c) Se han instanciado objetos a partir de clases predefinidas. d) Se han utilizado métodos y propiedades de los objetos. e) Se han escrito llamadas a métodos estáticos. f) Se han utilizado parámetros en la llamada a métodos. g) Se han incorporado y utilizado librerías de objetos. h) Se han utilizado constructores. i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.
RA3 Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.	a) Se ha escrito y probado código que haga uso de estructuras de selección. b) Se han utilizado estructuras de repetición. c) Se han reconocido las posibilidades de las sentencias de salto. d) Se ha escrito código utilizando control de excepciones. e) Se han creado programas ejecutables utilizando diferentes estructuras de control. f) Se han probado y depurado los programas. g) Se ha comentado y documentado el código.
RA4 Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.	a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase. b) Se han definido clases. c) Se han definido propiedades y métodos. d) Se han utilizado constructores y destructores. e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente. f) Se han utilizado mecanismos para controlar la visibilidad de las clases y sus miembros. g) Se han definido y utilizado clases heredadas. h) Se han creado y utilizado métodos estáticos. i) Se han definido y utilizado interfaces. j) Se han creado y utilizado conjuntos y librerías de clases.

<p>RA5</p> <p>Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>	<ul style="list-style-type: none"> a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información. b) Se han aplicado formatos en la visualización de la información. c) Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas. d) Se han utilizado ficheros para almacenar y recuperar información. e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros. f) Se han implementado técnicas para almacenamiento de objetos en ficheros g) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples. h) Se han programado controladores de eventos. i) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.
<p>RA6</p> <p>Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.</p>	<ul style="list-style-type: none"> a) Se han escrito programas que utilicen arrays. b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados. c) Se han utilizado listas para almacenar y procesar información. d) Se han utilizado iteradores para recorrer los elementos de las listas. e) Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles. f) Se han creado clases y métodos genéricos. g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto. h) Se han identificado clases relacionadas con tratamiento de documentos XML i) Se han realizado programas que realicen manipulaciones sobre documentos XML.
<p>RA7</p> <p>Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.</p>	<ul style="list-style-type: none"> a) Se han identificado los conceptos de herencia, superclase y subclase. b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos. c) Se ha reconocido la incidencia de los constructores en la herencia. d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase. e) Se han diseñado y aplicado jerarquías de clases. f) Se han probado y depurado las jerarquías de clases. g) Se han realizado programas que implementen y utilicen jerarquías de clases. h) Se ha comentado y documentado el código.
<p>RA8</p> <p>Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.</p>	<ul style="list-style-type: none"> a) Se han identificado las características de las bases de datos orientadas a objetos. b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos. c) Se han instalado sistemas gestores de bases de datos orientados a objetos. d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada. e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos. f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas. g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos. h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

RA9 Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.	<ul style="list-style-type: none">a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.b) Se han programado conexiones con bases de datos.c) Se ha escrito código para almacenar información en bases de datos.d) Se han utilizado diferentes asistentes del entorno de desarrollo.e) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.f) Se han efectuado borrados y modificaciones sobre la información almacenada.g) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.h) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.
---	---

3. CONTENIDOS

									Unidad de Trabajo		Horas
RA1	RA2	RA3	RA4	RA5	RA6	RA7	RA8	RA9			
X									1	Elementos de un programa informático.	3
	X	X							2	Lenguaje de programación Java. Primeros pasos. Estructuras de control. Estructuras básicas de almacenamiento (Arrays). Clases para el procesamiento de cadenas alfanuméricas.	21
			X						3	Fundamentos de la Programación Orientada a Objetos (POO). Clases y Objetos en Java.	6
						X			4	Utilización avanzada de Clases en Java.	12
					X				5	Estructuras de almacenamiento en Java. Colecciones de datos.	12
				X					6	Lectura y escritura de información en Java. Ficheros	6
							X	X	7	Persistencia de objetos sobre Sistemas de Gestión de Bases de datos.	6

UT1. ELEMENTOS DE UN PROGRAMA INFORMÁTICO	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Conceptos iniciales. • Sistemas de procesamiento de la información. • Algoritmos. • Aplicaciones informáticas. Ciclo de vida. • Lenguajes de programación. • Código fuente-máquina-intermedio. • Compiladores e intérpretes. • Herramientas de programación. • Estructura y empaquetado de una aplicación informática. • Elementos que forman parte del código: variables, constantes, literales, expresiones, operadores. • Abstracción en la programación: Programación Modular y reutilización de código. 	<p>RA1</p> <p>Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.</p>
UT2. LENGUAJE DE PROGRAMACIÓN JAVA. PRIMEROS PASOS. ESTRUCTURAS DE CONTROL. ESTRUCTURAS BÁSICAS DE ALMACENAMIENTO (ARRAYS). CLASES PARA EL PROCESAMIENTO DE CADENAS ALFANUMÉRICAS.	RESULTADOS DE APRENDIZAJE
<p>El Lenguaje JAVA</p> <ul style="list-style-type: none"> • El lenguaje JAVA. JRE y JDK. • Estructura de un programa JAVA. Paquetes. • Clases. Main Class o clase Driver. • Literales en Java. • Identificadores en Java. • Operadores en Java. • Tipos de sentencias en Java. • Sentencias de expresión. • Sentencias de declaración. • Sentencias de control de flujo. • Entrada y salida básica. • Modularidad. Métodos. <p>Estructuras de control</p> <ul style="list-style-type: none"> • Estructuras condicionales: Simples, dobles, múltiples, operador ternario. • Estructuras repetitivas: Bucles. <p>ARRAYS</p> <ul style="list-style-type: none"> • Operaciones con Arrays unidimensionales. • Operaciones con Arrays Bidimensionales. • La clase estática Array. Métodos. <p>STRINGS (Cadenas alfanuméricas)</p> <ul style="list-style-type: none"> • La Clase String: Métodos. • Clases String avanzadas: StringBuilder y StringBuffer. 	<p>RA2</p> <p>Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.</p> <p>RA3</p> <p>Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.</p>

UT3. FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS (POO). CLASES Y OBJETOS EN JAVA	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Programación estructurada y POO. • Conceptos iniciales. • Conceptos fundamentales de la POO. • Clases. Atributos y métodos. • Clases y Objetos. • Definición de clases en Java. • Control de acceso en Java. • Objetos o instancias de una clase. • Referencias a Objetos. • Métodos o mensajes. • Constructores. 	<p>RA4</p> <p>Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.</p>
UT4. UTILIZACIÓN AVANZADA DE CLASES EN JAVA	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Acceso a atributos privados. • Selectores y Mutadores. • Sobrecarga de métodos. • Sobrecarga de constructores. • Operadores this / Super. • Herencia. Control de acceso. • Redefinición de métodos. • Polimorfismo. • Clases Abstractas. • Interfaces. • Herencia múltiple con Interfaces. • Control de Excepciones. Clase Exception. 	<p>RA7</p> <p>Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.</p>
UT5. ESTRUCTURAS DE ALMACENAMIENTO. COLECCIONES	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Tipos de colecciones. • Jerarquía de colecciones. • Interface Collection. • Tipos de Colecciones: Listas, Conjuntos y mapas. • Objeto Iterator. • Operaciones con colecciones: Acceso a los elementos - recorridos. • Ordenaciones. Clase Comparator e interface Comparable. • Uso de las clases más importantes y significativas y sus métodos principales: ArrayList, LinkedList, HashSet, HashMap, TreeMap. 	<p>RA6</p> <p>Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.</p>

UT6. LECTURA Y ESCRITURA DE INFORMACIÓN EN JAVA. FICHEROS	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Introducción. Clase File. • Operaciones con archivos y carpetas a través de la clase File. • Clase para la gestión de Excepciones en operaciones con archivos. • Clases relativas a flujos de datos. • Flujos de Bytes. • Flujos de caracteres. • Lectura y escritura de datos binarios. • Archivos de acceso aleatorio. • Almacenamiento de objetos en ficheros: Persistencia. • Persistencia de objetos en Archivos de texto. • Persistencia de objetos en archivos binarios. Serialización. 	<p>RA5</p> <p>Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.</p>
UT7. PERSISTENCIA DE LOS OBJETOS SOBRE SISTEMAS DE GESTIÓN DE BASES DE DATOS	RESULTADOS DE APRENDIZAJE
<ul style="list-style-type: none"> • Clases para el establecimiento de conexiones. • Clases para la recuperación de la información contenida en las B.D. • Manipulación de la información contenida en las B.D. Creación de tablas, inserción, actualización y eliminación de registros. • Consultas. Ejecución de consultas sobre la base de datos. • Bases de Datos Orientadas a Objetos. • Características de las Bases de Datos Orientadas a Objetos. 	<p>RA8</p> <p>Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.</p> <p>RA9</p> <p>Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.</p>

4. CRITERIOS DE CALIFICACIÓN

La evaluación tomará como referencia los Resultados de Aprendizaje del módulo. Será **obligatoria** la realización de al menos una prueba objetiva presencial individual. **Opcionalmente** el alumnado podrá realizar 2 pruebas parciales, una por cuatrimestre.

4.1. EVALUACIÓN ORDINARIA.

- **Primer cuatrimestre.** En el mes de **febrero** se llevará a cabo una sesión de evaluación parcial (**opcional**) mediante la que se puede **superar** la materia impartida hasta la fecha. (**Fecha prueba: 8/2/2024**)
 - **Segundo cuatrimestre.** En el mes de **mayo** se llevará a cabo una sesión de evaluación FINAL ORDINARIA del módulo. En esta sesión, el alumnado **que haya superado** el primer cuatrimestre no tendrá que examinarse de la materia correspondiente al mismo. (**Fecha prueba: 16/5/2024**)
- Las sesiones de evaluación se calificarán con una nota numérica del 1-10.
 - Las sesiones de evaluación tendrán un peso del **60%** en la nota final, y será necesario superar **una nota de 4** para poder optar a sumar el otro **40%**, proveniente de la media obtenida en las **Tareas/Prácticas evaluables**.

Tareas evaluables:

A lo largo del curso se irán desarrollando diferentes tareas evaluables. El profesor indicará para cada trabajo una fecha límite de entrega. La entrega de tareas no es obligatoria. El alumno que no entregue tarea alguna será calificado con 0 puntos en el correspondiente apartado. Las tareas evaluables en su conjunto tendrán un peso del 40% de la nota (Máximo 4 puntos).

Cálculo de la NOTA: (Nota media trabajos evaluables * 0,4 + Nota prueba práctica presencial * 0,6)

ADVERTENCIA: La entrega de las tareas evaluables en tiempo y forma **no implica su valoración incondicional**. En determinados casos, y cuando el profesor lo estime oportuno, podrá convocar al alumnado en tutoría individual para realizar pruebas de **verificación de autoría**. Puede darse el caso de la no valoración de tareas entregadas en plazo si no se supera las pruebas de autoría. Las pruebas consistirán en la defensa del trabajo entregado y la realización “en directo” de pequeñas variaciones sobre la misma, propuestas por el profesor/a.

4.2. EVALUACIÓN FINAL EXTRAORDINARIA (JUNIO)

En fecha a determinar, entre el 17 y el 20 de junio de 2024, se realizará una prueba de **evaluación FINAL EXTRAORDINARIA** para el alumnado que no haya superado el módulo en la evaluación ordinaria. La prueba versará sobre toda la materia del curso, excepto en el caso de alumnos/as que se presentaron y superaron el primer parcial cuatrimestral, en cuyo caso esa parte se les considera superada y no se les evaluará de nuevo.

- Para superar esta prueba, el alumno/a tendrá que obtener una nota igual o superior a 5.
- No se pedirán tareas evaluables, por lo que el 100% de la nota final saldrá de la prueba.

4.3. (OPCIONAL) EVALUACIÓN ALUMNADO CON EL MODULO PENDIENTE (MARZO)

El alumnado que ya haya cursado este módulo en cursos anteriores, sin haberlo superado, puede (opcionalmente) solicitar la evaluación del módulo como PENDIENTE. En ese caso se le convocará a una prueba específica en el mes de **Marzo (con fecha a determinar)**. Esta prueba hace las veces de convocatoria ORDINARIA FINAL para los alumnos/as que soliciten la evaluación como PENDIENTE. En caso de no obtener una nota mayor o igual a 5 en dicha prueba, deberán optar a la **convocatoria EXTRAORDINARIA de junio**. El alumnado que quiera cursar el módulo como PENDIENTE **deberá solicitarlo en la secretaría del centro a principio de curso (Durante el mes de Septiembre)**

5. SESIONES DE FORMACIÓN TELEMÁTICA. TUTORÍAS INDIVIDUALES

Las sesiones de formación telemática (tutorías colectivas) se llevarán a cabo en el aula A202, los jueves de 17:20 a 20:20. **Los alumnos podrán asistir al aula o conectarse en remoto a través del equipo Teams en el que serán matriculados a principio del curso.**

El calendario de sesiones de tutoría colectiva (formación) es el siguiente (*Están excluidas las fechas de realización de exámenes presenciales en **febrero y mayo**, tanto de 1º como de 2º curso*):

2023

- Setiembre, días 14, 21, 28
- Octubre, días 5, 19, 26
- Noviembre, días 9, 16, 23, 30
- Diciembre, días 14, 21

2024

- Enero, días 11, 18, 25
- Febrero, días 1, 22, 29
- Marzo, días 7, 14, 21
- Abril, días 4, 11, 18, 25
- Mayo, días 9, 30
- Junio, días 6, 13

➤ **La tutoría individual será los viernes 9:25 a 10:20**
(Será necesario pedir cita previa por email/Teams y podrá realizarse en el CFP o de forma telemática)

6. PLANIFICACIÓN DE SESIONES DE FORMACIÓN A LO LARGO DEL CURSO

Se trata de una **previsión**, y cómo tal se podrá ver alterada por las circunstancias que vayan surgiendo a lo largo del curso. Más fiable será la planificación en tiempo real que se irá haciendo sesión a sesión, en la que se irá informando al alumnado de los contenidos y alcance de las tutorías colectivas, siempre con la debida antelación y a través de los medios adecuados (aula virtual Educastur + Teams).

13/09/2023	Acogida presencial en el Aula 202 del CIFP. Ajustes cuentas educastur/campus/teams.
21/09/2023	Presentación.
28/09/2023	UT 1. Explicación de conceptos básicos. Programación. Lenguajes. Características de Java
5/10/2023	UT 2. Introducción a la programación en JAVA. Primeros pasos.
19/10/2023	UT 2. Fundamentos de Java. Tipos de datos, variables, constantes, operadores, sentencias, sentencias de control.
26/10/2023	UT 2. Codificación de programas sencillos en Java utilizando el IDE BlueJ.
9/11/2023	UT 2. Codificación de programas en Java, utilizando diferentes estructuras de control y resto de estructuras del lenguaje.
16/11/2023	UT 2. Codificación de programas en Java. Manejo del API. Cadenas. String y StringBuilder
23/11/2023	UT 2. Codificación de programas en Java. Arrays.
30/11/2023	UT 2. Codificación de programas en Java. Repaso y conclusiones.
14/12/2023	UT 3. Programación orientada a objetos. Conceptos fundamentales. Clases, objetos, métodos y propiedades.
21/12/2024	UT 4. Clases y objetos en Java. Declaración de clases, constructores y métodos. Control de acceso. Sobrecarga de métodos. Selectores y mutadores.
11/01/2024	UT 4. Utilización avanzada de clases en Java. Herencia. Polimorfismo. Interfaces.
18/01/2024	UT 4. Utilización avanzada de clases en Java. Continuación.
25/01/2024	UT 4. Utilización avanzada de clases en Java. Continuación.
1/02/2024	Resolución de DUDAS previa a la prueba objetiva parcial de Febrero.
08/02/2024	PRUEBA OBJETIVA INDIVIDUAL PARCIAL (VOLUNTARIA) La prueba PARCIAL se solicita directamente al profesor del módulo, y para ello se habilitará en su momento un sondeo previo en el aula virtual.
15/02/2024	PRUEBAS DE 2º CURSO

22/02/2024	UT 4. Gestión de errores. Excepciones.
29/02/2024	UT 5. Estructuras de almacenamiento avanzado en memoria. Colecciones
07/03/2024	UT 5. Colecciones. Continuación.
14/03/2024	UT 5. Colecciones. Continuación.
21/03/2024	UT 6. Lectura y escritura de información en Java. Ficheros. Persistencia de Objetos.
04/04/2024	UT 6. Ficheros. Continuación.
11/04/2024	UT 7. Persistencia sobre SGBD. JDBC.
18/04/2024	UT 7. JDBC. Continuación.
25/04/2024	Sesión reservada para cubrir posibles retrasos o en su defecto para repaso/dudas
09/05/2024	Resolución de DUDAS previa a la prueba objetiva FINAL ORDINARIA
16/05/2024	PRUEBA FINAL ORDINARIA
23/05/2024	PRUEBAS DE 2º CURSO
30/05/2024	Repaso y resolución de dudas previo al examen final convocatoria EXTRAORDINARIA junio.
06/06/2024	Repaso y resolución de dudas.
13/06/2024	Repaso y resolución de dudas.
17-20 junio	PRUEBA FINAL EXTRAORDINARIA (fecha a determinar en la semana del 17-20 junio 2024)

PLANIFICACIÓN DE TAREAS DE EVALUACIÓN ENTREGABLES

- UT1, UT2, UT3, UT4 – Fecha límite de entrega **25 enero 2024**.
- UT5, UT6, UT7 – Fecha límite de entrega **2 mayo 2024**.

7. PRUEBAS PRESENCIALES DE EVALUACIÓN

- **8 de febrero de 2024.** Prueba **presencial parcial** en febrero. (Voluntaria para el alumnado que la solicite).
- **Marzo 2024** (fecha a determinar). Prueba **presencial final ORDINARIA** para aquellos alumnos/as que hayan solicitado ser evaluados como **PENDIENTES**. (Voluntaria para el alumnado que la solicite).
- **16 de mayo de 2024.** Prueba presencial final ORDINARIA. (Obligatoria para todo el alumnado)
- **Junio 2024** (fecha a determinar entre el 17-20). Prueba presencial **final EXTRAORDINARIA**. (Obligatoria si no se ha superado el módulo en la convocatoria ordinaria)