



GUÍA DOCENTE 2023-2024  
**Programación Concurrente y de Tiempo Real**

**1. Denominación de la asignatura:**

PROGRAMACIÓN CONCURRENTE Y DE TIEMPO REAL

**Titulación**

Grado en Ingeniería Informática

**Código**

8821

**2. Materia o módulo a la que pertenece la asignatura:**

Programación

**3. Departamento(s) responsable(s) de la asignatura:**

Ingeniería Informática y DIGITALIZACIÓN

**4.a Profesor/a que imparte la docencia (Si fuese impartida por más de uno/a incluir todos/as) :**

Carlos López (LSI)/Rubén Ruiz González (ISA)

**4.b Coordinador/a de la asignatura**

Carlos López (LSI)/Rubén Ruiz González (ISA)

**5. Curso y semestre en el que se imparte la asignatura:**

3 Curso 6 semestre

**6. Tipo de la asignatura: (Básica, obligatoria u optativa)**

Obligatoria



**7. Requisitos de formación previos para cursar la asignatura:**

Se recomienda tener superadas las asignaturas: Metodología de la Programación, Estructuras de Datos, Arquitecturas Paralelas.

**8. Número de créditos ECTS de la asignatura:**

6

**9. Competencias que debe adquirir el/la estudiante al cursar la asignatura**

Competencias transversales de la materia de programación  
CT1, CT2,CT3,CT4\*,CT5,CT6,CT7,CT8,CT9,CT10,CT12\*,CT14\*,CT16,CT18,CT22,CT25,CT26,CT27

\* Competencia/Contenido de sostenibilización curricular

Competencias genéricas de la materia de programación

CG1,CG3\*,CG4,CG5,CG6,CG8\*,CG9,CG10

\* Competencia/Contenido de sostenibilización curricular

Competencias específicas

CR14: Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación

paralela, concurrente, distribuida y de tiempo real.

**10. Programa de la asignatura**

**10.1- Objetivos docentes**

Demostrar los problemas principales de ejecución separada por múltiples tareas.

Dependencias de estados y exclusión mutua.

Conocer los conceptos fundamentales ligados a la programación concurrente.

Conocer los mecanismos y técnicas fundamentales de sincronización.

Dominar un conjunto de problemas tipo asociados a la programación concurrente.

Conocer los conceptos fundamentales de los sistemas en tiempo real.

Dominar la programación y planificación en tiempo real.

**10.2- Unidades docentes (Bloques de contenidos)**



## **Unidad A: Programación concurrente**

### **Conceptos básicos**

- Introducción
- Estructuras básicas de soporte a la concurrencia en Java  
Threads, ciclo de vida, prioridades, interrupciones, estados, ejecutores
- Objetos y concurrencia  
Concurrencia, estructuras para la ejecución concurrente, concurrencia y programación OO, transformaciones y modelos de objetos
- Imposiciones de diseño concurrente  
Seguridad vs. vivacidad, rendimiento vs. reutilización

### **Exclusión mútua**

- Introducción
- Inmutabilidad  
Aplicaciones y construcción
- Sincronización  
Formas de conseguir sincronización, objetos completamente sincronizados, estrategias de recorrido en colecciones, singleton, interbloqueo, atomicidad
- Confinamiento  
A través de métodos, dentro de hilos, de objetos, de grupos
- Patrones de diseño:

### **Dependencias de estados**

- Introducción
- Cómo enfrentarse a fallos  
Excepciones, Cancelación
- Métodos protegidos  
Suspensión protegida, funcionamiento del monitor, esperas con protección y notificaciones, más tipos de esperas y notificaciones
- Uso de bibliotecas Java
- Patrones de diseño concurrentes  
Productor – Consumidor, Bloqueo de lectura y escritura

## **Unidad B: Programación en tiempo real**

### **Introducción a los sistemas de tiempo real**

- Conceptos básicos
- Características y propiedades de los STR
- Tareas y ejecución en los STR
- Tipos de STR
- Ejemplos de STR



### **Programación de tiempo real**

- Plataformas de ejecución de un STR
- Modalidades de programación de los STR (hardware, firmware, software, RTOS, etc.)
- Lenguajes de programación con soporte a los STR
- Programación de STR apoyada en sistemas operativos de tiempo real (RTOS)

### **Planificación de tiempo real**

- Conceptos básicos
- Ejecución de tareas en tiempo real
- Cronogramas temporales
- Planificabilidad de un STR
- Tipos de planificadores en STR (estáticos, cíclicos, prioridades, EDF, etc.)

### **10.3- Bibliografía**

#### **BIBLIOGRAFÍA BÁSICA**

Alan Burns y Andy Wellings, (2003) Sistemas de Tiempo Real y Lenguajes de Programación, 3ª, Addison Wesley (Pearson),

Doug Lea, (2001) Programacion Concurrente Con Java, 2ª, Sun Microsystems y Addison Wesley, 8478290389,

#### **BIBLIOGRAFÍA COMPLEMENTARIA**

Cay S. Horstmann y Gary Cornell. , (2008) Core Java, Vol. 2: Advanced Features, 8, Prentice Hall,

Jane W. S. Liu, (2000) Real-Time Systems, 1, Prentice Hall,

M. Ben-Ari, (1990) Principles of Concurrent and Distributed Programming, Prentice Hall,

Mark Grand, (2002) Patterns in Java, Volume 1, A Catalog of Reusable Design Patterns Illustrated with UML, Segunda, Wiley, 978-0-471-22729-8,

Oracle, (2012) Tutorial Java Concurrente, Oracle and/or its affiliates.,

<http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>.

Sang H. Son , Insup Lee y Joseph Y-T. Leung, (2007) Handbook of Real-Time and Embedded Systems, Chapman and Hall/CRC,

### **10.4- Acceso a la Bibliografía Recomendada desde Leganto**

[https://leganto.ubu.es/leganto/public/34BUC\\_UBU/lists?courseCode=8821&auth=SAML](https://leganto.ubu.es/leganto/public/34BUC_UBU/lists?courseCode=8821&auth=SAML)



**11. Metodología de enseñanza y aprendizaje y su relación con las competencias que debe adquirir el/la estudiante:**

| Metodología                                 | Competencia relacionada                                                                                                     | Horas presenciales | Horas de trabajo | Total de horas |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------------------|------------------|----------------|
| Clases teóricas                             | CT1, CT2,CT3,CT4, CT5,CT6,CT7,CT8,CT9,CT10,CT12,CT14, CT16,CT18,CT22,CT25,CT26,CT27,CG1, CG3,CG4,CG5,CG6, CG8,CG9,CG10,CR14 | 24                 | 0                | 24             |
| Clases prácticas                            | CT1, CT2,CT3,CT4, CT5,CT6,CT7,CT8,CT9,CT10,CT12,CT14, CT16,CT18,CT22,CT25,CT26,CT27,CG1, CG3,CG4,CG5,CG6, CG8,CG9,CG10,CR14 | 24                 | 0                | 24             |
| Estudio personal                            | CT1, CT2,CT3,CT4, CT5,CT6,CT7,CT8,CT9,CT10,CT12,CT14, CT16,CT18,CT22,CT25,CT26,CT27,CG1, CG3,CG4,CG5,CG6, CG8,CG9,CG10,CR14 | 0                  | 48               | 48             |
| Realización de trabajos, Informes, Memorias | CT1, CT2,CT3,CT4, CT5,CT6,CT7,CT8,CT9,CT10,CT12,CT14, CT16,CT18,CT22,CT25,CT26,CT27,CG1, CG3,CG4,CG5,CG6, CG8,CG9,CG10,CR14 | 0                  | 48               | 48             |
| Pruebas de evaluación                       | CT1, CT2,CT3,CT4, CT5,CT6,CT7,CT8,CT9,CT10,CT12,CT14, CT16,CT18,CT22,CT                                                     | 6                  | 0                | 6              |



|                                                                |    |    |     |
|----------------------------------------------------------------|----|----|-----|
| 25,CT26,CT27,CG1,<br>CG3,CG4,CG5,CG6,<br>CG8,CG9,CG10,CR1<br>4 |    |    |     |
| <b>Total</b>                                                   | 54 | 96 | 150 |

## 12. Sistemas de evaluación:

La evaluación será continua atendiendo al Reglamento de evaluación de la Universidad de Burgos. A tal efecto se presentan dos opciones de evaluación: Modelo normal y Modelo excepcional.

Modelo normal. El artículo 10 del Reglamento de evaluación en su punto 10.2 establece que en la primera convocatoria se calificará al estudiante de acuerdo a los resultados de la evaluación continua.

Para superar la asignatura es requisito haber superado los siguientes procedimientos de evaluación:

- evaluación continua de prácticas de laboratorio de programación concurrente 25% (nota de corte 5 sobre 10)
- prueba final escrita de teoría de programación concurrente 25 % (nota de corte 5 sobre 10)
- evaluación continua de prácticas de laboratorio en programación en tiempo real 25% (nota de corte 5 sobre 10)
- prueba final escrita de teoría de programación en tiempo real 25 % (nota de corte 5 sobre 10)

Respecto de la segunda convocatoria, el artículo 10 del Reglamento de evaluación, en su punto 10.2, indica que todas las pruebas podrán recuperarse en la segunda convocatoria y en esta convocatoria no podrán exigirse aquellas pruebas superadas ya en la primera.

En el caso de los estudiantes que participen en el programa Universitario Cantera, la calificación se determinará en función del desempeño de las tareas que les sean asignadas en el marco del programa.

El sistema de evaluación para estudiantes de intercambio deberá ser modificado en el supuesto de que los calendarios académicos de las universidades de origen y de destino no sean coincidentes.



| Procedimiento                                                                  | Peso primera convocatoria | Peso segunda convocatoria |
|--------------------------------------------------------------------------------|---------------------------|---------------------------|
| Evaluación continua de prácticas de laboratorio de programación concurrente    | 25 %                      | 25 %                      |
| Prueba escrita de teoría de programación concurrente                           | 25 %                      | 25 %                      |
| Evaluación continua de prácticas de laboratorio de programación de tiempo real | 25 %                      | 25 %                      |
| Prueba escrita de teoría de programación de tiempo real                        | 25 %                      | 25 %                      |
| <b>Total</b>                                                                   | <b>100 %</b>              | <b>100 %</b>              |

#### **Evaluación excepcional:**

Modelo excepcional. Atendiendo a lo expuesto en el artículo 9 del Reglamento de Evaluación en su punto 9.1, los estudiantes que, por razones excepcionales, no puedan seguir los procedimientos habituales de evaluación continua deberán solicitar por escrito al Decano o Director de Centro acogerse a una «evaluación excepcional». Dicho escrito con las razones que justifiquen la imposibilidad de seguir la evaluación continua deberá presentarse antes del inicio del semestre lectivo o durante las dos primeras semanas de impartición de la asignatura.

Esta forma de evaluación implica los mismos requisitos que el modelo normal con la salvedad de la presencialidad, que deberá sustituirse con la asistencia a tutorías programadas con el profesor.

Para superar la asignatura es requisito haber superado los siguientes procedimientos de evaluación:

- evaluación continua de prácticas de laboratorio de programación concurrente 25% (nota de corte 5 sobre 10)
- prueba final escrita de teoría de programación concurrente 25 % (nota de corte 5 sobre 10)
- evaluación continua de prácticas de laboratorio en programación en tiempo real 25% (nota de corte 5 sobre 10)
- prueba final escrita de teoría de programación en tiempo real 25 % (nota de corte 5 sobre 10)

#### **13. Recursos de aprendizaje y apoyo tutorial:**

Documentación de la asignatura  
Pizarra y Proyector  
Páginas Webs relacionadas  
Bibliografía disponible en la Biblioteca



Aplicaciones interactivas en la Plataforma UBUvirtual  
Tutorías individualizadas o en grupo a demanda de los alumnos

**14. Calendarios y horarios:**

El calendario y horarios de EPS. Consultar <https://www.ubu.es/eps>

**15. Idioma en que se imparte:**

Español con bibliografía en inglés