

Asignatura: **Sistemas Informáticos**

Código: 10-09817

RTF

8

Semestre: 9

Carga Horaria

80

Bloque: Tecnologías Aplicadas

Horas de Práctica

48

Departamento: Computación

Correlativas:

- Seguridad informática

Contenido Sintético:

- Gestión de la configuración.
- Técnicas para la escalabilidad e interoperabilidad.
- Técnicas para la tolerancia a fallas y robustez.
- Gestión de la infraestructura.
- Gestión del proceso de construcción y despliegue.
- Arquitecturas de grandes datos.
- Licenciamiento y distribución de software.

Competencias Genéricas:

- CG1: Identificar, formular y resolver problemas de ingeniería.
- CG2: Concebir, diseñar y desarrollar proyectos de ingeniería (sistemas, componentes, productos o procesos).
- CG3: Gestionar -planificar, ejecutar y controlar- proyectos de ingeniería (sistemas, componentes, productos o procesos).
- CG4: Utilizar de manera efectiva las técnicas y herramientas de aplicación en ingeniería.
- CG5: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- CG10: Actuar con espíritu emprendedor.

Aprobado por HCD: 1042-HCD-2023

Res: Fecha: 27/11/2023

Competencias Específicas:

- CE1.4 Implementar y mantener sistemas informáticos.

Presentación

Sistemas Informáticos es una asignatura del último año (noveno semestre) de la carrera de Ingeniería en Computación. Dado el esquema de correlatividades, al momento de transitar esta asignatura los estudiantes han cursado todas las asignaturas fundamentales relativas a la programación (algoritmos y estructuras de datos), así como también aquellas centradas en los aspectos de la ingeniería de software, el aseguramiento de su calidad, y la seguridad de los sistemas informáticos. Esta asignatura complementa la formación antes mencionada incorporando dimensiones asociadas a la gestión del proceso de desarrollo de sistemas, el diseño de los mismos con la mira puesta en la integración con otros sistemas, su eficiencia computacional, su robustez, y en su despliegue en entornos productivos.

En este sentido, el objetivo del curso es facilitar herramientas tanto conceptuales como tecnológicas para dar cuenta de:

- Una gestión adecuada de los artefactos que componen un sistema informático y su proceso de desarrollo y construcción.
- Una gestión eficiente de la infraestructura para la ejecución de sistemas y el despliegue de sistemas en la misma.
- El diseño de sistemas informáticos que se ejecutan de forma eficiente, robusta y escalable frente a una demanda creciente y fallas externas.
- El diseño de sistemas informáticos que se integran de forma eficaz con otros sistemas.
- Las características y tecnologías necesarias para el desarrollo de sistemas que manejan grandes volúmenes de datos.
- Las estrategias y formas de licenciamiento para la distribución de sistemas informáticos.

Esta asignatura corona los conocimientos necesarios para desarrollar sistemas informáticos que se implementan en entornos productivos realistas. De esta manera, contribuye a desarrollar competencias fundamentales para que los estudiantes puedan desempeñarse efectivamente en el ámbito laboral relacionado al desarrollo de sistemas informáticos.

Teniendo en cuenta que las dimensiones abordadas en esta asignatura están relacionadas a un ejercicio de la profesión que suele desarrollarse en equipo, la asignatura propone casos de estudio realistas y trabajos prácticos a desarrollar en grupo, donde la discusión, la formulación de soluciones acordadas y la gestión de las mismas en equipo es parte intrínseca de las competencias que se espera desarrollar.

Contenidos

Unidad 1: Gestión de la Configuración

Introducción a la gestión de la configuración de sistemas informáticos. Conceptos básicos de control de versiones. Herramientas de control de versiones. Administración de cambios y versionado de software. Prácticas recomendadas en la gestión de configuración.

Unidad 2: Técnicas para la Escalabilidad e Interoperabilidad

Escalabilidad de sistemas informáticos: conceptos y estrategias. Interoperabilidad entre sistemas y aplicaciones. Uso de estándares y protocolos para mejorar la interoperabilidad.

Técnicas de diseño orientado a servicios. Prácticas recomendadas para la escalabilidad e interoperabilidad.

Unidad 3: Técnicas para la Tolerancia a Fallas y Robustez

Tolerancia a fallas en sistemas informáticos. Estrategias de redundancia y alta disponibilidad. Diseño de sistemas robustos y resistentes. Pruebas de resistencia y recuperación ante desastres. Monitoreo y detección de fallas en tiempo real.

Unidad 4: Gestión de la Infraestructura

Automatización de la infraestructura. Despliegue y gestión de máquinas virtuales y contenedores. Administración de recursos en la nube. Orquestación de servicios y aplicaciones en la nube. Políticas de seguridad y acceso a la infraestructura.

Unidad 5: Gestión del Proceso de Construcción y Despliegue

Introducción a la integración continua (CI) y entrega continua (CD). Automatización de pruebas de software. Despliegue automatizado en diferentes entornos (desarrollo, prueba, producción). Estrategias de rollback y rollforward. Herramientas para CI/CD.

Unidad 6: Arquitecturas de Grandes Datos

Fundamentos de arquitecturas de grandes datos. Tecnologías y plataformas para el procesamiento de grandes volúmenes de datos. Almacenamiento y análisis distribuido de datos. Desafíos y consideraciones de seguridad en arquitecturas de grandes datos.

Unidad 7: Licenciamiento y Distribución de Software

Conceptos de licenciamiento de software. Tipos de licencias y su impacto en la distribución. Estrategias de licenciamiento abierto y comercial. Cumplimiento de licencias y gestión de activos de software. Tendencias actuales en licenciamiento de software.

Metodología de enseñanza

El curso se estructura en base a dos clases semanales de naturaleza teórico-práctica, cada una con una duración de 2,5 horas. La metodología de enseñanza propuesta se apoya en el Aprendizaje Basado en Problemas (ABP), el análisis de casos prácticos realistas, y la utilización del aula virtual como recurso esencial complementario a las clases presenciales.

Los estudiantes trabajan a partir de los conceptos y herramientas presentados por el docente en el aula y los materiales disponibles en el entorno virtual de aprendizaje, que incluyen videos, lecturas y ejercicios prácticos, abarcando tanto aspectos teóricos como ejemplos concretos. El aula virtual facilita el acceso a estos recursos y permite a los alumnos establecer un ritmo de aprendizaje personalizado, independiente del que se imparte en las sesiones presenciales.

En las clases presenciales, el docente introduce los conceptos teóricos y prácticos clave, así como coordina la exploración de casos de estudio que presentan diferentes problemáticas. Esta actividad permite a los estudiantes aplicar y contextualizar los conceptos previamente aprendidos. Además de los casos, el docente plantea una serie de preguntas conceptuales

que fomentan la discusión y el análisis de posibles soluciones entre los estudiantes. El objetivo es profundizar en los conceptos fundamentales, ayudando a los alumnos a comprender la aplicación adecuada de soluciones y a identificar y debatir errores comunes.

Durante las clases presenciales, se brinda orientación y se resuelven dudas relacionadas con los trabajos prácticos, abordando los problemas específicos planteados por estos. Los estudiantes trabajan en grupos para discutir y buscar soluciones, con el docente ejerciendo como guía y facilitador. Al finalizar cada trabajo práctico, los grupos presentan sus soluciones, generando así un espacio de retroalimentación y discusión colectiva que enriquece el proceso de aprendizaje a través del intercambio de experiencias.

Además de las clases semanales, se fomenta el debate en torno a conceptos y problemas de los trabajos prácticos mediante un foro disponible en el aula virtual. Este espacio permite una comunicación continua pero asincrónica entre docentes y estudiantes, promoviendo un aprendizaje colaborativo y constante.

Evaluación

La evaluación se basa en un enfoque continuo e integrador que refleja su naturaleza progresiva. El propósito de esta metodología de evaluación es no solo medir el conocimiento adquirido, sino también fomentar una comprensión profunda y aplicada de los conceptos y habilidades aprendidos. En este contexto, la evaluación se lleva a cabo a lo largo del período del curso mediante tres evaluaciones parciales presenciales y cuatro trabajos prácticos.

Cada evaluación parcial consta de preguntas teóricas y análisis de casos prácticos, lo que garantiza una evaluación integral de los conocimientos y habilidades adquiridos hasta ese momento. Está prevista una evaluación hacia la mitad del curso y otra hacia la finalización del mismo. La tercera evaluación parcial, también prevista hacia el final del curso, sirve como instancia de recuperación para cualquiera de las dos evaluaciones parciales anteriores.

Los resultados principales de los trabajos prácticos se presentan de manera oral por parte de los grupos de estudiantes durante las clases, tal como se menciona en el apartado anterior. Además, los estudiantes deben entregar los trabajos prácticos a los docentes para su evaluación detallada. En caso de ser necesario, los trabajos prácticos pueden ser corregidos y reenviados para su revisión.

Condiciones de aprobación

Condiciones de regularización

Para alcanzar la condición de regular, el estudiante debe cumplir las siguientes condiciones:

1. Asistir al menos al 80% de las clases.
2. Alcanzar un rendimiento igual o superior a 60% en cada uno de los Trabajos Prácticos (TP).
3. Alcanzar un rendimiento igual o superior a 60% en dos de las Evaluaciones Parciales (EP).
4. Alcanzar el criterio de aceptación mínimo para los resultados de aprendizaje propuestos.

Condiciones de promoción

Para alcanzar la promoción, el alumno debe cumplir las siguientes condiciones:

1. Asistir al menos al 80% de las clases.
2. Alcanzar un rendimiento igual o superior a 80% en cada uno de los Trabajos Prácticos (TP).
3. Alcanzar un rendimiento igual o superior a 80% en al menos dos de las Evaluaciones Parciales (EP).
4. Alcanzar el criterio de aceptación mínimo para los resultados de aprendizaje propuestos.

Cumplidas estas condiciones, la nota final de promoción se calculará según la siguiente fórmula:

$$\text{Calificación} = \text{redondear}((0,5 * \text{TPs} + 0,5 * \text{EPs}) / 10)$$

donde TPs es el resultado de promediar el rendimiento de todos los trabajos prácticos, y EPs es el resultado de promediar el rendimiento de las dos evaluaciones parciales con mayor rendimiento.

Examen final

Los estudiantes regulares rinden un examen escrito individual con características similares a las evaluaciones parciales. La calificación final es el rendimiento obtenido en el examen.

Los estudiantes libres rinden en dos etapas. En la primera, presentan y defienden de forma oral los trabajos prácticos. Superada esta etapa, rinden un examen escrito individual equivalente a los estudiantes regulares.

Actividades prácticas y de laboratorio

Las actividades prácticas se desarrollan a partir de trabajos prácticos, que se presentan durante las clases presenciales, y cuyos recursos se disponibilizan en el aula virtual. Los trabajos prácticos abordan principalmente temas de diferentes unidades conceptuales:

- TP1: unidad 1
- TP2: unidades 2 y 3
- TP4: unidades 4 y 5
- TP5: unidades 6 y 7

Sin embargo, algunos aspectos de las unidades son transversales y se integran a todos los trabajos prácticos.

Los trabajos prácticos plantean situaciones y problemáticas a resolver en torno al desarrollo de un proyecto de software. Se espera que se simule un equipo de desarrollo y por lo tanto, los trabajos prácticos se diseñan para que sean trabajados en grupos.

Desagregado de competencias y resultados de aprendizaje

Los siguientes son las competencias esperables como resultado del aprendizaje, organizadas de acuerdo a las unidades de contenido:

1. Conocer los fundamentos de la gestión de la configuración en sistemas informáticos.
2. Comprender y aplicar conceptos básicos de control de versiones.
3. Utilizar herramientas de control de versiones de manera efectiva.
4. Evaluar y comparar diferentes enfoques de administración de cambios y versionado de software
5. Diseñar un sistema de gestión de configuración basado en prácticas recomendadas.

6. Definir y aplicar estrategias de escalabilidad en sistemas informáticos.
7. Diseñar sistemas orientados a servicios para mejorar la escalabilidad e interoperabilidad.
8. Aplicar estándares y protocolos para mejorar la interoperabilidad.
9. Aplicar técnicas de diseño orientado a servicios.
10. Identificar y aplicar prácticas recomendadas para la escalabilidad y la interoperabilidad.

11. Comprender el concepto de tolerancia a fallas en sistemas informáticos.
12. Diseñar estrategias de redundancia y alta disponibilidad.
13. Desarrollar sistemas robustos y resistentes a fallas.
14. Realizar pruebas de resistencia y planificar la recuperación ante desastres.
15. Implementar sistemas de monitoreo y detección de fallas en tiempo real.

16. Automatizar tareas relacionadas con la infraestructura de sistemas informáticos
17. Desplegar y administrar máquinas virtuales y contenedores de manera eficiente.
18. Gestionar recursos en entornos de nube de forma efectiva.
19. Orquestar servicios y aplicaciones en la nube.
20. Establecer políticas de seguridad y acceso para la infraestructura.

21. Comprender los conceptos de integración continua (CI) y entrega continua (CD).
22. Automatizar pruebas de software para garantizar la calidad del código.
23. Desplegar de manera automatizada en diferentes entornos.
24. Utilizar herramientas de CI/CD para optimizar el proceso de construcción y despliegue.

25. Comprender los fundamentos de las arquitecturas de grandes datos.
26. Evaluar y seleccionar tecnologías y plataformas para el procesamiento de grandes volúmenes de datos.
27. Diseñar soluciones de almacenamiento y análisis distribuido de datos.
28. Abordar desafíos y consideraciones de seguridad en arquitecturas de grandes datos.
29. Aplicar prácticas recomendadas en el manejo de grandes conjuntos de datos.

30. Comprender los conceptos fundamentales de licenciamiento de software.
31. Analizar los tipos de licencias y su impacto en la distribución de software.
32. Evaluar estrategias de licenciamiento tanto en el ámbito abierto como comercial.
33. Estar al tanto de las tendencias actuales en licenciamiento de software.

La siguiente matriz desagrega las competencias genéricas y específicas en los resultados anteriores:

	CG1	CG2	CG3	CG4	CG5	CG10	CE1.4
1			x	x		x	
2			x	x		x	
3			x	x		x	
4			x	x		x	
5			x	x		x	
6	x	x		x	x	x	x
7	x	x		x	x	x	x
8	x	x		x	x	x	x
9	x	x		x	x	x	x
10	x	x		x	x	x	x
11	x	x		x	x	x	x
12	x	x		x	x	x	x
13	x	x		x	x	x	x
14	x	x		x	x	x	x
15	x	x		x	x	x	x
16	x		x	x		x	
17	x		x	x		x	
18	x		x	x		x	
19	x		x	x		x	
20	x		x	x		x	
21	x		x	x		x	
22	x		x	x		x	
23	x		x	x		x	
24	x		x	x		x	
25	x	x		x	x	x	x

26	x	x		x	x	x	x
27	x	x		x	x	x	x
28	x	x		x	x	x	x
29	x	x		x	x	x	x
30					x	x	
31					x	x	
32					x	x	
33					x	x	
34					x	x	

Bibliografía

- Chacon, S., & Straub, B. (2014). Pro Git (2a. ed.). [Berkeley, California]: Apress.
- Nygard, M. T. (2007). Release It! Design and Deploy Production-Ready Software. Raleigh, NC: Pragmatic Bookshelf. ISBN: 978-0-9787-3921-8
- Humble, J., Farley, D. G. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Upper Saddle River, NJ: Addison- Wesley. ISBN: 978-0-321-60191-9
- Kim, G., Humble, J., Debois, P., Willis, J., & Allspaw, J.. (2016). The devOps handbook :how to create world-class agility, reliability, and security in technology organizations . Portland, OR :IT Revolution Press
- Jacobson, D., Brail, G., & Woods, D. (2011). APIs: A Strategy Guide.
- Marz, N., Warren, J. (2015). Big data: principles and best practices of scalable real- time data systems. Manning. ISBN: 9781617290343 1617290343
- Stallman R. & Free Software Foundation (Cambridge Mass). (2002). Free software free society : selected essays of Richard M. Stallman. Free Software Foundation.