

PROGRAMA DE METODOLOGÍA DE DESARROLLO DE SISTEMAS EN HARDWARE Y SOFTWARE

Carrera: Ingeniería en Automatización y Control Industrial.

Asignatura: Metodología de Desarrollo de Sistemas en Hardware y Software.

Núcleo al que pertenece: Núcleo Superior Orientación

Profesoras/es: A Designar*

Asignaturas previas necesarias para favorecer el aprendizaje: Diseño basado en Microcontroladores, Organización y Arquitectura de Computadores

Objetivos:

Que el estudiante:

- Entienda que para llevar a cabo un proyecto de desarrollo de hardware y software hace falta llevar a cabo varias actividades además de programar y armar un prototipo funcional, y tenga una noción de cuáles son estas actividades y las técnicas asociadas a cada una.
- Comprenda la importancia de la planificación proyecto y su documentación asociada.
- Conozca el concepto de metodología como definición de las actividades que involucra el desarrollo de hardware y software, su articulación y los roles que ocupan las personas que participan. Los conceptos principales asociados a metodologías en cascada, en V, ágiles y estructuradas. Las actividades y roles que involucran, y algunas similitudes y diferencias entre los distintos enfoques.
- Comprenda la relevancia de los distintos tipos de testing existentes y el alcance de cada uno de ellos.

- Incorpore la importancia de normas y estándares, cómo acceder a los mismos.
- Pueda interpretar requerimientos, y tenga noción de las actividades asociadas a tareas de diseño conjunto de hardware y software, simulación, y verificación para lograr la concreción y verificación exitosa de un proyecto.
- Practique aplicando las actividades y metodologías que se describen en la asignatura.

Contenidos mínimos:

Requerimientos: especificación inicial y seguimiento. Procesos de desarrollo y sus modelos: en cascada, en V y derivados. Documentación del desarrollo: plan y protocolos de prueba, especificación de hardware y software, diseño, arquitectura del proyecto, UML, análisis de riesgo. Tipos de pruebas: funcional, del sistema, unitarias. Testing, integración y entregas continuas. Guías de buenas prácticas para codificación del software. Estándares. Metodologías para el diseño conjunto de hardware y software, simulación, y verificación. Herramientas generales para el desarrollo: repositorios, plan de proyecto, seguimiento de problemas.

Carga horaria semanal: 4 horas.

Programa analítico:

Unidad 1: Introducción general.

- Surgimiento y problemáticas del desarrollo de hardware y software.
- Evolución de la computación y los sistemas embebidos.
- Introducción a las tecnologías de sistemas embebidos.

Unidad 2: Proyectos.

- Proyectos de ingeniería de hardware y software.
- Introducción a la gestión de proyectos.

Unidad 3: Procesos de desarrollo y sus modelos.

- Modelos de ciclo de vida: Cascada, evolutivo, incremental, modelo V.
- Metodologías ágiles. Scrum. El manifiesto ágil.

- Requisitos. Trazabilidad.

Unidad 4: Co-diseño de Software y Hardware.

- Flujo de diseño: Especificación del Sistema. Simulación Funcional. Exploración del Espacio de Diseño. Particionado Software y Hardware. Co-Simulación. Prototipado.
- Herramientas de Software de soporte para el codiseño de Software y Hardware.
- Opciones de implementación: microcontrolador, DSP, FPGA y SoC.

Unidad 5: Gestión de riesgos.

- Ciclo de la gestión de riesgo: Análisis, evaluación, control, reducción, seguimiento.
- Análisis de riesgo en el Software y Hardware.
- Seguridad.

Unidad 6: Técnicas de Verificación y Validación.

- Verificación y Validación de Software y Hardware.
- Herramientas.
- Análisis estático.
- Indicadores.
- Plan de pruebas.
- Diseño de casos de prueba.

Unidad 7: Certificaciones y ensayos.

- Normas y estándares, conformidad, proceso de normalización.
- Certificación, clases de certificación.
- Certificación en el campo voluntario.
- Certificación en el campo obligatorio. Regulaciones.
- Ensayos para certificación en el campo regulado.

Bibliografía obligatoria:

- Ingeniería de Software. (9na edición) Sommerville, Ian. Pearson Educación, México, 2011.
- Ingeniería de Software. Un enfoque práctico. (5ta edición) Pressman, Roger. McGRAW-HILL.

- A practical introduction to hardware/software codesign. Springer Science & Business Media. Schaumont, P. (2012).
- Software Engineering for Embedded Systems: Methods, Practical Techniques and Applications. Oshana R. (2013).
- Designing Concurrent, Distributed, and Real-Time Applications with UML (Addison-Wesley Object Technology), Hassan Gomaa.
- Real Time UML: Advances in the UML for Real-Time Systems (3rd Edition). Bruce Powel Douglass.
- Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems, Bruce Powel Douglass.
- The Art of Agile Development, Shore & Warden.
- Construcción de Software: una mirada ágil, Paez, Fontdevila, Suárez.

Bibliografía de consulta: *NO INDICA_

Organización de las clases:

Las clases serán en modalidades teórica, práctica y teórico-práctico dependiendo del tema a desarrollar. En las clases teóricas se profundizará con un material de lectura y en algunos casos con cuestionarios a realizar. Durante el curso se trabajará con los alumnos en un trabajo práctico grupal con entregas incrementales, donde tendrán oportunidad de poner en práctica los temas desarrollados durante el curso de la materia. El trabajo finalizará con una exposición de cada grupo frente al resto de sus compañeros.

Modalidad de evaluación:

Aprobación de la asignatura según Régimen de Estudios vigente de la Universidad Nacional de Quilmes (Res. CS N° 201/18): habrá un trabajo práctico final grupal con entregas incrementales y una evaluación integradora al terminar el cuatrimestre. La ponderación de las notas para la nota final será:

Actividad evaluadora	% de nota
Trabajo práctico Grupal	50 %

Evaluación integradora	50 %
------------------------	------

Modalidad de evaluación para exámenes libres:

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito y/u oral e instancias de evaluación similares a las realizadas en la modalidad presencial. Se evaluarán todos los contenidos especificados en el programa analítico.

ANEXO II

CRONOGRAMA TENTATIVO

Sema na	Tema/Unidad	Actividad				Evalua ción
		Teóri co	Práctico			
			Res. Prob.	Lab.	Otros	
1	Unidad 1: Introducción general.	x				
2 y 3	Unidad 2: Proyectos.	x				
4	Comienzo de trabajos grupales.		x			
4	Unidad 3: Procesos de desarrollo y sus modelos.	x				
5	Unidad 3: Procesos de desarrollo y sus modelos.	x			Dinám ica	
6	Unidad 4: Flujo de diseño: Especificación del Sistema. Simulación Funcional. Exploración del Espacio de Diseño. Particionado Software y Hardware. Co-Simulación. Prototipado.	x		x		
7	Unidad 4: Herramientas de Software de soporte para el co-diseño de Software y Hardware.	X		x		
8	Unidad 4: Opciones de implementación: microcontrolador, DSP, FPGA y SoC.	X		x		
8	Seguimiento de trabajos grupales.		x			
9	Unidad 5: Gestión de riesgos.	x				
10	Unidad 6: Técnicas de Verificación y Validación.	x	x			
11	Unidad 7: Certificaciones y ensayos.	x				
11	Seguimiento de trabajos grupales.		x			
12	Seguimiento de trabajos grupales.		x			
13	Seguimiento de trabajos grupales.		x			
14	Seguimiento de trabajos grupales.		x			
15	- Exposición de trabajos grupales.				Expos ición	x
16	- Evaluación integradora.					x
17	Actividad retrospectiva.				Dinám ica	