

Carrera: Especialización en Sistemas Embebidos

Curso de Posgrado: “Ingeniería de Software en Sistemas Embebidos”

Carga Horaria¹: 30 Hs.

Docente/s a cargo: Mgt Bioing. Analía Cherniz (FIUNER)

Ing. Víctor Valotto (FIUNER)

Semestre: 1º

Año: 2019

Modalidad²: Curso teórico-práctico

Carácter³: obligatorio

1:

Carga horaria: la cantidad de horas reloj.

2:

Curso teórico: curso donde se desarrolla en forma expositiva una temática propia de la disciplina

Curso teórico-práctico: curso que articula la modalidad del curso teórico con una actividad de la práctica con relación a la temática de estudio. Lo teórico y lo práctico se dan simultáneamente en forma interrelacionada.

3:

Carácter: si son obligatorios u optativos.

Programa Analítico: de foja: 1 a foja: 2

Bibliografía de foja: 2 a foja: 2

Programa Analítico

Introducción y procesos

Software: definiciones, instrucciones, estructuras, documentos, productos y proyectos. Ingeniería del Software: definiciones y alcances, etapas. Marco de procesos. Ciclo de vida: modelos. Métodos Ágiles: SCRUM

Requerimientos de Software

Definiciones. Niveles. Requerimientos funcionales y no funcionales. Proceso de requerimientos: desarrollo, administración. Buenas prácticas. Herramientas de análisis. Casos de uso. Requerimientos Ágiles: Historias de Usuario, Historias de Producto. Casos de Uso Ágiles.

Arquitectura y Diseño de Software

Definiciones. Principios de diseño. Modelos. Arquitectura de Software. Definiciones. Consideración de diseño. Arquetipos. Estrategias de despliegue. Tecnologías. Atributos de calidad. Aspectos Transversales. Arquitectura y Requerimientos. Diseño Detallado. Diagrama de Robustez. Diagrama de Secuencias.

Mantenimiento de Software

Ciclo de vida del producto. Costos. Evolución: diseño basado en cambio. Bases del Diseño, principios para la evolución y mantenibilidad: Cohesión, acoplamiento y ocultamiento de la información. Principios SOLID.

Pruebas de Software

Definiciones: errores, defectos, fallas, calidad. Principios del testing. Proceso de testing. Costo. Taxonomía de las pruebas.

Ciclo de vida del testing. Diseño de casos de prueba.

Administración de la Configuración

Definiciones, funciones y conceptos. Item de Configuración. Estrategias de Branching. Modelo SVN y GIT. Conceptos de build, empaqueta y entrega. Integración Continua.

Calidad de Software

¿Qué es la calidad? Definiciones y conceptos. Aseguramiento y control de la calidad. Principios de la Calidad. Costos de la Calidad (No Calidad). Aseguramiento de la calidad. Objetivos y funciones. Defectos, fallas y revisiones.

BIBLIOGRAFIA

Bibliografía:

- Roger Pressman. Ingeniería del Software, un enfoque práctico. MCGRAW-HILL, 2005, Edición Número 6.
- Ian Sommerville. Ingeniería del Software. PEARSON ADDISON WESLEY, 2005, Séptima edición.
- Mark J. Christensen, Richard H. Thayer. The Project Manager's Guide to Software Engineering Best Practices. IEEE Press, 2001.
- The Software Engineering Book of Knowledge, IEEE, 2004.
- Merlin Dorfman y Richard H. Thayer (eds.). Software Engineering. IEEE Press, 1997.
- Richard H. Thayer (ed.). Software Engineering Project Management. IEEE Press, 2000. Segunda Edición.
- Alexander, Ian F., and Richard Stevens. Writing Better Requirements. Addison-Wesley, 2002.
- Cockburn, Alistair. Writing Effective Use Cases. Boston, MA: Addison-Wesley, 2000.
- Karl E. Wiegers. Software Requirements, Second Edition. Microsoft Press, 2003.
- Craig Larman. UML y patrones, 2da Edición. Prentice Hall, 2002.
- Grady Booch, James Rumbaugh, Iva Jacobson. El lenguaje unificado de modelado. Prentice Hall, 1999.
- SorenLauesen. User Interface Design – A Software Engineering Perspective. Pearson – Addison Wesley.
- Jesse James Garret. The Elements of User Experience, 2nd Edition. New Riders, 2011.

PLANIFICACIÓN DEL CURSO

Objetivos Generales:

El objetivo de este curso es presentar los conceptos fundamentales a la hora de enfrentarse a una solución basada en software como herramienta para las actividades relacionadas con los Sistemas Embebidos, con un enfoque de ingeniería. La propuesta de este curso está centrada en ofrecer al futuro especialista la visión de que el software es un producto que tiene un proceso de construcción que determina la calidad y el éxito de su utilidad posterior. Por lo tanto, se hace necesario transmitir al cursante la importancia de la especificación en el proceso de desarrollo de software, así como los conocimientos sobre procesos de ingeniería del software, profundizando en los aspectos de especificación a través de la

introducción de formalismos y herramientas de especificación de modelos. De forma tal que el especializando logre comprender la importancia de las buenas prácticas y que toda construcción y mantenimiento de un producto de software con calidad debe ser soportado por procesos de ingeniería..

Objetivos Particulares:

- Comprender los conceptos, elementos y estrategias básicas de la ingeniería del software.
- Aplicar los conceptos desarrollados para diseñar proyectos de software relacionados con los Sistemas Embebidos.
- Reconocer los diferentes niveles y tipo de requerimientos.
- Adquirir la destreza necesaria para realizar la elicitación de requerimientos.
- Familiarizarse con los principios de diseño y arquitectura de los proyectos de software.
- Identificar estrategias de despliegue y atributos de calidad.
- Reconocer las etapas en el ciclo de vida de un producto de software y su evolución.
- Entender los conceptos y principios del testing.
- Adquirir la destreza necesaria para diseñar casos de prueba para las diferentes etapas del ciclo de vida del testing.
- Conocer las bases conceptuales de los diferentes aspectos asociados a la calidad del software

Metodología de Trabajo:

Una vez planteados los conceptos teóricos, se propondrá una aplicación inmediata por medio de estudios dirigidos y trabajos prácticos. Se estimulará el trabajo grupal, enfatizando la tarea crítica. Las actividades prácticas se compondrán de situaciones y/o escenarios que se pueden presentar en proyectos reales, donde en general los disparadores serán elementos de análisis y discusiones para las posibles resoluciones a los problemas. Las actividades apuntan a ir evolucionando en sus especificaciones de manera que los cursantes vayan proveyendo soluciones parciales. Estas soluciones, producto de la realización de las actividades, estarán basadas en la elaboración de modelos para cada una de las fases propuestas y serán presentadas por cada grupo o estudiante. Para el desarrollo de las actividades se propondrá la utilización de herramientas de software de acceso gratuito o de uso académico.

Estas actividades tienen por objetivo generar discusiones y debates sobre los criterios de uso de las diferentes técnicas, métodos y herramientas ante diferentes situaciones. Las mismas se desarrollarán durante los horarios de clases, a fin de revisar los resultados y analizar los mismos.

Equipo docente:

Ing. Víctor Valotto

Mg. Bioing. Analía Cherniz

Cronograma del Curso:

Viernes 07/06 de 14:00 a 20:00 hs.

Presentación del curso. Introducción a la ingeniería del software, marco de procesos y ciclo de vida. Presentación de métodos ágiles. Introducción a proceso de requerimientos. (Valotto / Cherniz).

Sábado 08/06 de 9:00 a 13:00 hs.

Requerimientos funcionales y no funcionales. Herramientas de análisis y casos de uso. (Valotto / Cherniz).

Viernes 21/06 de 14:00 a 20:00 hs.

Principios y definiciones de diseño. Arquitectura de software. Estrategias de despliegue y atributos de calidad. (Valotto / Cherniz).

Sábado 22/06 de 9:00 a 13:00 hs.

Mantenimiento del software. (Valotto / Cherniz).

Viernes 05/07 de 9:00 a 18:00 hs.

Principios y procesos del testing. (Valotto / Cherniz).

Sábado 06/07 de 9:00 a 13:00 hs.

Administración de la Configuración. Introducción a la Calidad de Software. (Valotto / Cherniz).

Condiciones de Regularidad y Promoción:

Para la aprobación del curso se solicitará la resolución de actividades en donde los cursantes deberán aplicar los conceptos, técnicas y métodos trabajados durante el cursado para definir un modelo y plantear las diferentes etapas del proceso de diseño de una aplicación particular. Las actividades desarrolladas durante las clases servirán de insumo para el Trabajo Práctico. Por lo tanto, no solo se evaluará la pertinencia y calidad de la solución propuesta, sino que se realizará una valoración del proceso de desarrollo de las actividades que serán luego integradas al Trabajo Práctico, por lo que se prevé llevar a cabo una evaluación continua de las tareas realizadas por el especializando.

Infraestructura necesaria:

Laboratorio de Computación con capacidad para 15 a 20 personas. Placas de desarrollo Rasperry Pi. Una PC por alumno. Cañón proyector.