

Formulario de Aprobación Curso de Posgrado Construcción de Software

Asignatura: Construcción de Software

Profesor de la asignatura ¹:

PMP MsC Ingeniero Leticia Perez Queiruga, Grado 3, Instituto de Computación
Ingeniero Pablo Garbusi, Grado 2, Instituto de Computación

Profesor Responsable Local ¹:

Otros docentes de la Facultad:

Docentes fuera de Facultad:

Instituto ó Unidad: Instituto de Computación

Departamento ó Area: Ingeniería de Software

¹ Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

Fecha de inicio y finalización: del 17/9/2013 al 11/10/2013

Horario y Salón: Martes, Jueves y Viernes, 18:30 – 21:30

Horas Presenciales: 31

(se deberán discriminar las mismas en el ítem Metodología de enseñanza)

Nº de Créditos: 4

(de acuerdo a la definición de la UdelaR, un crédito equivale a 15 horas de dedicación del estudiante según se detalla en el ítem metodología de la enseñanza)

Público objetivo y Cupos: Interesados en el desarrollo de software. En particular en la disciplina de construcción de software. Estudiantes de la Especialización en ingeniería de software.

Este curso no tiene cupos.

(si corresponde, se indicará el número de plazas, mínimo y máximo y los criterios de selección. Asimismo, se adjuntará en nota aparte los fundamentos de los cupos propuestos. Si no existe indicación particular para el cupo máximo, el criterio general será el orden de inscripción en el Depto. de Posgrado, hasta completar el cupo asignado)

Objetivos:

Presentar al alumno los principales temas relacionados con la construcción de software, desde los elementos básicos que componen una solución de software, hasta buenas prácticas para mejorar la calidad de dicho código.

Conocimientos previos exigidos: Conocimientos de programación y de ingeniería de software

Conocimientos previos recomendados:

Metodología de enseñanza:

(comprende una descripción de las horas dedicadas por el estudiante a la asignatura y su distribución en horas presenciales -de clase práctica, teórico, laboratorio, consulta, etc.- y no presenciales de trabajo personal del estudiante)

•Horas clase (teórico): 27

•Horas clase (práctico): 2

•Horas clase (laboratorio): 0

- Horas consulta: 0
- Horas evaluación: 2
- Subtotal horas presenciales: 31
- Horas estudio: 27
- Horas resolución ejercicios/prácticos: 4
- Horas proyecto final/monografía: 0
- Total de horas de dedicación del estudiante: 62

Forma de evaluación:

Se realizará una prueba teórico/práctica en donde se evaluará el conocimiento adquirido en el curso. Dicha prueba, será sin material.

Temario:

1. Aspectos fundamentales

El objetivo de este tema es el de introducir los aspectos fundamentales del proceso de construcción de software. Entre otros, nos interesa definir qué es la construcción de software y porqué es importante.

- a. Construcción de software
Qué es y porqué es importante. Qué metáforas pueden ayudarnos a entender qué es el proceso de construcción de software.
- b. Prerrequisitos
Porqué es importante contar con los requisitos del software a construir. Determinación de los aspectos del software a construir, según diversas dimensiones (El problema a resolver, la arquitectura, los recursos disponibles, etc.)
- c. Decisiones clave
Qué decisiones afectan el proceso de construcción? Que lenguajes y herramientas elegir, en qué punto del estado tecnológico nos encontramos, etc.}
- d. Objetivos y metas
Qué objetivos perseguimos al implementar un proceso de construcción de software? Minimización de la complejidad, anticipación al cambio, construcción pensada en la verificación, reutilización, etc.
- e. Estándares en construcción
Qué estándares se utilizan en los diferentes procesos de construcción de software.

2. Aspectos prácticos

El objetivo de este tema es el de presentar aspectos prácticos a tener en cuenta en el proceso de construcción. Por ejemplo, como realizar el diseño, como trabajar con el código, que estilos de programación tenemos, etc.

- a. Diseño en el proceso de construcción
Porque nos interesa el diseño en relación con la construcción del software, que desafíos introduce y como afecta la construcción propiamente dicha del software
- b. Tipos de datos
Como se deben organizar los tipos de datos en el software, que opciones tenemos, como afecta el proceso de abstracción y encapsulamiento de código. Como se debe usar correctamente el concepto de clase e interfaz.
- c. Funciones
Extendiendo el punto anterior, ahora nos concentramos en el comportamiento. Cómo debe organizarse la funcionalidad en el software, como se arman las funciones, como se estructuran, etc.
- d. Estilos de programación

Analizamos la programación defensiva, como tratar con los errores, como preparar el código ante un problema no esperado. También analizamos el estilo de programación basado en pseudocódigos.

e. Variables

Analizamos los aspectos generales en el uso de variables, su nombrado, el concepto de scope, el concepto de binding, las diferentes convenciones de nombrado y la relación entre las variables y los tipos de datos

f. Sentencias

Analizamos los aspectos de organización del código, los diferentes tipos de sentencias, las estructuras de control. También veremos los métodos basados en tablas (table driven methods), los cuales permiten implementar en forma sencilla código complejo.

3. Mejorando el código

Luego de presentar los aspectos prácticos en la construcción de software, veremos cómo mejorar ese código construido, de forma de eliminar errores, mejorar la performance, e incluso facilitar el proceso de desarrollo

a. Calidad del software

Analizamos los aspectos que intervienen en la calidad del software, así como las técnicas para mejorar dicha calidad. También veremos la efectividad de dichas técnicas, y cuando conviene aplicar las mismas.

b. Construcción colaborativa

Analizamos diferentes prácticas para la construcción de software en forma colaborativa, como ser la programación de a pares.

c. Testing (a nivel de desarrollo)

Analizamos cómo afecta el testing en la calidad del software construido, así como las recomendaciones para hacer un testing efectivo del código. También veremos los errores típicos que se presentan a la hora de desarrollar software.

d. Depuración

Analizamos el proceso de debugging del código. Como encontrar y reparar un defecto en el código. También analizamos los factores psicológicos que intervienen en el proceso de debug de un código.

e. Refactoring

Veremos los diferentes mecanismos para evolucionar un software. Veremos como el refactoring es uno de estos mecanismos, y como existen diferentes tipos de refactoring que pueden ser llevados a cabo para extender la vida de nuestras aplicaciones.

f. Tuning de código

Presentaremos el concepto de tuning de código, cómo y qué medir en un software para determinar el nivel de performance del mismo. Luego, presentaremos las diferentes técnicas existentes para mejorar la performance de un código

Práctico: En este tema se plantea la construcción de un pequeño práctico en el que se les presentan una serie de programas, y deben en cada uno de ellos aplicar las técnicas de refactoring y tuning vistas para mejorar dichos códigos.

4. Consideraciones globales

En este tema, analizaremos aspectos de más alto nivel relacionados con la construcción de software. Nos interesan aspectos de gestión de la construcción, como el tamaño de un sistema afecta su proceso de construcción, etc.

a. Gestión de la construcción

Veremos como alentar la buena codificación, como medir diferentes indicadores del código en construcción y como estimar (en líneas generales) un calendario de construcción. También analizaremos algunos aspectos organizacionales de relacionamiento entre los programadores y/o los managers del proyecto

b. El tamaño del software

Analizaremos como el tamaño del software afecta su construcción. También veremos cual es el efecto de ese tamaño en los errores, la productividad y las actividades de desarrollo

c. Integración

Analizaremos cuando debemos integrar en el proceso de construcción, si por fases o en forma incremental. Veremos cómo esto afecta el proceso de construcción del software.

d. Herramientas

Analizaremos los tipos de herramientas que participan en el proceso de construcción. Por ejemplo, herramientas de diseño, para el manejo de código fuente, de control de código fuente, de ejecución de código, etc.

5. Consideraciones personales

En este tema, analizaremos aspectos de índole personal que afectan el proceso de construcción de software. Como las acciones que realizamos, y que escapan a lo presentado en los temas anteriores, pueden afectar el desarrollo del software.

a. Estilos de programación

Presentaremos diferentes estilos para presentar el código, como esto afecta su legibilidad y cómo puede ayudar en la construcción del software.

b. Documentación

Presentaremos diferentes técnicas para auto documentar el código, de forma que no sea siempre necesario utilizar comentarios. También veremos en caso de que sea necesario, buenas técnicas para escribir comentarios en el código.

c. Carácter personal

Por último, veremos algunos aspectos generales en el carácter de los participantes en un proyecto, que pueden afectar la construcción del mismo. Entre otros, presentaremos temas como la comunicación y cooperación entre los pares,

Bibliografía:

Code Complete: A Practical Handbook of Software Construction, Second Edition – Steve McConnell – Microsoft Press – 978-0735619678 - 2004

(título del libro-nombre del autor-editorial-ISBN-fecha de edición)
