

UNIVERSIDAD DEL BÍO-BÍO
VICERRECTORÍA ACADÉMICA – DIRECCIÓN DE DOCENCIA

PROGRAMA DE ASIGNATURA

I. IDENTIFICACIÓN

Nombre asignatura: Ingeniería de Software	Período de Vigencia: 2023-2023
Código: 620515	
Tipo de Curso: Obligatoria, Formación disciplinaria, Ingeniería de Software, Semestral	

Carrera: Ingeniería de Ejecución en Computación e Informática	Departamento: DSI	Facultad: Ciencias Empresariales
Nº Créditos SCT: 5	Total de horas: 12 Cronológicas: 144 Pedagógicas: 216	Año/ semestre 3er año / 6to semestre
Horas presenciales: 90 HT: 3 HP: 2 HL: 0	Horas trabajo autónomo: 126 HT: 3 HP: 4 HL: 0	
Prerrequisitos: Si Asignatura: Metodología de Desarrollo Código: 620510	Correquisitos: No Asignatura: Código:	

II.- DESCRIPCIÓN

II.1 Presentación: Relación de la Asignatura con las Competencias del Perfil de Egreso

Asignatura teórico práctica que al finalizar permitirá al estudiante construir software usando diversos elementos propios de la Ingeniería de Software. Se ubica en el sexto semestre, tercer año, de la carrera. Contribuye a las subcompetencias específicas:

CE6: Construir software de acuerdo a estándares definidos, acorde a requerimientos y modelos previamente establecidos.

II.2 Descriptor de competencias (metas de la asignatura)

Construir software utilizando un conjunto de métodos, técnicas, herramientas y estándares que permitan obtener un producto que cumpla con los requerimientos técnicos, funcionales y no - funcionales.

Resultados de Aprendizaje:

1. Analiza cuál es el método más adecuado para un proyecto de desarrollo de software determinado.
2. Planifica las actividades requeridas por un proceso de desarrollo para la adecuada gestión de un proyecto de desarrollo de software.

3. Aplica métodos, técnicas, herramientas y estándares de Requerimientos, Análisis y Diseño de manera sistemática en el desarrollo de un software para que satisfaga los requerimientos técnicos, funcionales y no – funcionales especificados.
4. Aplica técnicas de prueba de software para comprobar que un software funciona correctamente y satisface los requerimientos funcionales.

II.3 Aprendizajes Previos

- Aplica técnicas de modelamiento de datos y funcionales.
- Construye software utilizando lenguajes de programación.
- Construye bases de datos que respondan a un conjunto establecido de necesidades.
- Reconoce el contexto de negocios donde se insertan las aplicaciones de software.

III. RESULTADOS DE APRENDIZAJE

Resultados de Aprendizaje
1. Analiza cuál es el método más adecuado para un proyecto de desarrollo de software determinado.
2. Planifica las actividades requeridas por un proceso de desarrollo de software para su adecuada gestión.
3. Aplica métodos, técnicas, herramientas y estándares de Requerimientos, Análisis y Diseño de manera sistemática en el desarrollo de un software para que satisfaga los requerimientos técnicos, funcionales y no – funcionales especificados.
4. Aplica técnicas de prueba de software para comprobar que un software funciona correctamente y satisface los requerimientos funcionales.

IV. SISTEMA DE EVALUACIÓN

RESULTADOS DE APRENDIZAJE	EVIDENCIAS DE APRENDIZAJE (proceso y producto)
1. Analiza cuál es el método más adecuado para un proyecto de desarrollo de software determinado.	<ul style="list-style-type: none"> • Informe/exposición Trabajo individual /Grupal • Exposición Trabajo Práctico Grupal • EVALUACIÓN I (acumulativo de RA1 y RA2)
2. Planifica las actividades requeridas por un proceso de desarrollo de software para su adecuada gestión.	
3. Aplica métodos, técnicas, herramientas y estándares de Requerimientos, Análisis y Diseño de manera sistemática en el desarrollo de un software para que satisfaga los requerimientos técnicos, funcionales y no – funcionales especificados.	<ul style="list-style-type: none"> • Informe/exposición Trabajo individual /Grupal • Exposición Trabajo Práctico Grupal • Portafolio del Proyecto de desarrollo de software. • EVALUACIÓN II (acumulativo de RA3 y RA4)
4. Aplica técnicas de prueba de software para comprobar que un software funciona correctamente y satisface los requerimientos funcionales.	

La evaluación de la asignatura considera: (%)	
EVALUACIÓN 1 y 2	60%
INFORMES/ EXPOSICIÓN TRABAJOS PRÁCTICOS y AYUDANTÍA, (grupal o individual) revisados en clases	05%
PORTAFOLIO DEL PROYECTO DE DESARROLLO DE SOFTWARE y SOFTWARE (informes parciales, avance de sw, informe final, y producto final).	35%

Notas importantes:

- **El no cumplimiento o entrega inferior al 50%** de lo requerido en algún hito del proyecto (informes y presentaciones de avance del proyecto) será causal de NCR. Lo anterior también aplica a la insistencia –no justificada o la incapacidad para cumplir con la interrogación.
- Durante la revisión (de avance/final) de software serán solicitados cambios en las **implementaciones presentadas a cualquier desarrollador**, esta interrogación es parte de la evaluación y es condición de aprobación.
- Las entregas de software y el software final **serán evaluados en el SERVIDOR de la FACE y como respaldo, en un server externo a la UBB**. La entrega en local o archivos comprimidos u otra forma que se le pueda imaginar no es considerada como entrega, es decir no cumple con lo exigido.
- Cada equipo debe utilizar un sistema de versionamiento de código, por ejemplo, Github UBB, Gitlab FACE. En cualquier caso, los repositorios deben ser compartidos con los profesores y ayudantes.
- **La copia en evaluaciones y trabajos será calificada con nota 1** para todos los involucrados en ella.
- **Sólo** los trabajos en clases que son presentados y revisados en clases **forman parte de la evaluación**.
- **Los alumnos cuyo** promedio de las 2 evaluaciones (certámenes) es menor a 4, deben rendir integradora.
- **Una vez aprobada la asignatura, los alumnos cuyo** promedio de las 2 evaluaciones es mayor o igual a 4, se premiará el esfuerzo destinado al proyecto a través de un promedio final considerando las ponderaciones 50%, 45% y 5% respectivamente.

V. BIBLIOGRAFÍA

Fundamental

- PRESSMAN, R. S. 2010. Ingeniería del Software. Un enfoque práctico, 7ª edición. McGraw Hill.
- SOMMERVILLE, I. 2010. Ingeniería de Software, 9na edición. Addison Wesley.
- LARMAN, C. 2003. UML y Patrones. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado, 2da. edición. Pearson.

Complementaria

- PFLEEGER, S.. 2002. Ingeniería de Software: teoría y práctica. 4ta edition. Prentice Hall.
- BOHEM, B. 2006; A View of 20th and 21st Century Software Engineering. The International Conference on Software Engineering (ICSE). Shanghai, China.
- BRAUDE, E. 2003. Ingeniería de Software. Una Perspectiva Orientada a Objetos. Alfaomega.
- RUMBAUGH, J., JACOBSON, I., BOOCH, G. 2007. El lenguaje Unificado de Modelado: manual de referencia. Pearson.
- STEVENS, P. 2007. Utilización de UML en Ingeniería del Software con Objetos y Componentes, Pearson.
- SOFTWARE ENGINEERING INSTITUTE (SEI), Carnegie Mellon University (<http://www.sei.cmu.edu/>)
- THE OBJECT MANAGEMENT GROUP (OMG) (<http://www.omg.org/>)
- WYSOCKI, R. K. 2006. Effective Software Project Management. ISBN: 978-0-7645-9636-0. Wiley Publishing Inc. 1º Edición.

- Videos primera unidad
- <https://youtu.be/Qvo2oISMgZQ>
- <https://youtu.be/H7tr5hyIIS4>
- <https://youtu.be/wRQ9TSdEUAE>
- <https://youtu.be/g55pMTih9CM>