Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI1201 - Programación II. Grupo 04. I-2014

Horario: L,J 15-17 103-IF. Consulta: L,J 17-16 201-IF
Profesor Jeisson Hidalgo-Céspedes
Casillero: 04. Correo: jeissonh@gmail.com

Carta al estudiante

Objetivo

Los estudiantes, en equipos, aprenderán a solucionar problemas mediante la programación de aplicaciones y bibliotecas reutilizables de software de mediana complejidad.

Competencias

La experiencia de aprendizaje del curso ayudará a los estudiantes a solucionar en equipos un problema real, de interés personal o social, mediante el desarrollo de una aplicación de software de mediana complejidad, que requiera al menos la construcción de un módulo o biblioteca reutilizable. Esta competencia general se alcanzará desarrollando las siguientes competencias específicas:

	Competencia	Descripción	Dominio	Nivel*
1	Trabajo en equipo	Desarrollar habilidades de comunicación y coordinación, así como el uso de herramientas para trabajo en equipo como control de versiones.	General	▶▶
2	Trabajo por proyectos	Proponer, administrar, ejecutar y trabajar por proyectos. Utilizar herramientas automatizadas para la administración de proyectos.	General	▶
3	Prácticas básicas de ingeniería de software	Crear soluciones de software con un mínimo de calidad utilizando principios básicos de ingeniería de software, como manejo de requerimientos, tareas y casos de prueba.	Técnica	▶
4	Dominio del lenguaje de programación	Dominar un lenguaje de programación, sus paradigmas y sus mecanismos de abstracción fundamentales (por ejemplo: orientación a objetos, herencia, polimorfismo, programación genérica, manejo de excepciones).	Técnica	▶▶
5	Buenas prácticas de programación	Aplicar buenas prácticas de programación para el mantenimiento del código y trabajo colaborativo. Por ejemplo: documentación del código no obvio, seguir una convención de estilo, hacer programación defensiva.	Técnica	▶▶▶
6	Reutilización de código	Reutilizar código, a través del uso de bibliotecas de software, y la construcción de componentes genéricos de software.	Técnica	▶▶▶
7	Resolución de problemas	Plantear varias soluciones a un mismo problema y determinar cuál es la más eficiente y eficaz de acuerdo a diferentes criterios.	Técnica	▶▶
8	Juicio crítico	Criticar las actuaciones y soluciones propias y de otras personas. Defender sus puntos de vista. Participar en la formulación de las evaluaciones.	General	▶
9	Comunicación	Escribir documentos y realizar exposiciones claras, concisas, convincentes y amenas.	General	▶
10	Autodidáctica	Indagar, aprender autodidácticamente sobre temas no tratados en el curso pero que son importantes para la solución de los problemas planteados por los estudiantes.	General	▶▶

* Nivel de desarrollo a alcanzar de la competencia: introductorio (▶), intermedio (▶▶), avanzado (▶▶▶).

Metodología

La **metodología de aprendizaje** del curso será constructivista, en particular *aprendizaje basado en proyectos*. La **metodología de evaluación** será una combinación de *evaluación entre pares* y *evaluación del profesor*.

Los estudiantes realizarán un ambicioso proyecto de desarrollo de software en parejas o en tríos. El problema a solucionar será escogido por los miembros de cada equipo, de acuerdo a sus intereses o pasiones. Debe ser un proyecto original y suficientemente desafiante que requiera más de un semestre para solucionarse por completo. Deberá además cumplir con ciertos requerimientos inicialmente propuestos por el profesor, los cuales se detallan en la **idea de proyecto**. Esta idea será planteada en forma individual y expuesta oralmente por cada estudiante. Recibirá retroalimentación por parte del profesor y demás compañeros sin influir sobre su calificación final, con el fin de que los estudiantes inicien su participación en el proceso de evaluación.

Los estudiantes conformarán equipos de acuerdo a intereses afines. Cada equipo elaborará una **propuesta de proyecto**, delimitando el producto que van a realizar durante el curso. Sobre el documento de diseño resultante, el equipo recibirá retroalimentación por parte del profesor y los otros equipos. El equipo deberá considerar cada retroalimentación, aplicándola o rechazándola justificadamente. Finalmente los miembros del equipo evaluarán su participación en la elaboración y mejora de la propuesta.

Cada equipo trabajará en la construcción de su aplicación. Aproximadamente cada dos semanas y media se realizará una **revisión de seguimiento** de proyecto, para un total aproximado de 6 avances. Se evaluarán mediante un coloquio, es decir, una reunión entre los miembros del equipo y el profesor donde se examinará el avance en el producto, la calidad del código y criterios afines. Los miembros del equipo evaluarán su participación durante el avance. Esta reunión permitirá decidir las tareas o requerimientos que el equipo realizará para el próximo avance. En forma democrática entre todos los equipos se decidirá además las temáticas a trabajar en cada clase y su método (laboratorio, clase magistral, taller, un híbrido, etc.).

Al final del semestre los equipos **presentarán su producto final** ante toda la clase. Tanto el equipo, los otros equipos, y el profesor retroalimentarán sobre la exposición realizada y el producto final.

Tarea	#	Producto	Semana	Otros	Equipo	Prof.	Peso
Idea de proyecto	0	Presentación	1 [13-mar]	25%	-	75%	0%
Propuesta de proyecto	1	Documento de diseño	2 [20-mar]	10%	20%	70%	**10%**
Seguimiento de proyecto	2	Avance 1, código ejecutable	5 [07-abr]	-	25%	75%	**10%**
	3	Avance 2, código ejecutable	7 [28-abr]	-	25%	75%	**10%**
	4	Avance 3, código ejecutable	9 [15-may]	-	25%	75%	**10%**
	5	Avance 4, código ejecutable	11 [29-may]	-	25%	75%	**10%**
	6	Avance 5, código ejecutable	13 [12-jun]	-	25%	75%	**10%**
	7	Avance 6, código ejecutable	15 [26-jun]	-	25%	75%	**10%**
Presentación final	8	Demostración del producto final	18 [17-jul]	10%	20%	70%	**20%**
Colaboración	9	Tarjetas de agradecimiento	*	-	-	-	**10%**

El sistema de **tarjetas de agradecimiento** pretende estimular la colaboración entre equipos de estudiantes durante el desarrollo de sus proyectos. Consiste en 10 tarjetas vacías que cada estudiante tendrá inicialmente en su poder. Cuando un estudiante recibe una ayuda significativa de otro, podrá firmar una tarjeta de agradecimiento y otorgársela. Al final del semestre cada estudiante podrá reclamar las tarjetas de agradecimiento que recibió, cada una con un valor de 1% de la calificación final. Naturalmente un estudiante puede llegar a poseer más de 10 tarjetas. Ningún estudiante puede otorgarse una tarjeta de agradecimiento a sí mismo. El profesor también dispone de 10 tarjetas de agradecimiento, pero no puede recibir tarjeats de los estudiantes.

Bibliografía

1. Deitel, H.M.; Deitel, P.J. *C++ How to Program*, 8th edition. Prentice-Hall, 2012.
2. Stroustrup, Bjarne. *The C++ Programming Language*, 4th edition. Addison-Wesley; 2013.
3. Josuttis, Nicolai. *The C++ Standard Library: A Tutorial and Reference*, 2nd edition. Addison-Wesley; 2012.
4. Kernighan, Brian; Ritchie, Dennis. *El lenguaje de programación C*, 2da edición. Pearson, México, 1991.