

Diseño de Software

Bienvenidos

Políticas del Curso

Semana 1

Agenda

- El Profesor
- Detalles del curso
- Esquema de evaluación
- Sugerencias para comunicaciones escritas

El Profesor

El Profesor

David Jurado Mosquera

- Detalles académicos
 - MSc. Inteligencia Artificial en Universidad Politécnica de Madrid, Es.
 - Ingeniero en Computación de ESPOL, Ec.
- Experiencia profesional
 - Consultor de Sistemas en el Consejo de Educación Superior.
 - Investigador en Centro de Tecnologías de Información, ESPOL.
- Experiencia académica
 - Profesor de ESPOL desde el 2014: Diseño de Software, Fundamentos de Programación, Programación NodeJS, Java, entre otros.
 - Ayudante Académico en ESPOL por 2 años: Fundamentos de programación y Estructura de datos.
- Detalles de contacto:
 - djurado@espol.edu.ec, djurado@fiec.espol.edu.ec
 - Oficina 11F-O106

Detalles del curso

Descripción del curso

- El curso de formación profesional presenta los diferentes **paradigmas, patrones y técnicas de modelado** en el desarrollo de un sistema de software.
- Busca un diseño de software que cumpla con las especificaciones de requerimientos, obteniendo como resultado un **producto de software de calidad**, mantenable y extensible.
- Además, introduce al uso de **herramientas** para la construcción de proyectos, el **control de versiones** y el uso de **marcos de trabajo para la validación** de productos de software.

Objetivos del curso

- Diseñar un producto de software aplicando **principios de diseño** orientado a objetos para que sea **robusto, mantenible y escalable**.
- Aplicar patrones de diseño en la **elaboración de diagramas** utilizando el lenguaje de **modelado** unificado para la resolución de los problemas propuestos.
- Reorganizar el código fuente de un sistema mediante técnicas de **refactorización** para la simplificación de los mantenimientos futuros.
- Emplear herramientas informáticas en el **control de versiones** y la ejecución de **pruebas unitarias** para la generación de software de **calidad** en un entorno colaborativo.

Resultados de aprendizaje - ABET

5. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa.
6. Habilidad para aplicar teoría de ciencias computacionales y fundamentos de desarrollo de software para producir soluciones basadas en computación.

Contenido del curso

- Para la primera evaluación:

1. Introducción al diseño
1.1. Diseño de software: conceptos, diseño arquitectural y diseño detallado.
1.2. Herramientas para el control de versiones de software.
1.3. Herramientas para desarrollo y operaciones (DevOps).
1.4. Entrega continua: beneficios y principios.
1.5. Paradigmas del diseño: orientado a aspectos y orientado a componentes.
2. Diseño orientado a objetos
2.1. Principios del diseño orientado a objetos: abstracción, encapsulación, herencia y polimorfismo.
2.2. Principios de diseño aplicados al desarrollo.
2.3. Diseño de arquitectura modelo vista controlador.
2.4. Lenguaje unificado de modelado: casos de uso, clases y secuencias.
2.5. Proceso de implementación a partir de un diseño.
3. Patrones de diseño
3.1. Introducción a los patrones de diseño.
3.2. Elementos de un patrón de diseño.
3.3. Patrones creacionales.

Contenido del curso

- Para la segunda evaluación:

3.4. Patrones estructurales.
3.5. Patrones de comportamiento.
4. Refactorización
4.1. Identificación de "malos olores" de programación.
4.2. Definición de refactorización y su clasificación.
4.3. Refactorización a nivel de métodos.
4.4. Refactorización a nivel de clases y objetos.
4.5. Refactorización a nivel de datos y condicionales.
5. Pruebas unitarias
5.1. Introducción a las pruebas de software.
5.2. Elaboración de casos de prueba.
5.3. Marcos de trabajo para pruebas de software.

Bibliografía

- Martin, Robert C. (2018). Clean Architecture: A Craftsman's Guide to Software Structure and Design. (First Edition).
- Stevens Perdita, Pooley Rob. (2006). Using UML: Software Engineering with Objects and Components. (Second Edition).
- Fowler Martin, Beck Kent. (2019). Refactoring: Improving the Design of Existing Code. (Second Edition).
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. (1st Edition).
- Elisabeth Freeman & Freeman, Eric & Bates, Bert & Sierra, Kathy. (2014). Head first design patterns. (2da).
- Pressman Roger, Maxim Bruce. (2019). Software Engineering: A Practitioner's Approach. (Ninth Edition).

Horarios de clase

- Las sesiones están planificadas los días Martes y Jueves de 09:00 a 11:00 / 11:00 a 13:00.
- Los talleres están planificados los días Jueves.
- Horas de oficina (11F-O106):
 - Viernes de 9:00 a 11:00.
 - Políticas de puertas abiertas.

Sesiones virtuales

- Se solicita que los estudiantes apaguen sus micrófonos en el momento que entran al aula virtual y que activen sus micrófonos solo cuando se lo requiera o se le haya dado la palabra.
- Se solicita que al ingresar a la clase saluden utilizando el chat de la sesión.
- A fin de agilizar el avance de la clase, la entrada no será permitida después de 15 minutos.
- Los controles de lectura se realizarán durante los primeros 10 minutos de la sesión.

Fechas claves

- Evaluaciones:
 - 18 de junio, 09 de julio y 16 de julio, Primera Evaluación *
 - 07 de septiembre de 2020, Segunda Evaluación **
 - 21 de septiembre de 2020, Tercera Evaluación **
- *Durante las sesiones semanales
- **Las evaluaciones están planificadas los Lunes de 11:00 a 13:00

Esquema de evaluación

Evaluaciones

- Cada evaluación constará de los siguientes componentes:

Actividad		Primera Evaluación	Segunda Evaluación	Tercera Evaluación
EHD/Teórico	Examen	--	20%	70%
	Lecciones **	35%	15%	--
EHTA/Autónomo	Trabajos Autónomos	25%*	25%*	--
	Controles de Lectura **	10%	10%	--
EHP/Práctico	Talleres	30%		--

- *Algunos trabajos requerirán sustentación (factor entre 0 y 1)
- ** No se considerará la nota más baja en cada parcial

Esquema de evaluación

- La deshonestidad (incluido el plagio) produce una nota de cero.
- Auto-registro en grupos de trabajo para talleres y tareas.
- Los grupos de trabajo se componen de entre 3 a 5 estudiantes.

Co-Evaluación del trabajo en equipo

- Las notas individuales en trabajos grupales serán determinadas con base en una rúbrica.
 - Por ejemplo, considere una tarea que resulta de un trabajo grupal (ej.: la presentación de un póster) que obtiene una nota del 85%. Un estudiante que obtiene un Nivel 5 de valoración, obtendrá la totalidad de los puntos (esto es 85%). Un estudiante que recibe un Nivel 2 de valoración, recibirá un 25% de los puntos ganados (esto es 21.25%).

Co-Evaluación del trabajo en equipo

- **Nivel 5: 100% de la nota del trabajo en equipo**

- El integrante participó en todas las discusiones, contribuyendo con sus propias ideas y sugerencias en cada fase del trabajo.
- El integrante participó en todas las actividades del trabajo en equipo (ej.: planificación, recolección de datos o información, análisis de datos, escritura, presentación, etc.).
- El integrante asistió a todas las reuniones grupales.
- El integrante completó todas las tareas que acordó llevar a cabo dentro del trabajo grupal.

Co-Evaluación del trabajo en equipo

- **Nivel 4: 75% de la nota del trabajo en equipo**

- El integrante participó en cada fase del trabajo pero mostró poca iniciativa para contribuir con ideas y sugerencias.
- El integrante usualmente participó en las actividades del trabajo, pero no siempre.
- El integrante usualmente asistió a la reuniones grupales.
- El integrante usualmente completó las actividades que había acordado llevar a cabo dentro del trabajo grupal.

Co-Evaluación del trabajo en equipo

- **Nivel 3: 50% de la nota del trabajo en equipo**

- El integrante participó ocasionalmente en varias fases del trabajo, generalmente como un “seguidor” más que como un contribuyente al trabajo grupal.
- El integrante ocasionalmente asistió a reuniones grupales.
- El integrante completó una o dos tareas menores de las que él acordó llevar a cabo dentro del trabajo grupal.

Co-Evaluación del trabajo en equipo

- **Level 2: 25% de la nota del trabajo en equipo**

- El integrante contribuyó muy poco con el trabajo grupal, sin mostrar iniciativa.
- El integrante se perdió la mayoría de las reuniones grupales.
- El integrante no completó la mayoría de las tareas que había acordado llevar a cabo dentro del trabajo grupal.

- **Nivel 1: Sin crédito**

- El integrante no participó en el desarrollo del trabajo.

Sugerencias para comunicaciones escritas

¿Cómo escribir correctamente un correos electrónico?

- <http://es.wikihow.com/escribir-un-e-mail-formal>
- <http://www.ecoescritura.com/%C2%BFcomo-escribir-correos-electronicos/>
- http://www.ehowenespanol.com/reglas-saludo-correo-electronico-lista_43535/

Antes de finalizar

Próxima sesión

- Diseño de software: conceptos, diseño arquitectural y diseño detallado.
- Herramientas para el control de versiones de software.