



NOMBRE DEL CURSO: Análisis y diseño de sistemas 2

| | | | |
|--------------------------------------|---------------------------------------|--|--------------------------------|
| CODIGO: | 775 | CREDITOS: | 5 |
| ESCUELA: | Ciencias y Sistemas | AREA A LA QUE PERTENECE: | Área de desarrollo de software |
| PRE REQUISITO: | 283 – Análisis y diseño de sistemas 1 | POST REQUISITO: | 780 – Software Avanzado |
| CATEGORIA: | Obligatorio | SEMESTRE: | Segundo Semestre 2014 |
| CATEDRÁTICO (A): | Ing. Luis Alberto Arias Solórzano | AUXILIAR: | Robson Giovanni Cruz Guerrero |
| EDIFICIO: | T-3 | SECCIÓN: | B |
| SALON DEL CURSO: | 209 | SALON DEL LABORATORIO: | 209 |
| HORAS POR SEMANA DEL CURSO: | 4 | HORAS POR SEMANA DEL LABORATORIO: | 2 |
| DÍAS QUE SE IMPARTE EL CURSO: | Sábado | DIAS QUE SE IMPARTE EL LABORATORIO: | Sábado |
| HORARIO DEL CURSO: | 10:50 PM – 14:10 PM | HORARIO DEL LABORATORIO: | 14:10 PM - 15:50 PM |

Descripción del Laboratorio:

El laboratorio de Análisis y diseño de sistemas 2 complementa el contenido impartido en clase así como con la continuación y complemento de conceptos Del curso de análisis y diseño 1 de sistemas.

Objetivo General:

Lograr que el estudiante adquiera los conocimientos necesarios para poder analizar y diseñar un sistema de acuerdo a las tecnologías y herramientas disponibles, adoptando para ello las buenas prácticas de análisis, diseño y de programación de sistemas.

Objetivos específicos:

- Capacitar al estudiante para que sea capaz de analizar requerimientos de un sistema utilizando las metodologías de desarrollo de sistemas adecuadas.
- Que el estudiante comprenda el proceso para efectuar la entrega continua de software.
- Que el estudiante conozca y ponga en práctica los conceptos de arquitectura de software abarcados durante el curso.
- Que el estudiante comprenda y practique los distintos patrones de diseño que Existen.
- Familiarizar al estudiante con las herramientas disponibles para aplicar un completo y correcto desarrollo de software aplicando técnicas de ingeniería.

Habilidades:

1. Que el estudiante investigue, comprenda y aplique conocimientos adquiridos durante cursos anteriores para diseñar y analizar sistemas de forma óptima.
2. Analizar requerimientos de software para la realización de un sistema cumpliendo con todas las etapas del ciclo de vida de software.
3. Extraer y representar el conocimiento necesario para construir o implementar una solución para un proyecto de software cumpliendo prácticas de administración de la configuración y el diseño de soluciones informáticas a un problema propuesto.

Competencias:

1. Que el estudiante pueda aplicar una buena administración de la configuración.
2. Capacidad para analizar y determinar requerimientos de software.
3. Resolver y reconocer problemas clásicos de la administración de proyectos.
4. Capacidad para crear y utilizar herramientas de integración continua, control de versiones y pruebas de software.
5. Capacidad de adaptarse a cambios en los requerimientos iniciales de software

Metodología:

- Se impartirán exposiciones presenciales con temas enfocados a la práctica de los conceptos del curso.
- Se realizarán exámenes cortos como parte de la evaluación de prácticas y contenido impartido en el laboratorio.
- Se realizarán prácticas donde se busca que el estudiante conozca y experimente con las distintas herramientas mencionadas a lo largo del contenido del laboratorio, aplicándolas sobre problemas reales.
- Se realizarán tareas e investigaciones para que el estudiante pueda complementar y comprender de mejor manera el tema tratado.

Evaluación Del Rendimiento Académico: El laboratorio se aprueba con una nota mayor o igual a 61 puntos.

| Procedimiento | Ponderación |
|--------------------------|--------------------|
| Tareas e investigaciones | 25 puntos |
| 3 Prácticas | 35 puntos |
| 2 exámenes cortos | 15 puntos |
| 1 exposición | 15 puntos |
| Examen Final | 10 puntos |
| | ----- |
| Nota de Promoción | 100 puntos |

Contenido Programático

UNIDAD 1: Administración de la configuración.

- 1.1 Control de Versiones
 - 1.1.1 Herramientas de control de versiones: SVN y GIT
- 1.2 Integración Continua
 - 1.2.1 Ciclo general de la integración continua
 - 1.2.2 Herramienta para integración continua: Hudson y Jenkins
 - 1.2.3 Otras herramientas relacionadas (Ej. Maven, Ant etc.)

UNIDAD 2: Entrega continua

- 2.1 Ambientes del software
- 2.2 Flujo de entrega de software
- 2.3 Componentes de la entrega continua

UNIDAD 3: Pruebas

- 3.1 Tipos de pruebas
 - 3.1.1 Pruebas unitarias, de integración y de sistema
 - 3.1.2 Pruebas de aceptación de requerimientos funcionales
 - 3.1.3 Pruebas de usabilidad, exploratorias y de demostración
 - 3.1.4 Pruebas de aceptación de requerimientos no funcionales
 - 3.1.5 Herramientas para pruebas

UNIDAD 4: Arquitectura de software

- 4.1 Arquitectura por capas
- 4.2 Arquitectura 4 + 1 vistas
- 4.3 MVC
- 4.3 Herramientas del mercado. (ArgoUML, StarUML. Etc.)

UNIDAD 5: Arquitecturas orientadas al servicio

5.1. SOA

5.2. Web services y SOA

5.3 Cloud Computing

UNIDAD 6: Patrones de diseño

6.1. Patrones de creación

6.2. Patrones de estructura

6.3. Patrones de comportamiento

Bibliografía:

- "Continuous Delivery", Jez Humble, David Farley
- "Version Control with Subversion", Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michal Pilato.
- "Git Community Book", Several Authors
- "Essential Software Architecture", Ian Gorton
- "Service Oriented Architecture, Concepts, Technology and Design", Thomas Erl
- "Head First Design Patterns", Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra