Programa

CARRERA: Analista Universitario de Sistemas Informáticos	**PLAN:** 2014
UNIDAD CURRICULAR: Modelación y Diseño de Sistemas	
CURSO: 3ero	
REGIMEN: Anual	
ASIGNACION HORARIA SEMANAL: 4	
PROFESOR/ES: Ing. Fernando Bono	
AÑO: 2021	

FUNDAMENTACIÓN

El modelado y diseño de sistemas informáticos es la piedra basal de un proyecto de software exitoso, pasar de un modelo de la vida real a un modelo informático requiere un gran desafío, no sólo en cuanto a los conocimientos tecnológicos que debe poseer el alumno, sino también la forma de interactuar con el dueño del producto (llámese cliente). Dicho de otra manera, en el modelado de software, convergen los conocimientos teóricos y técnicos que se han adquirido durante el transcurso de la materia. Por ello, se debe proveer al alumno los procedimientos y artefactos para el modelado y diseño de Software de forma agnóstica (independiente de las herramientas/lenguaje de codificación) conociendo patrones de diseño, arquitectura y metodologías ágiles de desarrollo.

Un buen diseño de software evita el retrabajo, un buen modelado facilita el cambio en etapas ulteriores, un software bien diseñado y modelado, permite: Escalabilidad, Mantenibilidad y Adaptabilidad.

OBJETIVOS GENERALES

- Aprender a utilizar UML (Unified Modeling Language) para modelar clases
- Entender y seleccionar entre los distintos patrones de diseño y arquitectura de software
- Comprender las metodologías de desarrollo ágiles
- Crear y gestionar tareas relacionadas a la construcción del software.
- Modelar requerimientos de software
- Aprender a utilizar herramientas online para el diseño, gestión y modelado de software
- Aprender a Utilizar Herramientas de Gestión de Proyecto

Contenido

Unidad I - Software: Visión general

1. Introducción

1.1. El Modelado de Software

1.2. Modelado orientado a Objetos y el UML

1.3. Diseño de Arquitectura de Software

1.4. Método y Notación

1.5. COMET: Metodología de modelado y diseño colaborativo para el desarrollo de software

1.6. El estándar UML

1.7. Evolución del modelado de software y métodos de diseño

1.8. Evolución de los métodos de Diseño y Análisis Orientado a Objetos

1.9. Repaso de los métodos de diseño Concurrentes, distribuidos y en tiempo real.

Unidad II - Modelado de Software

1 Modelando casos de uso

1.1 Modelado de Requerimientos

1.2 Casos de uso

1.3 Actores

1.4 Identificación de casos de uso

1.5 Documentación de casos de uso en el Modelo de Caso de Uso

1.6 Requerimientos No funcionales

1.7 Diagramas de actividades

2 Notación UML

2.1 Diagramas UML

2.2 Diagramas de Casos de Uso

2.3 Clase y Objetos

2.4 Diagramas de Clases

2.5 Diagramas de Interacción

2.6 Diagramas de State Machine

2.7 Packages (paquetes)

Unidad III – Proyectos, Metodologías y Marcos de Trabajo

1. Ciclo de vida de software: Modelos y Procesos

1.1 Modelos de ciclos de vida de software

1.2 Diseño, verificación y validación

1.3 Testing de software

2 Diseño de Software y Conceptos de Arquitectura

2.1 Conceptos de Orientación a Objetos

2.2 Ocultamiento de información

2.3 Herencia

2.4 Procesamiento concurrente

2.5 Patrones de Diseño

2.6 Arquitectura de Software y componentes

2.7 Atributos de Calidad de Software

3 Metodologías Ágiles

3.1 El Manifiesto Agile

3.2 Orígenes

3.3 Utilización

3.4 Características

3.5 Valores del Manifiesto Agile

3.6 Nivel de Ruido en el proyecto

3.7 Scrum Framework

3.7.1 Ceremonias

3.7.1.1 Sprint Planning

3.7.1.2 Dailies Meetings

3.7.1.3 Retrospectives

3.7.1.4 Showcase

3.7.2 Artefactos

3.7.2.1 Product Backlog

3.7.2.1.1 Backlog Item

3.7.2.1.2 Gestión del Backlog

3.7.3 Roles

3.7.3.1 Scrum master

3.7.3.2 Product owner

3.7.3.3 Team members

3.7.3.4 Stakeholders

3.7.4 Burndown Chart

METODOLOGÍA DE ENSEÑANZA

Las clases serán inicialmente virtuales, on-line con un cronograma de videos grabados y clases de consulta y práctica por reuniones virtuales.

BIBLIOGRAFÍA:

- David A. Ruble.

“Análisis y Diseño Práctico de Sistemas”.

Editorial Prentice Hall Hispanoamericana, S. A. 2000.

“Lenguaje Unificado de Modelado”.

- Ian Sommerville

“Ingeniería de Software”

Editorial Pearson, 2011

- Software Modeling and Design ByHassan Gomaa, Cambridge University Press
- Design Patterns: Elements of Reusable Object-Oriented Software 1st Edition, by Erich Gamma
- Software Architecture in Practice (2nd Edition) By Len Bass, Paul C. Clements, Rick Kazman
- Applying UML and patterns, by Craig Larman
- Agile and Iterative Development: A Manager’s Guide by Craig Larman
- Agile Estimating and Planning by Mike Cohn
- Agile Project Management with Scrum by Ken Schwaber
- Agile Retrospectives by Esther Derby and Diana Larsen
- Agile Software Development Ecosystems by Jim Highsmith
- Agile Software Development with Scrum by Ken Schwaber and Mike Beedle
- Scrum and The Enterprise by Ken Schwaber
- User Stories Applied for Agile Software Development by Mike Cohn