



Gobierno de la Ciudad de Buenos Aires
Ministerio de Educación
Unidad de Coordinación del Sistema de Formación Docente



Instituto Superior del Profesorado
"Dr. Joaquín V. González"

INSTITUTO SUPERIOR DEL PROFESORADO "DR. JOAQUÍN V. GONZÁLEZ"

Nivel: **Superior**

Carrera: **Profesorado de Educación Superior en Informática**

Eje: **Campo de Formación Específica**

Instancia curricular: **Diseño de Sistemas**

Formato: **Seminario**

Cursada: **Anual**

Carga horaria: **4 (cuatro) horas cátedra semanales**

Profesor/ a: **Matías E. García**

Curso: **5to.** Comisión: **B**

Año: 2024

Fundamentación:

Las organizaciones a nivel mundial cada día incorporan sistemas de información para controlar y hacer más eficientes sus procesos productivos y de negocio, lo que convierte a los sistemas de información en una parte estratégica dentro de las mismas, por lo que es importante comprender cada una de las etapas que forman el desarrollo eficaz y eficiente de un sistema de información.

El análisis y diseño de sistemas es una disciplina o área de la informática, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelvan las necesidades de las organizaciones y sus usuarios.

Es por lo anterior que la asignatura Diseño de Sistemas está orientada al proceso de análisis y diseño de sistemas informáticos en general y para la escuela y el aula en particular. Lo que permitirá al estudiante reflexionar acerca de los efectos que pueda producir su inclusión sobre el entorno en el que deba funcionar, adecuando los criterios de diseño a las características del mismo; los dispositivos digitales disponibles y fundamentalmente, las capacidades de las personas que van a utilizarlo, de modo que su operación sea sencilla, cómoda, efectiva y eficiente.

Se busca generar en los estudiantes competencias que van desde la identificación de requerimientos para el desarrollo de un sistema de información hasta llegar a plasmarlos en un análisis y diseño, para un posterior desarrollo.

El desarrollo y mantenimiento de un sistema de software implica tomar decisiones de planificación, diseño, implementación e implantación a lo largo de todo el ciclo de vida del producto software. Para ayudarnos en estas actividades, existen distintas metodologías, cada una con herramientas y técnicas particulares, que resultan más o menos adecuadas según el contexto de uso.

La ingeniería de software trata con áreas muy diversas de la informática y de las ciencias de la computación, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicable a una infinidad de áreas tales como: educación, negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, Internet, redes Intranet y Extranet, etc.

Detectar y estudiar las problemáticas de la informática en el contexto escolar, que se relacionan con las necesidades de nuevos programas que corran en cualquier plataforma y arquitectura, posibilitará el análisis de diferentes situaciones en forma colaborativa y el alcance de progresivos niveles de autonomía en la creación y producción de soluciones, en las que se utilice el lenguaje de programación más apropiado.

En este seminario se proporcionará a los estudiantes una visión general de distintas aproximaciones del desarrollo de software, desde las metodologías ágiles al desarrollo dirigido por modelos. Además, también se profundizará en las técnicas de desarrollo de software vistas en asignaturas anteriores (Programación 1, 2 y 3), poniendo especial énfasis en la creación de diseños robustos y flexibles mediante la elección de una buena arquitectura y la aplicación de patrones de diseño.

Con las nuevas metodologías de desarrollo de software, como la Orientada a Objetos, que permiten mejorar las técnicas de análisis y diseño que hasta el momento se implementaban, se desarrollaran aplicaciones Web y para dispositivos móviles.

El tratamiento de estos conceptos es fundamental para la formación del futuro profesor debido a que las aplicaciones Web son útiles para desarrollar materiales educativos tanto para el aula como para las actividades administrativas de la escuela.

Además, el uso de dispositivos móviles y la proliferación de sistemas embebidos requieren que los estudiantes posean los conocimientos para desarrollar aplicaciones para estas plataformas.

Objetivos:

Que el futuro profesor/a logre:

- Analizar diversas alternativas de solución a partir de la identificación y definición de requerimientos especificados por la organización o cliente.
- Establecer una propuesta para el análisis y diseño de un proyecto de software de acuerdo a la alternativa de solución planteada o establecida.
- Comprender la importancia de realizar un correcto análisis de sistemas antes del diseño y programación de un proyecto de software, incluyendo el seguimiento del ciclo de vida del mismo.
- Modelar casos de uso acorde a los requerimientos del proyecto.
- Documentar el proyecto.
- Resolver problemas mediante el diseño de sistemas con la aplicación del paradigma de desarrollo y la metodología más apropiada a cada situación.
- Seleccionar y utilizar de manera óptima técnicas y herramientas computacionales actuales y emergentes.
- Conocer las tecnologías para el desarrollo de aplicaciones web.
- Realizar pruebas del funcionamiento de aplicaciones web en ambientes reales y de simulación.

Contenidos:

Unidad 1: Fundamentos del análisis y diseño de sistemas

Definición de sistemas de información. Conceptos de análisis y diseño de sistemas. Propósito del análisis y diseño de sistemas. Participantes en el desarrollo de sistemas, roles. El ciclo de vida del desarrollo de sistemas. Metodologías para el análisis y diseño de sistemas. Análisis y diseño Orientado a Objetos. Actividades de gestión de proyectos. Estudio de viabilidad del proyecto. Planificación y calendarización del proyecto. Estimación de Proyectos: PERT y GANTT. Proceso de desarrollo unificado RUP. Uso de las herramientas CASE. Metodologías ágiles Scrum y Extreme Programming.

Unidad 2: Análisis de los requerimientos de información

Técnicas de recolección de información. Entrevistas. Cuestionarios. Revisión documental (Documentos cuantitativos y cualitativos). Observación del Entorno. Definición de un proyecto. Identificación y selección de un proyecto (Tema, Meta). Determinación y clasificación de los requerimientos del sistema. Requerimientos Funcionales. Requerimientos No Funcionales. Análisis del flujo de datos, características, estrategias, desarrollo de gráficas de proceso. Diagramas de flujo de datos. Métodos de entrada y salida.

Unidad 3: Modelado para el análisis de requerimientos.

Diagramas orientados a objetos usando Lenguaje Unificado de Modelado UML.

Modelos de caso de uso, diseño de estados, diseño de actividades, diseño de secuencia, diseño de distribución. Diagramas de clases, Diagramas de objetos, Relaciones y multiplicidad. Diseño de bases de datos, diseño de GUI, diseño de módulos, mapas de navegación de usuario y del sistema.

Documentación.

Unidad 4: El proceso de diseño del sistema

Conceptos sobre arquitecturas de sistemas. Subdivisión del software en capas y particiones. Criterios para la resolución de los requisitos no funcionales mediante una arquitectura apropiada. Diseño de la interfaz hombre máquina. Criterios para el empleo de códigos. Conceptos elementales de bases de datos. Diseño de datos bajo el modelo relacional. Normalización. Pasaje del modelo conceptual al modelo de diseño.

Conceptos básicos de patrones de diseño. Patrones de asignación de responsabilidades. Diagramas de clases de diseño.

Unidad 5: Desarrollo de sistemas web

Transformación de los diseños en código. Plataformas. Desarrollo FrontEnd y BackEnd de aplicaciones web.

Desarrollo con Laravel Framework con HTML5, CSS y PHP.

Unidad 6: Testing, QA e Implementación

Conceptos sobre los distintos métodos de prueba: de caja blanca y de caja negra. Preparación de casos de prueba. Pruebas de integración y del sistema. Implementación exitosa del software de información. Documentación del sistema. Soporte post-instalación.

Modalidad de trabajo / Estrategias Didácticas:

Las clases se desarrollan en forma teórica y práctica. Los estudiantes deben realizar ejercicios a efectos de afianzar las técnicas que van aprendiendo, trabajos prácticos de resolución de problemas concretos en términos de análisis y diseño de proyectos de software en sus hogares y un trabajo integrador final.

Las presentaciones multimedia, recursos didácticos, tareas y trabajos a realizar por los/as estudiantes se encontrarán en el aula virtual de Google Classroom donde el docente publicará el material y realizará las correcciones de los trabajos presentados.

Evaluación, aprobación y acreditación de la instancia curricular

El sistema de regularidad y aprobación se rige por los criterios vigentes en el Régimen de Evaluación Institucional e incorpora las decisiones metodológicas que la/el docente considere pertinentes para la modalidad remota, de manera excepcional.

Durante la cursada, el/la estudiante deberá realizar varios trabajos prácticos por unidad de

contenidos, estos deberán estar todos aprobados para poder regularizar el seminario y poder presentar el final.

Al ser un seminario requerirá de la presentación de un trabajo integrador como final del mismo a entregar en las fechas designadas por la institución. Este trabajo corresponde a la presentación y defensa de su proyecto de investigación y desarrollo, el cual deberá incluir un sistema de información real, comercial, educativo o experto con su respectivo análisis, diseño, implementación y documentación. Se verificará la capacidad de los estudiantes en la utilización de los conceptos fundamentales de la asignatura para la organización de su trabajo, así como el nivel y la calidad del sistema presentado.

Bibliografía específica obligatoria:

- García, Matías, Apuntes de clases (2020)

Bibliografía general:

- Bravo Carrasco Juan. "Gestión de Procesos" (5ta Ed). Evolución, 2013, Chile.
- Bruegge Bernd y Dutoit Allen H. "Ingeniería de Software Orientado a Objetos". Prentice Hall, 2002, Mexico.
- Kendall Kenneth y Kendall Julie. "Análisis y Diseño de Sistemas" (8va Ed). Prentice Hall, 2011, México.
- Larman Craig. "UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado" (2da. Ed). Pearson Prentice Hall, 2003, España.
- Pressman Roger. "Ingeniería de software un Enfoque practico" (7ma Ed). Mc. Graw Hill, 2010, Mexico.
- Rumbaugh J., Jacobson I. y Booch G.. "El Lenguaje Unificado de Modelado Manual de Referencia". Addison Wesley, 2007, España.
- Sommerville Ian. "Ingeniería de Software" (9na Ed). Pearson prentice Hall, 2011, Mexico.
- Stair Ralph y Reynolds George. "Principios de Sistemas de Información" (9na Ed). Cengage Learning, 2010, España.
- Weitzenfeld Alfred. "Ingeniería de Software Orientada a Objetos con UML, Java e Internet". Thomson. 2007, España.
- Wong Durand Sandra. "Análisis y requerimientos de software". Universidad Continental, 2017, Peru.
- Zapata Carlos y Olaya Yris. "Ingeniería de Software para Analistas". LitoNueve, 2007, Colombia.



Matías E. García
Prof. & Tec. en Informática Aplicada
matias@profmatiasgarcia.com.ar
www.profmatiasgarcia.com.ar