



Universidad Nacional de Lanús

Departamento de Desarrollo Productivo y Tecnológico

Carrera: *Licenciatura en Sistemas*

Asignatura: *ORIENTACIÓN A OBJETOS II*

Planes 2011, 2008

Equipo Docente:

Mg. Alejandra Vranic

Esp. Gustavo Siciliano

Lic. Oscar Ruina

Año: *2024*

Cuatrimestre: *3º Año - 1º Cuatrimestre*

1- Fundamentación de la Asignatura:

En el marco de crecimiento de las inversiones tecnológicas que se dan en la región, como centro de soluciones tecnológicas y desarrollo de software para todo el continente, desde el ámbito académico en nuestra asignatura emprendemos este desafío con la investigación del desarrollo de software desde el paradigma de objetos, dando la oportunidad de abrir más puertas para que nuestros jóvenes puedan expandir su nivel de conocimiento.

2- Objetivos:

- Identificar, formular y resolver problemas de Informática.

En la cátedra se propone en el proceso de aprendizaje la identificación de problemas del campo profesional y la resolución de problemas proponiendo modelos de soluciones verificables desde el paradigma de objetos en actividades de trabajos prácticos e instancias de evaluación.

- Idear, diseñar y desarrollar de proyectos de informática

Identificar las necesidades de problemas de informática, analizar su complejidad, conceptualizar y evaluar la viabilidad de posibles modelos de solución mediante técnicas informáticas

- Utilizar de técnicas y herramientas de aplicación en la informática

La cátedra aborda con la utilización de Diagramas UML el modelado de solución para las etapas de diseño y desarrollo de proyectos de informática de laboratorio.

- Propiciar la acción ética y responsable.

La cátedra propicia el uso adecuado de los recursos tecnológicos disponibles y la sustentabilidad ambiental.

- Propiciar para el desempeño en equipos de trabajo

La cátedra propone un trabajo práctico integrador grupal de un proyecto informático del campo profesional.

- Propiciar la comunicación efectiva

La cátedra instrumenta modelos de comunicación con soporte digital que permitan la comunicación efectiva para el desarrollo de trabajo práctico integrador.

- Propiciar para el aprendizaje continuo

La cátedra dispone de un calendario de hitos a cumplir en el trabajo práctico integrador que permite medir el aprendizaje y retroalimentar el proyecto.

- Profundizar los temas desarrollados por el estudiante en Orientación a Objetos I.
- Se enfatizará en la construcción de arquitecturas de software modulares, extensibles y reusables, conceptos claves para aplicaciones de gran porte.
- Utilizar framework para el desarrollo de software utilizados en la actualidad en el campo profesional.

3- Contenidos Mínimos:

- Metodologías de diseño orientado a objetos.
- Construcción de aplicaciones.
- Diseño.
- Patrones.

4- Contenidos:

UNIDAD 1:

Modelo de Casos de Uso: representación de los diagramas de secuencia del sistema.

Comportamiento del sistema. Diagramas de secuencia del sistema (DSS). Caso práctico de un DSS. DSS y los casos de uso. Eventos del sistema y los límites del sistema. Asignación de nombres a los eventos y operaciones. Mostrar el texto del caso de uso. Modelado de casos prácticos.

UNIDAD 2:

Modelo de Dominio: visualización de conceptos. Modelos del Dominio. Identificación de las clases conceptuales. Modelado del negocio. Notación UML, modelos y métodos. Mundo real y su modelado del mundo irreal. Modelado de casos prácticos.

UNIDAD 3:

Modelo de Dominio: añadir asociaciones. Asociaciones. Notación de las asociaciones en UML. Asignación de nombres a las asociaciones. Asociaciones e implementación. Modelado de casos prácticos.

UNIDAD 4:

Modelos del Dominio: añadir atributos. Atributos Notación de los atributos en UML. Tipos de atributos válidos. Clases de tipos de datos no primitivos. Modelado de cantidades y unidades de los atributos. Modelado de casos prácticos.

UNIDAD 5:

Modelo de diseño de objetos con responsabilidades. Responsabilidades y métodos.

Responsabilidades y los diagramas de interacción. Notación del diagrama de clases UML.

Bajo acoplamiento. Alta cohesión.

UNIDAD 6:

Modelo de Diseño: realización de casos de uso con los patrones. Realizaciones de casos de uso. Diseño de objetos: Crear, Buscar, agregar, actualizar, eliminar.

UNIDAD 7:

Modelo de Diseño: determinación de la visibilidad. Visibilidad entre objetos. Visibilidad de los atributos y métodos. Representación de la visibilidad en UML.

UNIDAD 8:

Modelo de implementación: transformación de los diseños en código. Programación y el proceso de desarrollo. Transformación de los diseños en código. Creación de las definiciones de las clases a partir de los diagramas de clases. Creación de métodos a partir de los diagramas de interacción. Manejo de excepciones y de errores. Definición de métodos. Orden de implementación. Programar realizando testeos parciales.

UNIDAD 9:

Diseño de las realizaciones de casos de uso con los patrones de diseño: Adaptador. Factoría. Singleton. Estrategia. Composite. Fachada. Observador.

UNIDAD 10:

Framework basado en composición ModelViewController MVC: Controlador: Java Servlets Vista: Java Server Pages (JSP). Introducción a Java Enterprise Edition Java EE. Entorno de desarrollo. Protocolo http y Lenguaje html. Java Servlet. Java Server Pages. Diseño de aplicaciones Java EE.

UNIDAD 11:

Frameworks para Persistencia de Objetos – Modelos relacionales: Hibernate:

Configuración. Archivo de mapeo. Mapeo de asociaciones: clave primaria y clave foránea.

Conexiones JDBC. Dialectos HQL: from, asociaciones y uniones Join, where, order by,

INSERT, UPDATE, DELETE. Archivo de configuración xml. Declaración de mapeo asociación: uno-a-muchos / muchos-a-uno, uno-a-uno, muchos-a-muchos.

5- Metodología de Trabajo:

El curso se desarrollará en Laboratorio de Software. Desarrollo de la fundamentación teórica de la materia y resolución Proyectos de Software de aplicación real que contienen los contenidos propuestos por la cátedra.

Resolución de módulos de proyectos de aplicación que los estudiantes realizarán en grupo, donde envolverán los contenidos y transitan las distintas etapas del desarrollo de software. Los casos a resolver son problemáticas abiertas que propicien el debate grupal, la consulta de bibliografía para profundizar los conocimientos, la comunicación efectiva y la tutoría del equipo docente para el desarrollo exitoso.

6- Desarrollo de Actividades Prácticas:

Objetivos:

- Estimular el desarrollo de capacidades innovadoras especialmente vinculadas al desarrollo de la industria del software.
- Promover el desarrollo de conocimientos aplicados, así como la socialización de la tecnología como medio para el mejoramiento de las condiciones de vida de nuestra sociedad.
- Alentar el desarrollo de productos e innovaciones tendientes a sustituir importaciones, fomentando la revalorización de la producción nacional.
- Conocer el paradigma de la orientación a objetos.
- Recorrer las distintas etapas del desarrollo de software.
- Identificación de las clases, atributos, métodos y relaciones.
- Identificar las técnicas de representación y documentación para el desarrollo de software basado en tecnologías orientada a objetos.
- Implementar los casos de uso en Java con la utilización de los framework para persistencia de datos y MVC.

Metodología:

Realizar en grupo la producción de un Proyecto Software de Aplicación que deberá orientarse al desarrollo de productos innovadores, destinados a atender necesidades de la comunidad, mejorar procesos de producción o solucionar problemáticas concretas. El período de desarrollo del proyecto será de diez semanas, donde el grupo atravesará por todas las etapas del desarrollo de software y la construcción de una presentación para compartir con sus compañeros la experiencia.

7- Evaluación y Acreditación:

Para acreditar la cursada se requiere un mínimo 75% de asistencia a clases.

Se tomarán 2 (dos) de evaluación obligatorias y la correspondiente instancia de recuperación.

Para aprobar la cursada se deberá obtener un mínimo de 4(cuatro) en las 2 instancias de evaluación o su recuperatorio.

Aprobada la cursada se deberá rendir examen final para la acreditación la materia. .

8- Bibliografía:

Ebersole, S., Mihalcea, V., Boriero, A., Meyer, B., & Vansa, R. (2017, 19 de enero).

Hibernate ORM 5.0 User Guide. Red Hat.

https://docs.jboss.org/hibernate/orm/5.0/userguide/html_single/Hibernate_User_Guide.html

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2009). *Patrones de diseño: elementos de software orientado a objetos reutilizable*. Pearson Educación.

<http://opac.unla.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=166038>

Larman, C. (2003). *UML y patrones* (2.^a ed.). Pearson Educación.

<http://opac.unla.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=161127>

Ordax Cassá, J. M., & Aranzazu Ocaña Díaz Ufano, P. (2012). *Programación web en Java*.

Aula Mentor. <https://elibro.net/es/lc/bibunla/titulos/49347>

Siciliano, G. (2024, 20 de febrero). *Bootstrap para proyectos Java Web MVC con Spring*.

GitHub <https://github.com/gussiciliano/ghsicilianotfi>