

UNIVERSIDAD NACIONAL DE ROSARIO

INSTITUTO POLITECNICO SUPERIOR

“GRAL SAN MARTIN”

PROGRAMA ANALITICO DEL ESPACIO CURRICULAR: Adaptación del ambiente de trabajo

CURSO: Sexto Año.

PLAN DE ESTUDIOS:
EDUCACION TECNICO-PROFESIONAL DE
NIVEL SECUNDARIO.

CARRERA:
TECNICO EN INFORMATICA
PROFESIONAL Y PERSONAL

DEPARTAMENTO:
Informática

VIGENCIA AÑO: 2017

CANTIDAD DE HORAS
CATEDRA SEMANALES: 03

PLAN DE ESTUDIOS RESOLUCION C.S. N°: 596/2008

RESOLUCION MINISTERIO DE EDUCACION N°: ...

OBJETIVOS GENERALES:

Al finalizar esta materia los estudiantes estarán en condiciones de demostrar un desempeño competente resolviendo las dificultades responsablemente y con autonomía en actividades como:

- Diseño de software con metodologías de programación orientada a objetos.
- Conceptos básicos de metodologías de desarrollo de software.
- Buenas prácticas de desarrollo de software en equipos de desarrollo.

FUNDAMENTOS

Esta asignatura procura completar la formación que el alumno necesita para poder trabajar en equipos, utilizando prácticas y técnicas que son de amplia difusión en la industria de software permitiendo al alumno:

- Insertarse en cualquier equipo de trabajo profesional, conociendo las prácticas y técnicas más utilizadas.
- Conocer y dominar las herramientas de trabajo colaborativas y de productividad utilizadas normalmente en la industria, dándoles una ventaja competitiva en el mercado laboral.
- Contar con herramientas tanto técnicas como metodológicas y aplicarlas

experimentalmente para el desarrollo de aplicaciones medianas y grandes.

Para ello el alumno debe ser capaz de:

- Utilizar técnicas modernas para la solución de la problemática que representa el desarrollo de software de calidad.
- Aplicar e integrar los conocimientos adquiridos en otras cátedras al desarrollo de aplicaciones que operen con requerimientos y restricciones que enfrentarán en una implementación real.
- Desarrollar software en equipo, colaborando y cooperando como un grupo de desarrolladores, con las complejidades que esto implica.

CONTENIDOS:

Unidad 1: Introducción a la programación orientada a objetos.

- Concepto de clases y objetos.
- Visibilidad de métodos (Público, Privado, Protegido).

Unidad 2: Conceptos avanzados de programación orientada a objetos.

- Herencia, polimorfismo.
- Interfaces y clases abstractas.

Unidad 3: Herramientas y técnicas de desarrollo de software.

- Sistemas de control de versiones.
- Herramientas de integración continua.
- Testing automatizado de software.

Unidad 4: Introducción a metodologías de desarrollo de software.

- Revisión de las primeras metodologías aplicadas al desarrollo de software.
- Introducción a metodologías ágiles.

ACTIVIDAD PRÁCTICA

La práctica desarrollada a lo largo de todo el curso, tiene por objeto afianzar los nuevos conocimientos relativos a los contenidos unidad por unidad. Dependiendo de los temas abordados puede consistir en trabajos individuales o colectivos, de los siguientes tipos: a) trabajos de investigación a cargo de los alumnos con ejecución de informes, b) talleres de práctica y entrenamiento en clase, c) desarrollo de ejercitación en clase, d) desarrollo de ejercitación por internet utilizando tics del campus virtual de la unr.

METODOLOGÍA

En principio las opciones metodológicas para desarrollar los contenidos son facultades del docente y generalmente derivan de sus capacidades y recursos para ejercer el arte de la enseñanza en este tema. Sin embargo conviene establecer algunas pautas generales como recomendación, que son:

- Propiciar el aprendizaje a través del diálogo, la discusión y la resolución de problemas,
- En el desarrollo de las clases, utilizar los recursos comunicacionales orales, escritos y audiovisuales más eficaces que se tengan en el momento de desarrollar

- el curso,
- Promover acciones y trabajos individuales y/o grupales que motiven la investigación y faciliten la lectura comprensiva y crítica de textos y todo otro material de apoyo para el estudio (como por ejemplo películas, publicaciones, publicidades, etc.),
- Generar el intercambio de opiniones desde diferentes puntos de vista, para practicar el análisis reflexivo que lleve al alumno a fijar conceptos, ensayar y profundizar interpretaciones y tomar posiciones fundamentadas, en búsqueda de consensos grupales,
- Promover la práctica de la escritura técnica precisa en la ejecución de trabajos en los cuales se deba abordar el asunto de redactar con coherencia el pensamiento en forma válida para la comunicación con terceras partes, perfeccionando en la medida posible la expresión de las ideas y sus fundamentos.

EVALUACIÓN.

La evaluación de esta materia es un proceso que debe desarrollar el docente a través de todos los actos de aprendizaje y entrenamiento en que participa el alumno. En tal sentido deberán tenerse en cuenta no sólo las pruebas que reglamentariamente proponga la institución, sino también la interacción y participación de cada sujeto con la clase, así como el desempeño en los trabajos individuales y grupales, ya sean obligatorios u optativos que se elaboren durante el año y que deberían recorrer todas las vertientes de presentación/exposición que se consideren procedentes para cada tema.

BIBLIOGRAFIA:

Básica:

- Object Design: Roles, Responsibilities, and Collaborations - Rebecca Wirfs-Brock, Alan McKean - Addison-Wesley Object Technology Series.
- Domain-Driven Design: Tackling Complexity in the Heart of Software - Eric Evans – Hardcover.
- Test Driven Development: By Example – Kent Beck - The Addison-Wesley Signature Series.
- Agile Estimating and Planning – Mike Cohn - Prentice Hall

Complementaria:

- The pragmatic programmer : from journeyman to master – Andrew Hunt, David Thomas – Paperback
- Refactoring: Improving the Design of Existing Code – Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts – Addison-Wesley Prentice Hall