



Programa de la Asignatura:

Lenguajes de Programación IV



Código: 763

Carrera: **Ingeniería en Computación**

Plan: **2008**

Carácter: **Obligatoria**

Unidad Académica: **Secretaría Académica**

Curso: **Tercer Año – Segundo cuatrimestre**

Departamento: **Ingeniería**

Carga horaria total: **60 hs.**

Carga horaria semanal: **4 hs.**

Formación Experimental: **00 %**

Formación teórica: **50 %**

Formación práctica: **50 %**

Materias Correlativas Obligatorias

- **Lenguajes de Programación III (cód. 758)**
- -----
- -----

Cuerpo Docente

Henrion, Guillermo

Índice

- Fundamentación pág. 2
- Encuadre y articulación de la asignatura pág. 2
 - Encuadre dentro del Plan de Estudios pág. 2
 - Articulación Horizontal pág. 2
 - Articulación Vertical pág. 2
- Objetivos pág. 2
 - Objetivo General pág. 2
 - Objetivos Específicos pág. 2
- Contenidos mínimos pág. 2
- Programa analítico pág. 3
- Bibliografía básica pág. 3
- Bibliografía de consulta pág. 3
- Metodología del aprendizaje pág. 3
 - Desarrollo de la asignatura pág. 3
 - Dinámica del dictado de las clases pág. 4
 - Trabajos prácticos pág. 4
- Metodología de evaluación pág. 4
- Planificación pág. 4
- Información de versiones pág. 5

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 04/06/2013

Firma Docente

Firma Coordinador

1. FUNDAMENTACION

Esta asignatura está incluida en el grupo de materias que se ocupa de brindar una integración de todos los conceptos vistos por los alumnos en las asignatura de programación precedentes. Esta integración es realizada de manera eminentemente práctica, revisando y profundizando en temas conocidos, y en algunos otros temas no vistos, que surgen precisamente de esta visión. Además se enfatizará en el trabajo grupal y colaborativo, poniendo al alumno en situación similar a la que encontrará en el ámbito laboral. Cabe aclarar que el tema seleccionado por los alumnos debe ser de aplicación práctica.

2. ENCUADRE Y ARTICULACIÓN DE LA ASIGNATURA

Articulación Horizontal

Requiere una sólida formación en programación, para poder llevar adelante el proyecto a desarrollar, por lo tanto es requerida como asignatura precedente Lenguajes de Programación III

Articulación Vertical

Se inserta dentro de las materias con una visión integradora de manera de preparar al alumno para la vida laboral.

3. OBJETIVOS

Objetivo General

Es brindar una integración de los conceptos vistos en asignaturas previas, de una manera práctica, centrada en el trabajo en equipo, con participación activa del alumnado.

Objetivos Específicos

Luego de cursar esta asignatura el alumno deberá estar capacitado para:

- Realizar un proyecto informático desde su concepción hasta su entrega y mantenimiento
- Comprender la problemática de las distintas etapas de la construcción de software, y conocer las herramientas, tanto técnicas como conceptuales, que pueden ayudar a resolverlas

4. CONTENIDOS MÍNIMOS

Construcción de un proyecto completo desde su concepción hasta su entrega. Visual Story Mapping. Desarrollo evolutivo. Versionado. Patrones arquitectónicos. Calidad de software.

Robustez. Mantenibilidad. Programación de a pares. Refactoring. Cobertura de las pruebas. Conceptos de SOLID. Android. Frameworks de desarrollo.

5. PROGRAMA ANALÍTICO

Unidad 1: Presentación de la materia. Armado de grupos. Búsqueda del proyecto a desarrollar.

Unidad 2: Visual Story mapping

Unidad 3: Versionado

Unidad 4: Calidad de Software y cobertura de pruebas.

Unidad 5: Patrones arquitectónicos.

Unidad 6: Programación de a pares.

Unidad 7: Conceptos SOLID.

Unidad 8: Frameworks de desarrollo.

Unidad 9: Robustez, mantenibilidad.

6. BIBLIOGRAFÍA BÁSICA

- Testing Computer Software – Cem Kaner, Jack Falk y Hung Quoc Nguye
- Large-Scale Software Architecture – Jeff Garland y Richard Anthony

7. BIBLIOGRAFÍA DE CONSULTA

- Principles behind the Agile Manifesto - Kent Beck
- SOLID Code – Donis Marshall y John Bruno

8. METODOLOGÍA DEL APRENDIZAJE

8.a DESARROLLO DE LA ASIGNATURA

La asignatura está centrada en el desarrollo de un software en equipo de trabajo. Comenzando por la selección del tema (con técnicas para seleccionar el tema adecuado para satisfacer los gustos de todos los integrantes del grupo a fin de generar el entusiasmo adecuado en los mismos), la exposición explicando la aplicación práctica

y los posibles usuarios involucrados, el desarrollo utilizando los conceptos aprendidos durante las materias de programación precedentes, y la presentación del producto terminado.

8.b DINÁMICA DEL DICTADO DE LAS CLASES

Cada clase los alumnos deberán traer una nueva versión del producto con un subconjunto de funcionalidades completas, agregando a cada clase una mayor funcionalidad. Sobre esta presentación se realizará la revisión técnica y funcional entre todos. Se dictarán además los temas teóricos correspondientes para obtener un software de calidad.

8.c TRABAJOS PRÁCTICOS

Como se mencionó, la materia es eminentemente práctica, constando del desarrollo de una aplicación completa.

9. METODOLOGÍA DE EVALUACIÓN

9.a NORMAS DE EVALUACIÓN.

Se evaluará a los alumnos en función de las distintas presentaciones del software y la aplicación de los distintos temas teóricos dentro del mismo.

9.b RÉGIMEN DE APROBACIÓN DE LA MATERIA.

Haber completado el software con la calidad y funcionalidad comprometida.

10. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Presentación de la materia. Búsqueda de proyectos a desarrollar
Semana 2	Visual Story mapping. Elección del proyecto.
Semana 3	Presentación del proyecto utilizando Visual Story mapping
Semana 4	Revisión técnica y funcional de la primera versión del proyecto. Versionado.
Semana 5	Revisión técnica y funcional de la segunda versión del proyecto. Calidad de software.
Semana 6	Revisión técnica y funcional de la tercera versión del proyecto. Cobertura de las pruebas.
Semana 7	Revisión técnica y funcional de la cuarta versión del proyecto. Calidad de software. Calidad del software.
Semana 8	Revisión técnica y funcional de la quinta versión del proyecto. Patrones arquitectónicos.
Semana 9	Revisión técnica y funcional de la sexta versión del proyecto. Programación de a pares.
Semana 10	Revisión técnica y funcional de la séptima versión del proyecto. Frameworks de desarrollo.

Semana 11	Presentación de video lanzamiento del proyecto. Robustez.
Semana 12	Revisión técnica y funcional de la octava versión del proyecto. Conceptos SOLID
Semana 13	Revisión técnica y funcional de la novena versión del proyecto. Calidad de software.
Semana 14	Revisión técnica y funcional de la décima versión del proyecto. Robustez y mantenibilidad.
Semana 15	Exposición de los proyectos.
Semana 16	Revisión y exposición de futuras direcciones de los proyectos.
Del al de	FINAL

Información de Versiones

Nombre del Documento:	Ficha Académica de la asignatura Lenguajes de Programación IV
Nombre del Archivo	Lenguaje de programación IV – Plan 2008
Documento origen:	Lenguaje_de_programación_IV_-_Plan 2008
Elaborado por:	Guillermo Henrión
Revisado por:	
Aprobado por:	
Fecha de Elaboración:	31/05/2013
Fecha de Revisión:	
Fecha de aprobación	
Versión:	