

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA
FACULTAD DE INGENIERÍA
Escuela Profesional de Ingeniería en Informática y Sistemas

PROGRAMACIÓN DEL SÍLABO DE APRENDIZAJE

I. DATOS GENERALES

- | | |
|----------------------------|--|
| 1. Facultad | : Ingeniería |
| 2. Departamento Académico | : Ingeniería en Informática y Sistemas |
| 3. Nombre de la Asignatura | : Programación Avanzada |
| 4. Escuela Profesional | : Ingeniería en Informática y Sistemas |
| 5. Docente(s) | : MSc. HINOJOSA RAMOS, EDWIN ANTONIO
Ing. EDITH ELIZABETH ALFARO GONZALES |
| 6. Régimen | : Semestre II |
| 7. Año Académico | : 2016 |
| 8. Horas de Clase | : Teoría: 04 / Práctica: 02 |
| 9. Número de créditos | : 04 |

II. SUMILLA

APOORTE DE LA ASIGNATURA AL PERFIL PROFESIONAL

Tiene como propósito brindar al futuro profesional en ingeniería Informática y Sistemas, para comprender los fundamentos de la ciencia de la computación (lógica, teoría de autómatas y lenguajes formales, paradigmas y lenguajes de programación), así como su aplicación al desarrollo de software (métodos formales en ingeniería de software).

SUMILLA

Introducción. Ingeniería de Sistemas, Ciencia de la Informática, Ciencias de la Computación. Programación Basada en métodos formales. Diseño y complejidad algorítmica. Programación estructurada.

III. COMPETENCIA(S) DE LA ASIGNATURA

- Diferencia la ciencia de sistemas, Informática y Computación, expresando mediante un algoritmo computacional, un modelo matemático y el diseño correspondiente del diagrama de flujo y codifica en un lenguaje de programación para resolver las aplicaciones por micro-computadora.
- Conoce los sistemas, tipos de datos y los códigos que se utilizan en la ciencia de la computación.
- Identifica los diferentes recursos necesarios para el desarrollo de páginas web desde una perspectiva actual.

IV. PROGRAMACIÓN DE CONTENIDOS

SEGUNDO SEMESTRE: Del 15 de agosto al 16 de Diciembre
108

Total Horas

COMPETENCIA (Conceptual, procedimental, actitudinal)	CONTENIDOS SIGNIFICATIVOS	CRONOGRAMA (Del – al)
Analiza, diseña e implementa algoritmos de ordenación de datos en memoria interna del ordenador. Determina la complejidad de los algoritmos estudiados.	Capítulo I: ORDENACION Y BÚSQUEDA DE DATOS ↳ Ordenación interna de datos ↳ Clasificación de los algoritmos de ordenación ↳ Métodos de Ordenación interna: <ul style="list-style-type: none"> • Intercambio directo • Shaker sort • Inserción directa • Inserción binaria 	Del 15 de agosto al 23 de setiembre

<p>Desarrolla aplicaciones computacionales bajo el paradigma de la programación orientada a objetos, explotando las ventajas y potencialidades que este ofrece.</p> <p>Implementa algoritmos con E/S de datos provenientes de memoria externa utilizando la memoria interna como memoria temporal de datos. Crea flujos de datos para comunicar la memoria interna y el procesador con los dispositivos de almacenamiento externo de datos.</p> <p>Desarrolla proyectos computacionales basados en aplicaciones implementadas bajo los conceptos de la programación visual, explotando las interfaces gráficas amigables para el usuario y la reducción de tiempo de desarrollo para el programador.</p>	<ul style="list-style-type: none"> • Shell • Selección directa • Quick sort <p>↗ Búsqueda de datos</p> <ul style="list-style-type: none"> • Búsqueda secuencial • Búsqueda binaria <p>Capítulo II: PROGRAMACIÓN ORIENTADA A OBJETOS</p> <p>↗ Jerarquía de clases</p> <p>↗ Características de la POO</p> <p>↗ Protección de datos en la programación</p> <p>↗ Sobrecarga de operadores</p> <p>↗ Sobrecarga de funciones</p> <p>↗ Comunicación de datos entre objetos</p> <p>↗ Construcción de estructuras</p> <p>↗ Operación de registros de datos</p> <p>↗ Implementación de aplicaciones</p> <p style="text-align: center;">Primera Examen Parcial</p> <p>Capítulo III : Archivos</p> <p>↗ Almacenamiento de datos</p> <p>↗ Almacenamiento interno y externo de datos</p> <p>↗ Organización de archivos</p> <p>↗ Tipos de almacenamiento</p> <p>↗ Archivos de texto</p> <p>↗ Lectura y Escritura de datos en archivos</p> <p>↗ Operadores de acceso a datos en archivos</p> <p>↗ Creación de flujos de datos hacia y desde archivos</p> <p>↗ Operaciones con datos contenidos en memoria externa</p> <p>↗ Archivos binarios</p> <p>↗ Acceso secuencia a datos en archivos</p> <p>↗ Acceso aleatorio a datos en archivos</p> <p>↗ Aplicaciones</p> <p>Capítulo IV. Programación de aplicaciones avanzadas</p> <p>↗ Aplicaciones para matemática discreta</p> <p>↗ Codificadores de datos</p> <p>↗ Conceptos de la Teoría de la información</p> <p>↗ Programación modular en la programación visual</p> <p>↗ Interfaz de usuario</p> <p>↗ Programación basada en controles y eventos</p> <p>↗ Entrada y salida de datos</p> <p>↗ Desarrollo de proyectos en entorno visual</p> <p>↗ Tópicos avanzados en programación</p> <p>↗ Aplicaciones</p> <p style="text-align: center;">Segundo Examen Parcial</p>	<p>Del 26 de setiembre al 21 de octubre</p> <p>Del 24 de octubre al 18 de noviembre</p> <p>Del 21 de noviembre al 14 de diciembre</p>
--	--	---

EVALUACIÓN:

Procedimientos		Porcentaje	Ponderación
Teoría 55%	Exámenes parciales	50 %	10
	Asistencia, evaluación continua, interés y esfuerzo personal.	20 %	4
Práctica 45%	Implementación de algoritmos en Laboratorio	15 %	3
	Examen de Laboratorio y trabajos encargados	15 %	3

Observación

- ♣ La calificación será vigesimal (de 00 a 20), siendo la nota mínima de aprobación final 11 (once).
- ♣ Sólo para el promedio final se considerará la fracción 0.5 a favor del estudiante.
- ♣ Los alumnos que no se presenten a rendir sus evaluaciones en las fechas señaladas y definidas en clase tendrán la calificación 00. En caso la inasistencia sea debidamente justificada mediante los canales respectivos, según normas vigentes en la universidad, se reprogramará dicha evaluación
- ♣ *La evaluación de los estudiantes es permanente y el contenido de los exámenes es de todo el desarrollo del curso a la fecha de aplicación de la evaluación. La evaluación del estudiante es continua y no necesariamente requiere de aviso previo (excepto los exámenes parciales), siempre que las evaluaciones se realicen en los horarios establecidos para el curso.*
- ♣ El porcentaje de inasistencia a clases y/o laboratorio de más del 20% será causal de desaprobación del curso.

El promedio Final del curso se obtiene de acuerdo a los criterios de evaluación que se señalaron anteriormente.

V. METODOLOGÍA

Clases Teóricas:

- Método: Analítico y deductivo.
- Técnica : Exposición temática en aula propiciando el análisis y participación activa del alumno
- Procedimiento: Planteamiento de problemas y la búsqueda de soluciones mediante el planteamiento de algoritmos computacionales. La complejidad de los problemas será progresiva, de acuerdo al avance del curso.

Prácticas de laboratorio y/o en aula

- Método: Análisis y validación de algoritmos como resultado de la solución de problemas.
- Técnica : Implementación de algoritmos mediante el uso de lenguajes de alto nivel.
- Procedimiento : Planteamiento de problemas, elaboración de algoritmos que soluciones estos problemas, implementación y depuración de los algoritmos. Evaluación de resultados.

VI. BIBLIOGRAFÍA

CAIRO O. y GUARDATI S. (2007). *Estructura de Datos* (3ª edición). México: McGraw-Hill

CEVALLOS, Francisco (2006) Programación en C++ (3ª edición). España: Editorial RA-MA

CHAPMAN S.(2008). Matlab, Programming for Engineers (4ta. Edición). Canadá: Edit. Thomson

CORMEN T., LEISERSON C., RIVEST, R. & STEIN C. (2009). Introduction to algorithms. Third edition. The MIT Press. USA.

DROZDEK A. (2005). *Data Structures and Algorithms in C++ (2ª ed)*. Massachusetts: Grace Fujimoto

GUTIERREZ, J.M. (2006). Programación Avanzada. Alcalá – España: Universidad de Alcalá, Servicio de Publicaciones

JOYANES L. (2006). *Fundamentos de Programación: Algoritmos y Estructura de Datos* (3ª ed.). México. Editorial McGraw-Hill

- JOYANES L. (2006) C++ a su alcance (2ª edición). Universidad Pontificia de Salamanca de Madrid
- KIRANI Y. & CHAUDHURI B.B. (2007). Matlab Programming. New Delhi – India: Edit. Prentice-Hall
- SCHILD T. H. (2002). C++ Guía de referencia completa (4ª edición) Editorial Reviews
- SMITH S. (2006). Advanced GUI Development. Indianapolis – USA: Edit. Dog ear publishing.
- THE MATH WORKS (2009) MATLAB: Guía del Usuario. España, Ed. Prentice – Hall International Group
- VASQUEZ PARAGULLA, J (2005) Diseño de programación: 200 algoritmos y un proyecto de aplicación. Lima – Perú Editorial San Marcos.