

**VICERRECTORADO ACADÉMICO
MODALIDAD PRESENCIAL****PLAN DOCENTE DEL COMPONENTE ACADÉMICO****A. Datos básicos del componente académico****Nombre:** PROGRAMACIÓN DE ALGORITMOS**Titulación:** SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**Número de créditos:****Periodo académico:** Oct/2015 - Feb/2016**Conocimientos previos recomendados:**

El estudiante debe tener conocimientos de los siguiente componentes:

- Fundamentos de Programación.
- Fundamentos Informáticos.
- Lógica de la Programación.
- Computación Básica.

Importancia del componente dentro del perfil de egreso de la titulación:

Para construir aplicaciones en el caso de la programación siempre es necesario partir de algoritmos bien definidos, por ello, para este componente se a creído conveniente analizar y programar aquellos algoritmos y técnicas que deben ser de dominio de cualquier persona que desea aprender a desarrollar aplicaciones robustas y eficientes. Se a escogido como lenguaje de programación orientados a objetos Java en su última versión 8. Los temas planificados tienen que ver con: el manejo de excepciones, algoritmos de ordenamiento y búsqueda, archivos, recursividad, poliforismo, herencia y el uso de librerías o paquetes en Java.

Horario de clases presenciales:

Docente	Paralelo	Día	Aula	Horario
Rene Elizalde	A	Miércoles	NO ASIGNADA	09:00 - 12:00
Pedro Daniel Irene Robalino	B	Lunes	NO ASIGNADA	09:00 - 12:00
Pablo Vicente Torres Carrion	C	Miércoles	NO ASIGNADA	09:00 - 12:00

B. Datos básicos del(os) docente(s)**Profesor:** Pablo Vicente Torres Carrion**Título:** Máster en Desarrollo de la Inteligencia y Educación**Departamento:** Ciencias de la Computación y Electrónica**Sección departamental:** Inteligencia Artificial**Currículo profesional resumido:**

Licenciado en Ciencias de la Educación Mención Informática (2004), Ingeniero en Computación y Ciencias de la Informática(2005), Máster en Desarrollo de la Inteligencia y Educación (2010), Máster en

Redes Sociales y Aprendizaje Digital (2012), PhD Student en Comunicación y Educación en Entornos Digitales UNED (desde 2013), PhD Student en Ingeniería en Informática aplicable a la Industria ULL (desde 2014), Certificación Internacional en Aprendizaje Cooperativo (2015)

Cursos y certificaciones nacionales e internacionales en el campo de la informática, redes de datos, pedagogía y educación. Publicaciones científicas en revistas IEEE, ACM con índice SCOPUS e ISI. Participación en Congresos Científicos Nacionales e Internacionales, en calidad de ponente y expositor.

Líneas de investigación:

"Affective Computing" aplicable desde Interfacer Humano-Computador Inclusivos.

Estrategias Didácticas Digitales aplicables desde HCI con técnicas de gamificación, para niñas y niños con Síndrome de Down.

Horario de tutoría:

Día	Horario	Teléfono	Extensión
Miércoles	9:00 - 10:00	073701444	2526

Profesor: Pedro Daniel Irene Robalino

Título: Ingeniería en Sistemas Informáticos y Computación

Departamento: Ciencias de la Computación y Electrónica

Sección departamental: Tecnologías Avanzadas de la WEB

Currículo profesional resumido:

Ingeniero en "Sistemas Informáticos y Computación" Universidad Técnica Particular de Loja - Ecuador. Docente Universitario en la UTPL en asignaturas de Programación, Cálculo o Matemática desde hace 5 años. Labora en el Departamento de Computación y Electrónica en la Sección departamental Tecnologías Avanzadas de la Web y se encuentra involucrado en el desarrollo de Laboratorios Virtuales de Simulación en ramas de ingeniería civil.

Horario de tutoría:

Día	Horario	Teléfono	Extensión
Miércoles	18:00 - 19:00	07 370 1444	2531

Profesor: Rene Elizalde

Título: Magister

Departamento: Ciencias de la Computación y Electrónica

Sección departamental: Tecnologías Avanzadas de la WEB

Currículo profesional resumido:

Máster Universitario en Ingeniería del Software para la Web - Universidad de Alcalá.

Ingeniero en Sistemas - Universidad Nacional de Loja.

Docente Investigador Sección Tecnologías Avanzadas de la Web.

Analista Desarrollador de la Sección de Desarrollo de Software de la Universidad Nacional de Loja.

Docente invitado de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja.

Horario de tutoría:

Día	Horario	Teléfono	Extensión
Miércoles	18:00 - 19:00	07 370 1444	2533

Competencias genéricas (CG) de la UTPL:

- Organización y planificación del tiempo.

Competencias específicas (CE) de la titulación:

- Analizar problemas y plantear soluciones mediante métodos computacionales.
- Construir modelos y especificaciones de software que permitan validar un producto previo a su implementación.
- Implementar aplicaciones a partir de especificaciones y modelos de software utilizando estándares de documentación y de programación

Competencia del componente (CC) académico:

- Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java.
- Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos.
- Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes.
- Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.

D. Planificación general del componente académico. Estrategias de enseñanza aprendizaje planificadas para el desarrollo de competencias y para el logro de los resultados de aprendizaje esperados por parte del**Primer Bimestre****Competencias genéricas (CG) de la UTPL que corresponden al primer bimestre:**

- Organización y planificación del tiempo.

Competencias específicas (CE) de la titulación que corresponden al primer bimestre:

- Analizar problemas y plantear soluciones mediante métodos computacionales.
- Implementar aplicaciones a partir de especificaciones y modelos de software utilizando estándares de documentación y de programación

Semana 1	
Competencias del componente académico	- Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java.
Contenidos	Clases y Objetos en Java <ul style="list-style-type: none">• Revisión: Qué es programación orientada a objetos (Clases, Objetos, métodos y variables de instancia).?• Atributos (primitivos/por referencia), niveles de acceso (public, private, protected, static) e inicialización (final, this).
Resultados de aprendizaje (RA)	<ul style="list-style-type: none">- Comprende qué son las clases, objetos, métodos y variables de instancia (atributos de una clase/objeto).- Declara e implementa clases, métodos y atributos para crear objetos.- Usa los objetos que referencian clases y miembros (atributos y métodos) de acuerdo a sus niveles de acceso e inicialización.- Diferencia el concepto de variables de instancia de una clase y las variables locales de un método.- Conoce las diferencias entre los tipos primitivos y los tipos por referencia.

Actividades en clase	<ul style="list-style-type: none"> - Presentación de la Asignatura, plan de estudios y metodología de trabajo del ciclo académico. - Pre-evaluación de conocimientos requeridos para la asignatura a través del Entorno Virtual de Aprendizaje. - Clase Revisión de Programación Orientada a Objetos, conceptos básicos. - Desarrollo de ejercicios prácticos y reales aplicando Programación Orientada a Objetos (individual y grupal). - Planteamientos de temas para el proyecto práctico.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 3 y 8. Clases y Objetos: Un análisis mas detallado, del libro base. - Instalación y configuración del ambiente de desarrollo (NetBeans 8.0 y Jdk 1.8) - Implementación, ejecución y pruebas de los ejercicios propuestos en el texto base: - Análisis y desarrollo de ejercicios extraclase. - Revisión de videos y lecturas complementarias del EVA.
Horas de trabajo	6.00

Semana 2	
Competencias del componente académico	- Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java.
Contenidos	Clases y Objetos en Java Métodos, Propiedades (set, get), constructores y sobrecarga. Encapsulamiento, ocultación y paquetes.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Utiliza los constructores (sobrecargados o no) para asegurar que los datos de un objeto se inicialicen cuando se crea el objeto. - Comprende el concepto de encapsulamiento y ocultamiento de datos. - Organiza las clases en paquetes para promover la reutilización. - Implementa constructores y métodos sobrecargados.
Actividades en clase	<ul style="list-style-type: none"> - Review de la Clase anterior: Conceptos básicos de POO. - Clase Magistral: Métodos, propiedades, constructores, sobrecarga, encapsulamiento, ocultamiento y paquetes. - Desarrollo de ejercicios prácticos y reales aplicando POO (individual y grupal). - Revisión de propuestas y avances del proyecto.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 8. Clases y Objetos: Un análisis mas detallado, del libro base. - Revisión de videos y lecturas complementarias del EVA. - Implementación, ejecución y pruebas de los programas del Texto Base: - Análisis y desarrollo de ejercicios extraclase.
Horas de trabajo	6.00

Semana 3	
Competencias del componente académico	- Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java.
Contenidos	Clases y Objetos en Java: Herencia Superclases y subclases. Métodos y Propiedades (set, get), constructores, sobrecarga Encapsulamiento, ocultación y paquetes
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Comprende cómo la herencia fomenta la reutilización de software. - Entiende qué son las superclases y las subclases y utiliza los miembros de superclases mediante super. - Utilizar la palabra clave extends para crear una clase que herede los atributos y comportamientos de otra clase. - Comprende y usa el modificador de acceso protected para dar a los métodos de la subclase acceso limitado a los miembros de la superclase. - Comprende y utiliza los constructores en las jerarquías de herencia.

Actividades en clase	<ul style="list-style-type: none"> - Review de la clase anterior: Conceptos básicos de POO. - Clase Magistral: La herencia, miembros protected y constructores. - Desarrollo de ejercicios prácticos y reales aplicando POO-Herencia (individual y grupal). - Evaluación teórico-Práctica de POO conceptos básicos vía EVA.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 9. Programación Orientada a Objetos: Herencia. Libro base. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: "EmpleadoPorComisión" - Desarrollo de ejercicios prácticos y reales aplicando POO: Herencia. - Desarrollo del proyecto práctico.
Horas de trabajo	6.00

Semana 4	
Competencias del componente académico	- Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java.
Contenidos	Clases y Objetos en Java: Polimorfismo <ul style="list-style-type: none"> • Fundamentos de Polimorfismo • Demostración del comportamiento Polimorfo. • Clases y métodos abstractos.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Comprende el concepto de polimorfismo. - Aprende a utilizar métodos sobrescritos para llevar a cabo el polimorfismo - Distingue y declara clases/métodos abstractas y concretas. - Aprender la manera en que el polimorfismo hace que los sistemas puedan extenderse y mantenerse
Actividades en clase	<ul style="list-style-type: none"> - Review de la clase anterior: Conceptos básicos de POO: Herencia - Clase Magistral: Polimorfismo - Desarrollo de ejercicios prácticos y reales aplicando POO-Polimorfismo - Evaluación teórico-Práctica de POO: Herencia y Polimorfismo vía EVA. - Revisión de avances del proyecto práctico.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 10. Programación Orientada a Objetos: Polimorfismo. Libro base. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: "Empleado" - Desarrollo de ejercicios prácticos y reales aplicando POO: Polimorfismo.
Horas de trabajo	6.00

Semana 5	
Competencias del componente académico	- Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos.
Contenidos	Gestión de Excepciones en java <ul style="list-style-type: none"> • Fundamentos sobre el manejo de excepciones. • Las excepciones en el API de Java. • Uso de try-catch-finally.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Define los fundamentos sobre el manejo de excepciones. - Reconoce escenarios que pueden generar excepciones dentro del programa. - Identifica ventajas/desventajas, buenas/malas prácticas al implementar manejo de excepciones. - Conoce e implementa la estructura try-catch-finally y throw para el tratamiento de excepciones.
Actividades en clase	<ul style="list-style-type: none"> - Clase Magistral: Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando control de excepciones.

Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capitulo nro. 13. Excepciones. Libro base. - Revisión de vídeos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando Manejo de Excepciones. - Continua desarrollo del proyecto.
Horas de trabajo	6.00

Semana 6	
Competencias del componente académico	- Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos.
Contenidos	Gestión de Excepciones en java (continua) <ul style="list-style-type: none"> • La instrucción throw-throws. • Limpieza de la pila y excepciones encadenadas. • Generar excepciones propias.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Lanzar excepciones (throw-throws) que deben ser manejadas con try-catch. - Genera, atrapa y maneja sus propias excepciones en escenarios prácticos de aplicación.
Actividades en clase	<ul style="list-style-type: none"> - Review de la clase anterior: Manejo de excepciones - Clase Magistral: Manejo de Excepciones (continua) - Desarrollo de ejercicios prácticos y reales aplicando Control de Excepciones. - Presentación de avances del proyecto-práctico.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capitulo nro. 13. Excepciones. Libro base. - Revisión de vídeos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando Manejo de Excepciones.
Horas de trabajo	6.00

Semana 7	
Competencias del componente académico	<ul style="list-style-type: none"> - Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java. - Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos.
Contenidos	Revisión y conclusiones de: Programación Orientada a Objetos. Paquetes API-Java Herencia Polimorfismo y Manejo de Excepciones.
Resultados de aprendizaje (RA)	- Aplica conceptos de programación orientada a objetos en función de las necesidades a resolver en una determinada problemática
Actividades en clase	<ul style="list-style-type: none"> - Review de temas: POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando Control POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones. - Defensa de avances de proyectos.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Review de los temas del 1er. bimestre: POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando Control POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones.
Horas de trabajo	6.00

Semana 8	
Competencias del componente académico	<ul style="list-style-type: none"> - Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java. - Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos.
Contenidos	Evaluación de los contenidos del 1er. bimestre. POO-Conceptos básicos. POO-Herencia. POO-Polimorfismo. Manejo de Excepciones.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Entiende, programa y aplica los conceptos de POO, paquetes, herencia y polimorfismo para la eficiencia y reutilización de Clases. - Implementa manejo de excepciones para lograr aplicaciones robustas y tolerantes a fallos.
Actividades en clase	- Evaluación de competencias.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Review de los temas del 1er. bimestre: POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones. - Desarrollo de ejercicios prácticos y reales aplicando Control POO, Paquetes, API-Java, Herencia, Polimorfismo y Manejo de Excepciones.
Horas de trabajo	6.00

Total de horas de trabajo del primer bimestre:

Horas presenciales: 24.00 Horas extraclase: 48.00

Fechas importantes (actividades académicas):

- 1 de octubre: Inicio general de la actividad académica.
- 16-20 de noviembre: Review de todo el contenido del 1er. bimestre
- 23-27 de noviembre: Evaluación de competencias y resultados de aprendizaje.
- 26-30 de octubre. Revisión de avances del proyecto.
- 2 de diciembre: Publicación de resultados del 1er. bimestre.
- 16-20 de noviembre: Revisión de Proyectos Final - Primer Bimestre

Segundo Bimestre

Competencias genéricas (CG) de la UTPL que corresponden al segundo bimestre:

- Organización y planificación del tiempo.

Competencias específicas (CE) de la titulación que corresponden al segundo bimestre:

- Analizar problemas y plantear soluciones mediante métodos computacionales.
- Construir modelos y especificaciones de software que permitan validar un producto previo a su implementación.
- Implementar aplicaciones a partir de especificaciones y modelos de software utilizando estándares de documentación y de programación

Semana 1	
Competencias del componente académico	<ul style="list-style-type: none"> - Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes.
Contenidos	Programación de ficheros con Java <ul style="list-style-type: none"> • Conceptos preliminares sobre los datos. • Flujos en Java.

Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Conoce los fundamentos sobre archivos y la jerarquía de datos (bit, caracter, campo, registro y archivo) - Comprende la jerarquía de clases de flujos de entrada/salida en Java. - Obtiene información acerca de un archivo y directorio utilizando la clase File . - Conoce las diferencias entre los archivos de texto y los archivos binarios. - Comprende el procesamiento de archivos de acceso secuencial.
Actividades en clase	<ul style="list-style-type: none"> - Presentación general de los contenidos y directivas para el 2do. bimestre. - Clase magistral: Archivos y flujos en Java. - Desarrollo de ejercicios prácticos aplicando flujos de entrada salida archivos.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 14. Archivos y flujos. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Acceso a archivos. - Desarrollo de ejercicios prácticos aplicando flujos de entrada salida archivos.
Horas de trabajo	6.00

Semana 2	
Competencias del componente académico	<ul style="list-style-type: none"> - Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la más eficiente para la entrada/salida de datos persistentes.
Contenidos	Programación de ficheros con Java (continua) <ul style="list-style-type: none"> • Escritura y lectura de datos en ficheros. • Uso de las clases FileInputStream y FileOutputStream.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Conoce y utiliza las clases Scanner y Formatter para procesar archivos de texto. - Conoce y utiliza las clases FileReader y FileWriter para leer de, y escribir en, archivos (flujos basados en caracteres). - Conoce y utiliza las clases FileInputStream y FileOutputStream para leer de, y escribir en, archivos (flujos basados en bytes). - Conoce y utilizar las clases ObjectInputStream y ObjectOutputStream para leer de, y escribir objetos en, archivos.
Actividades en clase	<ul style="list-style-type: none"> - Clase magistral: Archivos y flujos en Java. (continuidad) - Desarrollo de ejercicios prácticos aplicando flujos de entrada salida archivos. - Revisión de avances del proyecto.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Lectura comprensiva del capítulo nro. 14. Archivos y flujos. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Acceso - archivos. - Desarrollo de ejercicios prácticos aplicando flujos de entrada salida archivos.
Horas de trabajo	6.00

Semana 3	
Competencias del componente académico	<ul style="list-style-type: none"> - Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Algoritmos recursivos <ul style="list-style-type: none"> • Introducción y conceptos básicos de recursividad • Aplicación de la recursividad
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Comprende, codifica y prueba algoritmos recursivos para incrementar la eficiencia en ejecución los programas. - Determina el caso base y el paso de recursividad necesarios para generar un algoritmo recursivo. - Conoce cómo las aplicaciones maneja las llamadas (ida/retorno) a métodos recursivos directa o indirectamente. - Diferencia entre recursividad e iteración y sabe cuando aplicar eficientemente una o otra.

Actividades en clase	- Clase magistral: Recursividad - Desarrollo de ejercicios sobre Recursividad
Horas de trabajo	3.00
Actividades extraclase	- Lectura comprensiva del capítulo nro. 15. Recursividad. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL). - Implementación, ejecución y pruebas de los programas del Texto Base: Factorial, Fibonacci, torres de Hanoi - Desarrollo de ejercicios prácticos y reales aplicando recursividad.
Horas de trabajo	6.00

Semana 4	
Competencias del componente académico	- Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Algoritmos sobre arreglos <ul style="list-style-type: none"> • Estructura de un array. • Manejo de la Estructura For optimizada. • Arreglos multidimensionales. • Algoritmos de búsqueda.
Resultados de aprendizaje (RA)	- Conoce y aplica los fundamentos de los arreglos unidimensionales y multidimensionales. - Conoce y utiliza la instrucción for mejorada para iterar a través de los elementos de un arreglo. - Pasa a métodos y recibe/recorre correctamente los arreglos y matrices de una lista de parametros. - Busca valores en arreglos usando la búsqueda lineal y binaria.
Actividades en clase	- Clase Magistral: Arreglos/matrices, for mejorado y algoritmos de búsqueda - Desarrollo de ejercicios prácticos y reales aplicando algoritmos de búsqueda lineal y binario. - Revisión de avances del proyecto.
Horas de trabajo	3.00
Actividades extraclase	- Lectura comprensiva del capítulo nro. 7 y 16. Arreglos / Búsqueda y Ordenamiento. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Búsqueda lineal y binaria. - Desarrollo de ejercicios prácticos aplicando un algoritmo de búsqueda.
Horas de trabajo	6.00

Semana 5	
Competencias del componente académico	- Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Algoritmos sobre arreglos (continua) <ul style="list-style-type: none"> • Algoritmos de ordenamiento . • Operaciones adicionales en un arreglo.
Resultados de aprendizaje (RA)	- Ordena arreglos usando los algoritmos iterativos de ordenamiento por selección, inserción y burbuja. - Opera arreglos/matrices a través de paquetes del API de Java, y de ser necesario los amplía o mejora.
Actividades en clase	- Clase magistral: Métodos de ordenamiento y otras operaciones sobre arreglos. - Desarrollo de ejercicios prácticos y reales aplicando algoritmos de ordenamiento.
Horas de trabajo	3.00
Actividades extraclase	- Lectura comprensiva del capítulo nro.16. Búsqueda y Ordenamiento. - Revisión de videos y otros Recursos Educativos Abiertos (REA's) vía Entorno Virtual de Aprendizaje (EVA-UTPL) - Implementación, ejecución y pruebas de los programas del Texto Base: Búsqueda lineal y binaria. - Desarrollo de ejercicios prácticos aplicando un algoritmo de ordenamiento.

Horas de trabajo	6.00
-------------------------	------

Semana 6	
Competencias del componente académico	<ul style="list-style-type: none"> - Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes. - Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Review de los contenidos del 2do. bimestre: Arreglos/matrices Algoritmos de ordenamiento, busqueda, recursivos Finalmente archivos en java.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Codifica aplicaciones con lectura/escritura de datos persistentes en archivos. - Implementa métodos de ordenamiento y búsqueda sobre arreglos de datos primitivos u objetos.
Actividades en clase	<ul style="list-style-type: none"> - Review de los contenidos del 2do. bimestre: archivos en java, arreglos/matrices, algoritmos de ordenamiento, busqueda y recursivos. - Desarrollo de ejercicios prácticos y reales aplicando archivos y algoritmos de ordenamiento, búsqueda y recursividad.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Review de los contenidos del 2do. bimestre: archivos en java, arreglos/matrices, algoritmos de ordenamiento, busqueda y recursivos. - Desarrollo de ejercicios prácticos y reales aplicando archivos y algoritmos de ordenamiento, búsqueda y recursividad.
Horas de trabajo	6.00

Semana 7	
Competencias del componente académico	<ul style="list-style-type: none"> - Entender y aplicar los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java. - Identificar y aplicar las estrategias de control de excepciones en flujos de código propensos a fallos. - Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes. - Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Defensa del Proyecto de fin de ciclo. Fase de desarrollo. Fase de implementación. Fase de pruebas. Fase de documentación.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Entiende y aplica los conceptos de Programación Orientada a Objetos, Herencia y Polimorfismo en el desarrollo de programas en lenguaje Java. - Identifica y aplica las estrategias de control de excepciones en flujos de código propensos a fallos. - Evalua técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes. - Conoce, comprende y programa algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Actividades en clase	<ul style="list-style-type: none"> - Defensa del proyecto final. - Entrega de documentación.
Horas de trabajo	3.00
Actividades extraclase	- Desarrollo y redacción del informe del Proyecto de fin de ciclo. Face de: desarrollo, implementación, pruebas y documentación.
Horas de trabajo	6.00

Semana 8	
Competencias del componente académico	<ul style="list-style-type: none"> - Evaluar técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes. - Conocer, comprender y programar algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Contenidos	Evaluación de contenidos y competencias sobre los temas del 2do. bimestre: Archivos en java, arreglos/matrices, algoritmos de ordenamiento, búsqueda y recursivos.
Resultados de aprendizaje (RA)	<ul style="list-style-type: none"> - Evalua técnicas de gestión de archivos de acceso secuencial adoptando la mas eficiente para la entrada/salida de datos persistentes. - Conoce, comprende y programa algoritmos de búsqueda, ordenamiento y recursividad sobre arreglos u otras estructuras.
Actividades en clase	- Evaluación de contenidos y competencias.
Horas de trabajo	3.00
Actividades extraclase	<ul style="list-style-type: none"> - Review de los contenidos del 2do. bimestre: archivos en java, arreglos/matrices, algoritmos de ordenamiento, búsqueda y recursivos. - Desarrollo de ejercicios prácticos y reales aplicando archivos y algoritmos de ordenamiento, búsqueda y recursividad.
Horas de trabajo	6.00

Total de horas de trabajo del segundo bimestre:

Horas presenciales: 24.00 Horas extraclase: 48.00

Fechas importantes (actividades académicas):

- 30 noviembre - 4 de diciembre: Inicio de la actividad académica del 2do. bimestre.
- 25-29 de enero: Evaluación del 2do. bimestre.
- 18-22 de enero: Entrega y defensa final de proyecto.
- 10-16 de febrero: Evaluación final (supletorio).

E. Evaluación del componente académico

Primer Bimestre

Instrumento	Peso %	Puntos
Tareas Extraclase	10.0	2.0
Talleres (grupales, individuales)	10.0	2.0
2 evaluaciones parciales	10.0	2.0
Proyecto de investigación y desarrollo	30.0	6.0
Evaluación teórico – práctica del 1er. bimestre	40.0	8.0
TOTAL:	100.0%	20.0

Segundo Bimestre

Instrumento	Peso %	Puntos
Tareas Extraclase	10.0	2.0
Talleres (grupales, individuales)	10.0	2.0
2 evaluaciones parciales	10.0	2.0
Proyecto de investigación y desarrollo	30.0	6.0
Evaluación teórico – práctica del 2do. bimestre	40.0	8.0
TOTAL:	100.0%	20.0

F. Examen final

Tiempo	Instrumento	Puntos
Primer Bimestre	Evaluación teórico – práctica.	6
Primer Bimestre	Proyecto de investigación y desarrollo.	4
Segundo Bimestre	Evaluación teórico – práctica.	6
Segundo Bimestre	Proyecto de investigación y desarrollo.	4

G. Recursos a utilizar para el desarrollo del componente académico

Bibliografía básica

Nombre del Texto:

Dietel, P. y Dietel, H. (2012). Java Como Programar. Mexico: Pearson Prentice Hall.

Información general del texto:

Se ha seleccionado este texto por las ventajas pedagógicas y técnicas que brinda al estudiante. Tratan temas relacionados con el lenguaje desde conceptos, estructuras, sintaxis básicas, hasta el uso de tecnologías avanzadas. De forma tal, este recurso servirá al estudiante no solo en el presente ciclo sino durante el resto de su carrera, adicionalmente en el texto se puede encontrar una gran cantidad de ejercicios los cuales han sido explicados a detalle por parte de los autores.

¿El texto está disponible en la biblioteca general física de la UTPL?

Si

Bibliografía complementaria

Nombre del Texto:

Montero, M. (2011). Java 7. España: Ediciones Anaya.

Información general del texto:

Java 7 es una obra puntual que presenta los conceptos de Programación Orientada a Objetos de manera sencilla y ejemplificados, desde estructuras de control básicas hasta tecnologías avanzadas de programación.

¿El texto está disponible en la biblioteca general física de la UTPL?

Si

Nombre del Texto:

Barnes, D. (2012). Programación Orientada a Objetos usando BlueJ. España: Pearson.

Información general del texto:

Este libro es muy completo para iniciar en la programación con java usando la Orientación a Objetos y el IDE BlueJ, que se presenta al estudiantes como una alternativa al Netbeans y a Eclicpse. La obra presenta programas muy bien explicados y aplicando los conceptos de desarrolla en cada capitulo.

¿El texto está disponible en la biblioteca general física de la UTPL?

Si

Biblioteca virtual

Nombre de la base de datos	Link
----------------------------	------

Recursos Educativos Abiertos

Nombre de la base de datos	Link
Programación en Java I [OCW de la UPM]	García, B; Arranz, S (2009). Programación en Java I. Madrid UPM. Disponible en: http://ocw.upm.es/lenguajes-y-sistemas-informaticos/programacion-en-java-i
Conceptos y ejemplos de Fundamentos de la Programación [OCW de la UPM]	Jaén J. Martínez R. García A. y otros. Conceptos y ejemplos de Fundamentos de la Programación. Madrid UPM http://ocw.upm.es/ciencia-de-la-computacion-e-inteligencia-artificial/fundamentos-programacion
Introducción a la Programación en Java. [OCW de la UPV]	Sarriguren A. Serrano J. Alonso L. Introducción a la Programación en Java. Madrid Universidad del País Vasco. Disponible en: http://ocw.ehu.es/course/view.php?id=116
A Gentle Introduction to Programming Using Python	http://carbono.utpl.edu.ec:8080/OERRanking/resource/recordserver?urlTarget=http%3A%2F%2Fserendipity.utpl.edu.ec%2Fserendipity%2Focw%2Fbrowse%3Fdetails%3D1%26fq%3D%3A832&source=planesacad&resourceuri=http%3A%2F%2Fserendipity.utpl.edu.ec%2Fserendipity%2Focw%2Fbrowse%3Fdetails%3D1%26fq%3D%3A832
Introduction to Algorithms	http://carbono.utpl.edu.ec:8080/OERRanking/resource/recordserver?urlTarget=http%3A%2F%2Fserendipity.utpl.edu.ec%2Fserendipity%2Focw%2Fbrowse%3Fdetails%3D1%26fq%3D%3A781&source=planesacad&resourceuri=http%3A%2F%2Fserendipity.utpl.edu.ec%2Fserendipity%2Focw%2Fbrowse%3Fdetails%3D1%26fq%3D%3A781

Enlaces web

Otros recursos

UTPL FICHEROS EN JAVA [(INFORMÁTICA)(PROGRAMACIÓN DE ALGORITMOS)] - <https://www.youtube.com/watch?v=sYhgL30IDIQ>

UTPL PROGRAMACIÓN ORIENTADA A OBJETOS [(INFORMÁTICA)(PROGRAMACIÓN DE ALGORITMOS)] - <https://www.youtube.com/watch?v=pRlInyrC0Reg>

UTPL RECURSIVIDAD EN JAVA [(INFORMÁTICA)(PROGRAMACIÓN DE ALGORITMOS)] - <https://www.youtube.com/watch?v=Un5Vhz70kxk>

UTPL ARREGLOS EN JAVA [(INFORMÁTICA)(PROGRAMACIÓN DE ALGORITMOS)] - <https://www.youtube.com/watch?v=8FVNJc4AhAM>

NOTA: Durante todo el bimestre el docente deberá utilizar un portafolio docente digital donde respalde todo el material utilizado para el desarrollo del componente académico, sean diapositivas, pruebas, recursos, etc.) El uso del EVA es obligatorio para las dos modalidades.

Elaborado por:

.....
Pablo Vicente Torres Carrion

.....
Pedro Daniel Irene Robalino

.....
Rene Elizalde

Revisado por:

.....
Responsable de Sección Departamental

Aprobado por el Consejo de Departamento, según acta Nro._____ de fecha:_____