

Nombre de asignatura
Planificación Ciclo lectivo 2022

Datos administrativos de la asignatura			
Departamento:	Sistemas de Información	Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN
Asignatura:	Programación avanzada		
Nivel de la carrera	4	Duración	48
Bloque curricular:	Electivas		
Carga horaria presencial semanal:	3	Carga Horaria total:	48
Carga horaria no presencial semanal (si correspondiese)		% horas no presenciales (si correspondiese)	
Profesor/es Titular/Asociado/Adjunto:	Mg. Vanzetti Juan José	Dedicación:	Simple
Auxiliar/es de 1º/JTP:		Dedicación:	

Presentación, Fundamentación

Con el correr de los años las habilidades específicas en el desarrollo de sistemas de información ha cobrado una creciente importancia en la industria del Software. El mercado actual y el perfil de la carrera, demanda en nuestros egresados conocimientos sobre programación y construcción de sistemas, en forma manual, con uso de asistentes y con orientación a programación orientada a eventos. Esta cátedra culmina el ciclo de conocimientos de los alumnos que, hasta el momento de cursarla, sólo han realizado el análisis y el diseño de un sistema de información, aportando los conocimientos para la construcción del mismo.

En las asignaturas obligatorias de los dos primeros años del Área Programación, se abordan gran cantidad de conocimientos y actitudes que son puestos en práctica en la construcción de sistemas de información. A la vez, no se tratan en profundidad muchos otros aspectos específicos y particulares, los cuales necesitan ser tratados en forma planificada y detenida, para poder desarrollar en los alumnos las destrezas necesarias para afrontar la participación adecuada en la dirección o en la ejecución del desarrollo de sistema de información.

- **Relación de la asignatura con el perfil de egreso.** (Describir la relación y los aportes de la asignatura al perfil de egreso).
- **Relación de la asignatura con los alcances del título.** (Describir la relación y los aportes de la asignatura con los alcances del título).

--

Relación de la asignatura con las competencias de egreso de la carrera

Detallar, en la tabla siguiente, la relación de la asignatura con las competencias de egreso específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera. Indicar a cuáles competencias de egreso tributa (aportes reales y significativos de la asignatura) y en qué nivel (0=no tributa, 1=bajo, 2=medio, 3=alto). Agregar un comentario general de justificación. (Este detalle se integrará en una matriz de tributación de la carrera, dictada en la Facultad Regional, en la cual se explicita el desarrollo de las competencias específicas y genéricas de la carrera y el nivel en que tributa cada asignatura).

Competencias específicas de la carrera (CE)	Competencias genéricas tecnológicas (CT)	Competencias genéricas sociales, políticas y actitudinales (CS)
CE1: Especificar, proyectar y desarrollar software (Alta)	CT1: Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas (Media)	CS1: Desempeñarse de manera efectiva en equipos de trabajo (Media)

Propósito

El propósito de esta materia es que el alumno/a construya programas en un entorno visual orientado a objetos. Además, que le permita gestionar datos de una base de datos relacional de manera consistente, desarrolle listados y reportes estadísticos que colaboren en la toma de decisiones del sistema de información, fortaleciendo el trabajo en equipo mediante una aproximación a una experiencia profesional concreta.

Objetivos establecidos en el Diseño Curricular

- Introducir y capacitar al alumno en la construcción y programación avanzada de sistemas de información
- Realizar prácticas avanzadas en herramientas de programación
- Adquirir conocimientos que le permitan construir programas en forma manual o con uso de asistentes en entornos visuales
- Comprender los roles de los integrantes de un equipo en un proyecto de desarrollo de software
- Asimilar el avance de diferentes tecnologías de hardware y software en el ámbito de las técnicas de programación avanzada.
- Aprender diferentes técnicas modernas de programación avanzada que permitan utilizarse en diferentes plataformas de hardware y software.

Resultados de aprendizaje

- RA 1 Reconocer las sentencias necesarias para recuperar y modificar la base de datos donde persisten los datos de una aplicación en el marco de bases de datos relacionales.
- RA 2 Emplear los conceptos del paradigma orientados a objetos para diseñar e implementar programas mediante el lenguaje Java.
- RA 3 Adquirir las habilidades para desarrollar servicios web basados en el standard WebApi orientado a integrar el front end de una aplicación web con el back end desarrollado en un entorno JAVA.
- RA4 Confeccionar Test Set de ejecución de pruebas con la finalidad de poder organizarlas.

Asignaturas correlativas previas

Para cursar debe tener cursada:

Para cursar debe tener aprobada:

Para rendir debe tener aprobada:

Asignaturas correlativas posteriores

Programa analítico, Unidades temáticas

CONTENIDOS Por ejes temáticos

Eje Nº 1 Clases

- Clases y miembros de clases
- Control de Acceso a una Clase
- Control de flujo
- Herencia
- Constructores y Destruyores
- Mensajes
- Encapsulamiento
- Polimorfismo
- Interfaces

Eje Nº 2 Acceso a base de datos

- Introducción a la Persistencia
- Mapeo objeto/relacional
 - Mapeo de las entidades
 - Mapeo de las relaciones
 - Tipos de cascadas
- Gestión de la persistencia
 - Transacción típica
 - Configuración e Inicialización de Hibernate
- Bloqueos
- Introducción a HQL
- Criteria
- Generación de Reportes

Eje Nº 3 Conceptos de Programación

- Recursividad
- Iteración

Ordenamiento y Búsqueda

Eje Nº 4 Estructuras dinámicas

Concepto de pilas, listas y colas junto a sus operaciones básicas implementando iteración y recursión

Eje Nº 5 Hilos y Multi hilos

Introducción a hilos
Estado de los hilos
Propiedades de los hilos
Sincronización.

Eje Nº 6 Patrones de diseño en el lenguaje de programación

Concepto
Clasificación
Implementación

Eje Nº 7 Excepciones, depuración y evaluación

Manejo de errores
Clasificación de excepciones
Declaración de excepciones comprobadas
Creación de clases de Exception
Captura de Excepciones
Cobertura estructural
Cobertura de sentencias
Cobertura de ramas
Cobertura de decisiones
Herramientas de testeo

Eje Nº 8

Concepto de Web API.. ¿Qué es REST?. Por qué usar Web API. • URL, sintaxis de la URI. Verbo GET análisis del Request y Response.
Códigos de respuesta Análisis del encabezado, cuerpo del mensaje.
Formato JSON. JSON vs XML.
Esquema de funcionamiento de una aplicación RESTFull.

Metodología de enseñanza

ESTRATEGIAS METODOLÓGICAS

La cátedra contempla las siguientes estrategias metodológicas:

Exposición oral del profesor y los ayudantes de cátedra en las cuales se enfatizara el uso de la lectura anticipada de los temas a tratar, elaboración de mapas conceptuales e integración de contenidos.

El dictado de la materia se complementa con la presentación y análisis de publicaciones, presentación y análisis de videos, resolución de guías prácticas y la elaboración de un trabajo de investigación.

Clases teóricas

Durante el dictado de las clases teóricas se pretende que el alumno vaya adquiriendo de manera gradual conocimientos referidos a la Ingeniería de Software de acuerdo a diferentes aspectos tratados en las unidades temáticas

Clases Prácticas

Por medio de las actividades prácticas planteadas por el docente, se procura la comprensión y asimilación de los conceptos teóricos planteados por el mismo. Durante el desarrollo de todas las unidades temáticas se plantearán actividades prácticas que promuevan el desarrollo de aptitudes del alumno en la materia.

Recomendaciones para el estudio

--

Metodología de evaluación

EVALUACIÓN

Para llevar a cabo la evaluación desde el enfoque formativo es necesario que el docente incorpore en el aula estrategias de evaluación congruentes con las características y necesidades individuales de cada alumno y las colectivas del grupo. El cumplimiento en la entrega en tiempo y forma de las actividades prácticas. En caso que las actividades contengan errores, los alumnos tendrán la posibilidad de ir corrigiéndolos a media que transcurre el cuatrimestre en entregas sucesivas. Al finalizar el cuatrimestre, los alumnos deberán haber concluido correctamente todas las actividades prácticas. Se evaluarán además a los alumnos en instancias parciales

Momentos: Evaluación Diagnostica, Evaluación formativa o continua

Instrumentos:

- 2 parcial práctico y un recuperatorio
- Resolución de guías de trabajos prácticos.

Criterios de:

A) Regularidad:

Para obtener la regularidad de la materia el alumno deberá:

- Tener el 75% de asistencias
- Tener completas y entregadas las guías de trabajos prácticos.
- Tener aprobados los parciales prácticos y teóricos con nota promedio 6 o superior.
- Podrá recuperar una instancia de parcial práctico y uno teórico.

B) Promoción:

Para obtener la promoción directa de la materia el alumno deberá:

- Tener el 75% de asistencias
- Tener completas y entregadas las guías de trabajos prácticos.
- Tener aprobados los parciales prácticos y teóricos con nota 8 o superior.

Podrá recuperar una instancia de parcial práctico y uno teórico

Cronograma de clases/trabajos prácticos/exámenes (tentativo)

Cronograma:

Clase Nº	Día	Fecha	Tema	Observaci
1	Lunes	15/08/2022	Feriado	
2	Lunes	22/08/2022	Presentación materia. Unidad 1: Clases y miembros de clases. Control de Acceso a una Clase. Control de flujo. Herencia. Constructores y destructores	
3	Lunes	29/08/2025	Unidad 1: Introducción a la Persistencia. Mapeo objeto/relacional. Mapeo de las entidades. Mapeo de las relaciones. Tipos de cascadas. Gestión de la persistencia. Transacción típica .Configuración e Inicialización de Hibernate. Bloqueos	
4	Lunes	05/09/2022	Unidad 2: Mapeo objeto/relacional. Mapeo de las entidades. Mapeo de las relaciones. Tipos de cascadas. Gestión de la persistencia. Transacción típica. Configuración e Inicialización de Hibernate. Bloqueos.	

5	Lunes	12/9/2022	Unidad 2: Introducción a HQL. Criterios		
6	Lunes	19/09/2022	Unidad 2: Generación de Reportes		
7	Lunes	26/09/2022	Ejercitación Reportes		
8	Lunes	3/10/2022	Unidad 3: Recursividad. Iteración. Ordenamiento y Búsqueda		
9	Lunes	10/10/2022	Feriado		
10	Lunes	17/10/2022	Unidad 4: Concepto de pilas y colas junto a sus operaciones básicas implementando iteración y recursión		
11	Lunes	24/10/2022	Ejercitación		
12	Lunes	31/10/2022	Unidad 5: Introducción a hilos. Estado de los hilos. Propiedades de los hilos. Sincronización		
13	Lunes	7/11/2022	Unidad 6: Patrones. Concepto. Clasificación		
14	Lunes	14/11/2022	Unidad 7: Manejo de errores. Clasificación de excepciones. Declaración de excepciones comprobadas. Creación de clases de Exception. Captura de Excepciones Cobertura estructural. Cobertura de sentencias. Cobertura de ramas. Cobertura de decisiones. Herramientas de testeo		
15	Lunes	21/11/2022	Feriado		
16	Lunes	28/11/2022	Unidad 8 Concepto de Web API.. ¿Qué es REST?. Por qué usar Web API. • URL, sintaxis de la URI. Verbo GET análisis del Request y Response. Códigos de respuesta Análisis del encabezado, cuerpo del mensaje. Formato JSON. JSON vs XML. Esquema de funcionamiento de una aplicación RESTFull.		

Recursos necesarios

Recopilación de apuntes de la Cátedra

Revistas y publicaciones científicas específicas de Programación Avanzada
Ejemplos de código fuentes de los diferentes temas tratados en la cátedra

Referencias bibliográficas (citadas según Normas APA)

Bibliografía:

a) Obligatoria o básica:

- Patrones de Diseño aplicados a Java – Autor : Stephen Stelting – Olav Maassen – Editorial Pearson
- Core Java 2 – Autor : Cay Horstmann – Gary Cornell – Editorial Pearson
- JAVA como programar. Paul Dietel. Harvey Deitel, Pearson Educacion

b) Complementaria:

- Enterprise Java Beans 3.0 - Autor : Uwe Rozanski – Editorial Alfaomega

Función Docencia

Reuniones de asignatura y área

Atención y orientación a las y los estudiantes

Se establecerán canales y horarios para la atención virtual, dependiendo de las necesidades de los alumnos.