



Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: **METODOLOGÍA DE SISTEMAS I** **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

PROGRAMA ANALITICO

1) OBJETIVOS DE LA ASIGNATURA

El alumno debe ser capaz de:

- Comprender metodologías del análisis, arquitectura y diseño de sistemas.
- Interpretar los modelos resultantes del análisis y diseño de sistemas.
- Elaborar estrategias para la programación de sistemas, aplicando las adecuadas arquitecturas y tecnologías para resolver el problema.

2) CONTENIDOS MÍNIMOS

Sistemas. Ciclo de Vida. Metodologías. Herramientas. Técnicas. Análisis Orientado a Objetos.
Diseño. Técnicas, arquitectura y lenguajes de desarrollo de software avanzada.

3) PROGRAMA ANALÍTICO

UNIDAD TEMÁTICA	CONTENIDOS
	PARTE I: Metodologías y Modelos
1	CICLO DE VIDA Conceptos fundamentales de Sistemas, Sistemas de Información e Ingeniería de Software. Ciclo de Vida de Sistemas. Modelos Evolutivos: Incremental, Modelo Espiral. Estrategias de Prototipado y Reutilización de componentes. Proceso Unificado de Desarrollo de Software. Participantes del proyecto. <u>Bibliografía Principal/ Guías de estudio</u> - PRESSMAN, Roger S. (2010). <i>"Ingeniería del Software: Un Enfoque Práctico"</i>. 7° edición. Editorial Mc Graw Hill. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Proceso Unificado de Desarrollo de Software"</i>. Editorial Addison Wesley. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Lenguaje Unificado de Modelado"</i>. Editorial Addison Wesley. - WEITZENFELD, Alfredo (2005). <i>"Ingeniería de Software Orientada a Objetos con UML, Java e Internet"</i>. Editorial Thomson. - Apuntes de cátedra. Trabajo práctico: Investigación de otras metodologías.
2	CAPTURA DE REQUISITOS Visión general de la captura de requisitos. Clasificación de requisitos. Modelo de Dominio. Modelo de Negocio. <u>Bibliografía Principal / Guías de estudio</u>



Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: METODOLOGÍA DE SISTEMAS I **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

UNIDAD TEMÁTICA	CONTENIDOS
	- PRESSMAN, Roger S. (2010). <i>"Ingeniería del Software: Un Enfoque Práctico"</i>. 7° edición. Editorial Mc Graw Hill. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Lenguaje Unificado de Modelado. Manual de Referencia"</i>. Editorial Addison Wesley. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Proceso Unificado de Desarrollo de Software"</i>. Editorial Addison Wesley. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Lenguaje Unificado de Modelado"</i>. Editorial Addison Wesley. - WEITZENFELD, Alfredo (2005). <i>"Ingeniería de Software Orientada a Objetos con UML, Java e Internet"</i>. Editorial Thomson. Trabajo práctico: Modelo de Negocio y Modelo de Dominio
3	ANÁLISIS y DISEÑO de SISTEMAS El Modelo de Análisis. El Modelo de Diseño. Estereotipos de clases. Diagrama de clases. Diagramas de Interacción: Secuencia y Colaboración. Diagrama de Transición de Estados. Diagrama de Actividades. Diagrama de Paquetes. Patrones. Interfaces. <u>Bibliografía Principal/ Guías de estudio</u> - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Lenguaje Unificado de Modelado. Manual de Referencia"</i>. Editorial Addison Wesley. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Proceso Unificado de Desarrollo de Software"</i>. Editorial Addison Wesley. - RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady (2000). <i>"El Lenguaje Unificado de Modelado"</i>. Editorial Addison Wesley. - WEITZENFELD, Alfredo (2005). <i>"Ingeniería de Software Orientada a Objetos con UML, Java e Internet"</i>. Editorial Thomson. <u>Bibliografía Complementaria</u> - PRESSMAN, Roger S. (2010). <i>"Ingeniería del Software: Un Enfoque Práctico"</i>. 7° edición. Editorial Mc Graw Hill. - LARMAN, Craig (2003). <i>"UML y Patrones"</i>. Editorial Pearson Education. Trabajo práctico: Proceso Unificado y Aplicación de UML a ejercicios integradores.
4	METODOLOGÍAS AGILES Desarrollo ágil: proceso, principios, políticas y factores humanos. Programación Extrema (XP): valores, proceso. Desarrollo adaptativo de software (DAS) Scrum. <u>Bibliografía Principal / Guías de estudio</u> - PRESSMAN, Roger S. (2010). <i>"Ingeniería del Software: Un Enfoque Práctico"</i>. 7° edición. Editorial Mc Graw Hill. - SOMMERVILLE (2011) <i>"Ingeniería de Software"</i> 9° edición. Editorial Pearson Trabajo de campo: formular proyecto para prácticas profesionales



Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: METODOLOGÍA DE SISTEMAS I **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

UNIDAD TEMÁTICA	CONTENIDOS
	PARTE II: Diseño, Arquitectura y Programación
5	LABORATORIO DE ENTORNOS DE DESARROLLO Y HERRAMIENTAS Objetivo : Conocer las herramientas utilitarias que se usan en un Software Factory y las distintas plataformas que se usan en el desarrollo de software. 1- Configuración del entorno de Desarrollo Web LAMP : Instalación del Framework Netbeans, Bootstrap, JQuery – Instalación del entorno XAMPP y su configuración- Aplicación a php. 2- Configuración del entorno de Desarrollo Web MEAN : Instalación de MongoDB, Express, Angular, React, NodeJS y Loopback4. 5- Control de Versiones Instalación de Git y GitHub. -Plugin del lado del cliente para la conexión a un servidor - Protocolos de interconexión a un servidor de versionado. 5- Maven Introducción a la gestión y construcción de proyectos. Instalación y configuración. Gestión del archivo POM. Conceptos de dependencias Bibliografía Principal / Guías de estudio - GAUCHAT, Juan Diego. "El gran libro de HTML5, CSS3 y Javascript". Editorial Marcombo. - SVENNERBERG, Gabriel; (2010). "Beginning Google Maps API 3". Editorial Apress. Bibliografía Complementaria - SIKOS, Leslie,F.(2011) "Web Standards, Mastering Html5,CSS3 and XML"; Editorial Apress. Trabajo práctico: Instalación del software de Base. Trabajo práctico modelo del Docente: Equipo de Futbol – Básico HTML5 Trabajo práctico individual: Sitio Turístico Adaptando el modelo del docente. Trabajo práctico : Elección de tema de trabajo Proyecto Final I - Carpetas de Ejemplo P.P. Trabajo práctico:Google Maps.
6	TECNOLOGIAS DE PROGRAMACION DEL LADO DEL CLIENTE Objetivo : Establecer los conocimientos avanzados de las distintas tecnologías de programación del lado del cliente para el desarrollo de software requeridas en un programador Full Stack.



Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: METODOLOGÍA DE SISTEMAS I **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

UNIDAD TEMÁTICA	CONTENIDOS
	<u>Contenido :</u> 1– Maquetación de proyectos e integración a proyecto de Laboratorio IV : Uso de Herramientas de maquetación de Componentes para la vista . Uso de Balsamiq. Formulación de Historias de Usuarios, Desarrollo del modelo de dominio de proyectos. Diagrama de clases. Modelado de Entidad relación. 2– BootStrap Instalación de plantilla básica – Sistema de grillas – Distintos tipos de componentes CCS – Componentes JS - Ejemplos de maquetación. 3 – JavaScript Funcional Introducción, Qué es DOM, Javascript ESXX (Diferencias con con versiones anteriores. Transpilers. Sentencias de control, funciones. Funciones flecha, funciones puras, inmutabilidad, problema de estado compartido. Promesas y async/await. 4 – Uso Avanzado de Json: Introducción a JSON. Estructura de JSON. Sintaxis. Arrays en JSON – Pruebas en Postman 5 –ReactJS : Introducción a React JS. Usando Create-React-App CLI. Sintaxis JSX y Virtual DOM. Componentes de función y de clase. Manejo de Estado. Ciclo de Vida y eventos. Composición de Componentes. Estilos CSS con React. Introducción a ReactRouter. Introducción a Service Workers con React. Futuro de React: Hooks y otras ideas. 6- Introducción a tecnologías Móviles Tipos de Tecnologías Web, Híbrida y Nativas. Diferencias. Introducción a React Native
7	PERSISTENCIA <u>Objetivo :</u> Conocer los mecanismos de persistencia ofrecidos por el estándar JPA <u>Contenido :</u> 1- Persistencia Api Java. Entidades en JPA. Anotaciones de entidad. Relaciones en JPA con anotaciones. Herencia en JPA. String Data. DAO. Configuración. Consultas y eliminaciones. Cascadeo. Paginación. Lenguaje JPQL. Consultas. Configuración en Spring.
8	



Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: METODOLOGÍA DE SISTEMAS I **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

UNIDAD TEMÁTICA	CONTENIDOS
	TECNOLOGÍAS DE PROGRAMACION DEL LADO DEL SERVIDOR Objetivo : Establecer el conocimiento de la arquitectura del Backend a través Spring Boot Contenido : Spring Boot: 1- Introducción: Arquitectura Web. Arquitectura Web con Spring. Arquitectura del software. Api. Tecnologías del servidor. Características de Spring. Módulos de Spring Framework. Dao(Data Acces Object). Inyección de dependencias. Tipos de Bean, Anotaciones. Ciclo de Vida. Configuración. Maven. Test unitarios. Test unitarios con Inyección de dependencias de Spring. Test unitarios inyección de Mocks. Test de Integración. Spring Framework y Spring Data. Utilización de JPA. Programación Orientada a Aspectos. Introducción. Los problemas transversales. Configuración. Conceptos básicos. Puntos de corte. Consejos. Api Rest – Aplicación de operaciones CRUD. Cliente Rest. Rest Builder. Recursos. Inyección de parámetros. Datos de cuerpo. Tratamiento de errores en la Api Rest con HTTP- Respuestas mediante Excepciones. Seguridad en Api Rest. Proyecto Spring Security. Dependencias. Roles. Usuarios. Servicios de Usuarios.
9	INTEROPERABILIDAD ENTRE APLICACIONES Objetivo : Conocer la tecnología de Rest que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Identificar distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutarlas sobre cualquier plataforma. Intercambiar datos en redes de ordenadores como Internet. Identificar la interoperabilidad se consigue mediante la adopción de estándares abiertos. Contenido : 1 - Concepto de REST Principios de REST - Ejemplo de diseño basado en REST – Verbos Http. Generación de una Api 2 – Noción básica de Arquitectura de Microservicios.



Ministerio de Educación, Ciencia y Tecnología
Universidad Tecnológica Nacional
Facultad Regional Mendoza

Departamento: Ingeniería en Sistemas de Información.
Carrera: Tecnicatura Universitaria en Programación.
Cátedra: **METODOLOGÍA DE SISTEMAS I** **Nivel:** 2 año. **Año lectivo:** 2019
Semestre de cursado: Cuarto **Cantidad de horas semanales:** 12 (doce)
Profesores: Lic. Adriana José - Lic. Alberto Cortez
J.T.P.:
Ayudante de 1ª: **Ayudante de 2ª:**

4) PROGRAMA DE EXAMEN

Ídem programa analítico.

Lugar y fecha: Mendoza, Febrero 2019	COORDINADOR DE CÁTEDRA Apellido y nombre: Lic. Adriana José N° de Legajo: 31192 Lic. Alberto Cortez N° de Legajo: 58255
	Firma: