

MODALIDAD ACADÉMICA

Asignatura	Arquitectura de Software	
Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN	
Ciclo Lectivo	2018	
Vigencia del programa	Desde el ciclo lectivo 2018	
Plan	2008	
Nivel	<input type="checkbox"/> 1er. Nivel <input type="checkbox"/> 2do. Nivel <input type="checkbox"/> 3er. Nivel <input type="checkbox"/> 4to. Nivel <input checked="" type="checkbox"/> 5to. Nivel	
Coordinador de la Cátedra	Ing. Pablo Sebastián Frias	
Área de Conocimiento	<input checked="" type="checkbox"/> Programación <input type="checkbox"/> Computación <input checked="" type="checkbox"/> Sistemas de Información <input type="checkbox"/> Gestión Ingenieril <input type="checkbox"/> Modelos <input checked="" type="checkbox"/> Complementaria	
Carga horaria semanal	6 horas	
Anual/ cuatrimestral	Cuatrimestral	
Contenidos Mínimos, según Diseño Curricular-Ordenanza 1150 (sólo para asignaturas curriculares)		
Correlativas para cursar (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> • Diseño de Sistemas • Redes de Información 	<ul style="list-style-type: none"> • Análisis de Sistemas • Gestión de Datos
Correlativas para rendir (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> • Ingeniería de Software • Redes de Información 	<ul style="list-style-type: none"> • Diseño de Sistemas • Gestión de Datos
Objetivos de la Asignatura	<p>Objetivos Generales:</p> <ul style="list-style-type: none"> • Comprender la importancia de la arquitectura de software y su relación con el diseño a lo largo del ciclo de vida de desarrollo. • Conocer el rol y las responsabilidades del arquitecto de software y su relación con los diversos stakeholders del proyecto. • Identificar las mejores opciones de definición de arquitectura de acuerdo a los requerimientos del sistema de información a desarrollar. • Conocer las diferentes herramientas conceptuales para ayudar a definir una arquitectura de software. • Adquirir los conocimientos necesarios para liderar una arquitectura de software. <p>Objetivos Particulares:</p> <ul style="list-style-type: none"> • Complementar las habilidades adquiridas en materias correlativas anteriores, 	

	<p>tanto de programación como en sistemas de información.</p> <ul style="list-style-type: none"> • Fomentar el espíritu crítico del ingeniero para tomar decisiones de impacto a largo plazo. • Comprender la importancia de definir una arquitectura que sustente la fiabilidad de un sistema de software a lo largo de su ciclo de vida. • Determinar la mejor arquitectura de acuerdo a los requerimientos del sistema y las restricciones del proyecto. • Comprender el estado actual de los patrones arquitectónicos más usados para diferentes tipos de sistemas de software. • Conocer las herramientas para la implementación y verificación del diseño arquitectónico especificado. • Evaluar los aspectos de seguridad del sistema de información al momento de diseñar la arquitectura.
--	--

Programa Analítico

UNIDAD NRO 1: INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE

Objetivos Específicos:

- Conocer la relación de la arquitectura con el diseño.
- Comprender las responsabilidades del rol del arquitecto de un sistema y los principios generales de arquitectura.
- Conocer el impacto de las decisiones de arquitectura en relación con los requerimientos funcionales del sistema y la capacidad del equipo de desarrollo de entregar el producto de acuerdo a lo planificado.
- Comprender el impacto de la arquitectura en la flexibilidad del sistema ante los cambios de requerimientos del software a lo largo de su ciclo de vida.

Contenidos:

- Concepto de Arquitectura de Software y su Relación con el Diseño.
- Responsabilidades del Arquitecto.
- Principios de Arquitectura.
- Requerimientos no funcionales y restricciones.

Bibliografía:

- BURRIS, Eddie. Programming in the Large with Design Patterns. Pretty Print Press, 2012
- TAYLOR, R. N. y otros. Software Architecture: Foundations, Theory, and Practice. Wiley, 2009.

Evaluación:

Este temario se incluye en la evaluación parcial, el trabajo integrador y el coloquio del examen final.

UNIDAD NRO 2: ESTILOS ARQUITECTÓNICOS

Objetivos Específicos:

- Conocer los principales patrones de diseño arquitectónico estructurales.
- Profundizar en el conocimiento de los Estilos Arquitectónicos más utilizados.
- Comprender la diversidad de arquitecturas estructurales, ventajas y desventajas, dependiendo del dominio del problema.
- Tener herramientas y referencias de arquitecturas para liderar un software en un nuevo dominio de problema.

Contenidos:

- Definiciones generales, Componentes, Conectores y Configuraciones.
- Patrones arquitectónicos: SOA, Capas, Monolítica
- Estilos arquitectónicos en Capas, Dataflow, Intérprete, P2P.
- Arquitecturas Cliente Servidor: Clientes Pesados, Clientes Livianos
- Arquitectura de Proyectos de Fuente Abierta.
- Arquitecturas de Microservicios.
- Arquitecturas Stream Processing.

Bibliografía:

- BURRIS, Eddie. Programming in the Large with Design Patterns. Pretty Print Press, 2012.
- TAYLOR, R. N. y otros. Software Architecture: Foundations, Theory, and Practice..Wiley, 2009.
- BROWN, Amy y otros, The Architecture of Open Source Applications Wilson, Greg 2011.

Evaluación:

Este temario se incluye en la evaluación parcial, el trabajo integrador y el coloquio del examen final.

UNIDAD NRO 3: SEGURIDAD EN MODELOS Y PATRONES ARQUITECTÓNICOS

Objetivos Específicos:

- Comprender la importancia de considerar la seguridad y privacidad de los datos como parte integral de la arquitectura.
- Distinguir entre los problemas de seguridad y privacidad de datos.
- Integrar el análisis de seguridad en todas las etapas del ciclo de desarrollo.
- Resolver los problemas más comunes de seguridad y privacidad en aplicaciones de arquitectura cliente-servidor.

Contenidos:

- La importancia de la seguridad en el diseño arquitectónico.
- El Ciclo de Desarrollo Software Seguro (SDL por sus siglas en inglés).
- Creación de documento de Threat Model.
- OWASP - Open Web Application Security Project - Prácticas de Código seguro..
- Conceptos generales de Pen Testing. Herramientas.
- Introducción a Kali Linux.

Bibliografía:

- HOWARD, M., LIPNER, S. The Security Development Lifecycle, SDL: A Process for Developing Demonstrably More Secure Software Microsoft Press 2006.
- TAYLOR, R. N. y otros. Software Architecture: Foundations, Theory, and Practice..Wiley, 2009.

Evaluación:

Este temario se incluye en la evaluación parcial, el trabajo integrador y el coloquio del examen final.

UNIDAD NRO 4: PATRONES, SERVICIOS DE INFRAESTRUCTURA

Objetivos Específicos:

- Conocer los patrones de alto nivel más comunes para capas de persistencia.
- Profundizar en el conocimiento de los frameworks más comunes que facilitan la implementación de patrones de capa de persistencia.
- Comprender importancia de pensar la arquitectura teniendo en cuenta la facilidad de testeo y aseguración de la calidad del código.
- Comprender los conceptos claves arquitectónicos para la implementación de lógica de negocio.
- Conocer las herramientas más usadas para la implementación.
- Conocer los beneficios de contar con una arquitectura de servicios eficiente y que sustente la evolución del sistema de software.
- Contar con las herramientas necesarias implementar servicios de infraestructura robustos, seguros y adaptables al cambio.

Contenidos:

- Diseños de capas de persistencia aptas para testeo unitario y mocking.
- Patron Active Record, DAO y Repository.
- Framework Hibernate. Entity Framework.
- Patrones para la asignación de Responsabilidades (GRASP): Experto de información, creador, etc.
- Patrones IoC y DI
- Framework de IoC castle Windsor y Spring.
- Patrones de UI, MVC - Model View Controller, MVP - Model View Presenter, MVVM - Model View - View Model
- Introducción al paradigma de orientación a aspectos (AOP) y Frameworks.
- Implementación Transacciones, flujo de la transacción entre objetos.
- Infraestructura con Hypervisors/Containers. Docker

Bibliografía:

- GUERNSEY, Max. Test Driven Database Developmen, Unlocking Agility - Addison Wesley 2013.
- BAUER, C, y otros. Java Persistence with Hibernate Mannig Publications 2013.
- DESSI, Massimiliano. Spring 2.5 Aspect Oriented Programming – Packt Publishing 2009.
- GROVES, Matthew. AOP in .NET – Practical Aspect Oriented Programming Manning Publication 2011.
- LADDAD, Ramnivas. AspectJ in Action – Manning Publication 2009
- BERNSTEIN, P, y otros. Principles of Transaction Processing, 2nd Edition Morgan Kaufmann 2009.

Evaluación:

Este temario se incluye en la evaluación parcial, el trabajo integrador y el coloquio del examen final.

UNIDAD NRO 5: ARQUITECTUAS PARA EL CLOUD, SISTEMAS DISTRIBUIDOS Y SISTEMAS IoT

Objetivos Específicos:

- Conocer los beneficios de contar con una arquitectura de servicios eficiente y que sustente la evolución del sistema de software.
- Contar con las herramientas necesarias implementar servicios de infraestructura robustos, seguros y adaptables al cambio.

Contenidos:

- Introducción al Cloud Computing. Beneficios y ventajas.
- Paradigma MapReduce.
- Protocolo Gossip.
- Sistemas P2P.
- Concepto de consistencia eventual, NoSql, Teorema CAP.

Bibliografía:

- Anthony T. Velte, Toby J. Velte, Cloud Computing a practical approach, 2009
- Bob Lozano, Executive's guide to Cloud Computing, 2010
- Dr Mark I Williams, A quick start guide to Cloud Computing, moving your business into the cloud 2010

Evaluación:

Este temario se incluye en la evaluación parcial, el trabajo integrador y el coloquio del examen final.

Metodología de enseñanza y aprendizaje	<p>Se desarrollarán clases teóricas con el propósito de conceptualizar los contenidos de la asignatura y formular una base de conocimiento a partir de la cual se emprenderán ejercicios prácticos correspondientes. Al mismo tiempo, se desarrollarán talleres sobre temáticas específicas, donde se incluirán experiencias concretas de sistemas reales, con disertantes invitados.</p> <p>Las clases prácticas implicarán una alta participación del alumno en la resolución de problemas de arquitectura, al mismo tiempo en el que se participará en un proyecto de aplicación, orientado a ejercitar en un ambiente real los temas teóricos y prácticos abordados en la asignatura.</p>																																	
Sistema de evaluación	<p>El alumno será evaluado mediante un parcial, teniendo la posibilidad de recuperar en caso de no alcanzar las expectativas mínimas de aprobación.</p> <p>Además se deberá presentar un trabajo práctico integrador que se desarrollará a lo largo de toda la materia.</p>																																	
Regularidad: condiciones	<p>Las condiciones mínimas para acceder a la regularización de la materia serán:</p> <ul style="list-style-type: none"> • Aprobar 3 entregas del trabajo práctico integral con nota no menor a 4. • Aprobar el examen teórico/práctico con nota no menor a 4, con posibilidad de recuperar en caso de aplazo. <p>El estudiante en condición de regular, deberá rendir un coloquio final de la materia. Durante el periodo de un ciclo lectivo, podrá rendir sin control de correlativas aprobadas.</p> <p>La siguiente tabla muestra los porcentajes necesarios para las calificaciones.</p> <p>Escala de notas de regularidad(*)</p> <table border="1" data-bbox="678 1514 1248 1892"> <thead> <tr> <th>NOTAS</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>2</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>3</td> <td></td> <td>No Aprobado</td> </tr> <tr> <td>4</td> <td>55% a 57%</td> <td>Aprobado</td> </tr> <tr> <td>5</td> <td>58% a 59%</td> <td>Aprobado</td> </tr> <tr> <td>6</td> <td>60% a 68%</td> <td>Aprobado</td> </tr> <tr> <td>7</td> <td>69% a 77%</td> <td>Aprobado</td> </tr> <tr> <td>8</td> <td>78% a 86%</td> <td>Aprobado</td> </tr> <tr> <td>9</td> <td>87% a 95%</td> <td>Aprobado</td> </tr> <tr> <td>10</td> <td>96% a 100%</td> <td>Aprobado</td> </tr> </tbody> </table>	NOTAS	PORCENTAJE	CALIFICACIÓN	1		No Aprobado	2		No Aprobado	3		No Aprobado	4	55% a 57%	Aprobado	5	58% a 59%	Aprobado	6	60% a 68%	Aprobado	7	69% a 77%	Aprobado	8	78% a 86%	Aprobado	9	87% a 95%	Aprobado	10	96% a 100%	Aprobado
NOTAS	PORCENTAJE	CALIFICACIÓN																																
1		No Aprobado																																
2		No Aprobado																																
3		No Aprobado																																
4	55% a 57%	Aprobado																																
5	58% a 59%	Aprobado																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Aprobado																																
8	78% a 86%	Aprobado																																
9	87% a 95%	Aprobado																																
10	96% a 100%	Aprobado																																

	(*) Escala acordada en reunión de Docentes Coordinadores																																	
Promoción: condiciones	No existe esta condición para esta Cátedra																																	
Aprobación Directa: condiciones.	<p>Todo aquel estudiante que apruebe el parcial y el trabajo integrador al momento de regularizar, con una nota promedio de 7 (siete) o más, no deberá rendir coloquio final oral. La calificación será la nota registrada como Nota Final en Autogestión. El estudiante en esta condición, puede registrar su nota en examen en el plazo de un ciclo lectivo, sin control de correlativas aprobadas, y después de ello se le exigirán correlativas aprobadas. Es requisito obligatorio la presentación del trabajo práctico integrador con las siguientes condiciones:</p> <ul style="list-style-type: none"> • Realizar 3 entregas de trabajos prácticos en formato digital editable para realizar comentarios correspondientes. Cada trabajo se subirá por medio de la plataforma Moodle. No se aceptarán PDFs. • Código fuente accesible en Github. • Incluir documento de Threat Model de la aplicación. • Presentación de avances y demo en clase. 																																	
Modalidad de examen final	<p>El alumno que haya alcanzado sólo la condición de regular, deberá rendir un coloquio final oral. El objetivo del mismo es repasar el temario visto en clase y conocer los criterios del alumno a la hora de resolver problemas de software.</p> <p>Escala de Notas para Examen Final (*)</p> <table border="1"> <thead> <tr> <th>NOTA</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td>Insuficiente</td></tr> <tr><td>2</td><td></td><td>Insuficiente</td></tr> <tr><td>3</td><td></td><td>Insuficiente</td></tr> <tr><td>4</td><td></td><td>Insuficiente</td></tr> <tr><td>5</td><td></td><td>Insuficiente</td></tr> <tr><td>6</td><td>60% a 68%</td><td>Aprobado</td></tr> <tr><td>7</td><td>69% a 77%</td><td>Bueno</td></tr> <tr><td>8</td><td>78% a 86%</td><td>Muy Bueno</td></tr> <tr><td>9</td><td>87% a 95%</td><td>Distinguido</td></tr> <tr><td>10</td><td>96% a 100%</td><td>Sobresaliente</td></tr> </tbody> </table> <p>(*) Escala acordada en reunión de Docentes Coordinadores</p>	NOTA	PORCENTAJE	CALIFICACIÓN	1		Insuficiente	2		Insuficiente	3		Insuficiente	4		Insuficiente	5		Insuficiente	6	60% a 68%	Aprobado	7	69% a 77%	Bueno	8	78% a 86%	Muy Bueno	9	87% a 95%	Distinguido	10	96% a 100%	Sobresaliente
NOTA	PORCENTAJE	CALIFICACIÓN																																
1		Insuficiente																																
2		Insuficiente																																
3		Insuficiente																																
4		Insuficiente																																
5		Insuficiente																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Bueno																																
8	78% a 86%	Muy Bueno																																
9	87% a 95%	Distinguido																																
10	96% a 100%	Sobresaliente																																
Actividades en laboratorio	Las actividades prácticas se desarrollarán en laboratorio, con la utilización de herramientas y lenguajes de programación acorde, dependiendo de las herramientas elegidas.																																	
Horas/año totales de la asignatura (hs. cátedra)	90 horas																																	
Cantidad de horas prácticas totales (hs. cátedra)	50 horas																																	
Cantidad de horas teóricas totales (hs. cátedra)	40 horas																																	
Tipo de formación práctica	<input type="checkbox"/> Formación experimental <input checked="" type="checkbox"/> Resolución de problemas de ingeniería <input checked="" type="checkbox"/> Actividades de proyecto y diseño																																	

	<input type="checkbox"/> Prácticas supervisadas en los sectores productivos y /o de servicios														
Cantidad de horas cátedras afectadas a la formación práctica indicada en el punto anterior	<ul style="list-style-type: none"> • 35hs a resolución de problemas de ingeniería • 15hs a Actividades de Proyecto y diseño 														
Descripción de los prácticos	<p>Sumados a los prácticos áulicos, el alumno deberá realizar un trabajo práctico integrador que consta de 3 etapas, donde se demuestren el uso y aprendizaje de los contenidos de las distintas unidades de la material.</p> <p>Las fechas de presentación de los avances por etapas serán estipuladas en el cronograma de actividades del ciclo lectivo.</p> <p>Es condición necesaria para regularizar la materia que el alumno apruebe el trabajo práctico integrador.</p> <p>Al final del cursado, se obtendrá una única nota de trabajo práctico integrador, que resultará del promedio de las distintas entregas estipuladas.</p> <p>En el enunciado de cada etapa del trabajo práctico integrador se describirán los criterios de evaluación específicos que se tendrán en cuenta para la corrección del mismo.</p> <p>Cada etapa del trabajo práctico integrador entregado será calificado con una nota entera comprendida entre 1 y 10 según la escala de notas para regularidad.</p> <p>El Trabajo Práctico Integrador deberá ser desarrollado en grupos de hasta 3 (tres) alumnos.</p>														
Criterios generales	<ul style="list-style-type: none"> • Alcance de las soluciones planteadas. • Factibilidad de implementación de las soluciones planteadas, de acuerdo a restricciones de alcance, tiempo y costo de desarrollo. • Mantenibilidad del sistema generado. • Uso de patrones de diseño / desarrollo en las soluciones planteadas. • Uso de herramientas existentes para el tipo de arquitectura planteada. • Uso de herramienta de versionado de código, tests unitarios y revisiones de código. • Coherencia de la solución planteada con respecto a los requerimientos establecidos. 														
Cronograma de actividades de la asignatura	<table border="1"> <thead> <tr> <th>Semana</th> <th>Fecha</th> <th>Teórico</th> <th>Práctico</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>19/03/2018</td> <td>Unidad 1: Introducción a la Arquitectura de Software y rol del arquitecto de software. Conceptos generales de Componentes, Conectores y Configuraciones.</td> <td>Presentación y elección de Trabajos Prácticos. Condiciones de aprobación del curso. Presentación de Trabajo Práctico de ejemplo.</td> </tr> <tr> <td>2</td> <td>26/03/2018</td> <td>Unidad 2: Estilos arquitectónicos generales.</td> <td>Introducción y práctica de sistemas de versionado de código. Git y revisiones de</td> </tr> </tbody> </table>			Semana	Fecha	Teórico	Práctico	1	19/03/2018	Unidad 1: Introducción a la Arquitectura de Software y rol del arquitecto de software. Conceptos generales de Componentes, Conectores y Configuraciones.	Presentación y elección de Trabajos Prácticos. Condiciones de aprobación del curso. Presentación de Trabajo Práctico de ejemplo.	2	26/03/2018	Unidad 2: Estilos arquitectónicos generales.	Introducción y práctica de sistemas de versionado de código. Git y revisiones de
Semana	Fecha	Teórico	Práctico												
1	19/03/2018	Unidad 1: Introducción a la Arquitectura de Software y rol del arquitecto de software. Conceptos generales de Componentes, Conectores y Configuraciones.	Presentación y elección de Trabajos Prácticos. Condiciones de aprobación del curso. Presentación de Trabajo Práctico de ejemplo.												
2	26/03/2018	Unidad 2: Estilos arquitectónicos generales.	Introducción y práctica de sistemas de versionado de código. Git y revisiones de												

			código.
3	02/04/2018	Unidad 2: SOA - Micro servicios – Stream processing. Ejemplo de Sistema de microservicio.	Presentación de avances para la primer entrega del TP. Práctica de arquitectura de sistema REST.
4	09/04/2018	Unidad 3: Seguridad Informática. Introducción al SDL Introducción al documento de Threat Model.	Primer entrega de TP. Testing Unitario, herramientas de mocking. Ejemplo práctico. Métricas de cobertura de testing. Introducción general a herramientas de Building tools: Ant/Gradle/Maven/Nuget/MSBuild Jenkins/TeamCity
5	16/04/2018	Unidad 3: Patrones de Seguridad Introducción al Pen testing. Herramientas	Presentación de avances de TPs. Práctica con herramientas de Pen Testing.
6	20/04/2018	Unidad 3: Introducción a Kali Linux.	Presentación de avances de TPs. Práctica general sobre Sistema Kali Linux.
7	27/04/2018	Unidad 4: Práctica de codificación y resolución de problemas con patrones. Virtualización y Containers. Práctica con Docker.	Presentación de avances de TPs. Herramientas de persistencia. Práctica con Hibernate/NHibernate.
8	04/05/2018	Unidad 4: Práctica de codificación de patrones de UI. Práctica de AOP.	Segunda entrega de TP. Práctica de frameworks de MVC – MVP
9	11/05/2018	Unidad 4: IoC: práctica con Spring Framework.	Presentación de avances de TPs. IoC: Windsor Castle.
10	18/05/2018	Unidad 4: Practica sobre sistemas Streams. Sistemas de mensajería, MQTT, RabbitMQ.	Presentación de avances de TPs.

	11	25/05/2018	Repaso general de herramientas, estableciendo entornos de configuración, prueba y producción.	Presentación de avances de TPs. Práctica integral de herramientas.
	12	01/06/2018	Unidad 5: Sistemas para el Cloud.	Entrega 3 de TP. Práctica de Cloud computing y sistemas distribuidos. Load balancers, key-value stores.
	13	08/06/2018	Unidad 5: Sistemas IoT.	Práctica Arduino/Raspberri Pi
	14	15/06/2018	Clase de consulta y repaso	Parcial
	15	22/06/2018	Firma de libretas	
Propuesta para la atención de consultas y mail de contacto.	<p>Las actividades de consulta previa a cada instancia de evaluación ya sea de parcial o de entrega de prácticos se realizarán en espacio destinado a tal fin dentro del horario de cursado de la asignatura.</p> <p>Uso de email y del foro del aula virtual, todo el año. No existe un cronograma que limite o restrinja las consultas de alumnos.</p> <p>Los correos electrónicos de consultas serán provistos el primer día de clases.</p>			
Plan de integración con otras asignaturas	<p>El contenido de la materia presupone que el alumno ha adquirido de manera adecuada los conocimientos de las siguientes materias:</p> <ul style="list-style-type: none"> • Algoritmos y Estructuras de Datos • Paradigmas de Programación • Análisis de Sistemas • Diseño de Sistemas 			
Bibliografía Obligatoria	<ul style="list-style-type: none"> • BURRIS, Eddie. Programming in the Large with Design Patterns. Pretty Print Press, 2012. • TAYLOR, R. N. y otros. Software Architecture: Foundations, Theory, and Practic. Wiley, 2009. • BROWN, Amy y otros, The Architecture of Open Source Applications Wilson, Greg 2011. • HOWARD, M., LIPNER, S. The Security Development Lifecycle, SDL: A Process for Developing Demonstrably More Secure Software Microsoft Press 2006. • DHANJI, R, P. Dependency Injection Manning Publications 2009. • BERNSTEIN, P, y otros. Principles of Transaction Processing, 2nd Edition Morgan Kaufmann 2009. 			
Bibliografía Complementaria	<ul style="list-style-type: none"> • GUERNSEY, Max. <i>Test Driven Database Development, Unlocking Agility</i> - Addison Wesley 2013. • BAUER, C, y otros. <i>Java Persistence with Hibernate</i> Mannig Publications 2013. • DECK, Paul. <i>Spirng MVC (A Tutorial Series)</i> – BrainySoftware 2014. • GAROFOLO, Raffaele <i>Building Enterprise Applications with Windows Presentation Foundation and Model View – View Model</i> – Microsoft Press 2011. 			

	<ul style="list-style-type: none"> • DESSI, Massimiliano. <i>Spirng 2.5 Aspect Oriented Programming</i> – Packt Publishing 2009. • GROVES, Matthew. <i>AOP in .NET – Practical Aspect Oriented Programming</i> Manning Publication 2011. • LADDAD, Ramnivas. <i>AspectJ in Action</i> – Manning Publication 2009t 																																				
<p>Distribución de docentes</p>	<table border="1"> <thead> <tr> <th><i>Curso</i></th> <th><i>Turno</i></th> <th><i>Día y Horas</i></th> <th><i>Profesor</i></th> <th><i>JTP</i></th> <th><i>Ayudante</i></th> </tr> </thead> <tbody> <tr> <td>5K4</td> <td>Noche</td> <td>Lunes 20:40 hs Viernes 18:30 hs</td> <td>Pablo Frias</td> <td>Germán Romani</td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>	5K4	Noche	Lunes 20:40 hs Viernes 18:30 hs	Pablo Frias	Germán Romani																									
<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>																																
5K4	Noche	Lunes 20:40 hs Viernes 18:30 hs	Pablo Frias	Germán Romani																																	

Firma:

Aclaración: