PATRONES DE ARQUITECTURAS DE SOFTWARE DE APLICACIONES ENTERPRISE

Año 2018

Carrera:
Licenciatura en Informática Plan 2015
Licenciatura en Sistemas Plan 2015
Licenciatura en Sistemas Plan 2007 y Plan 2012.
Año:
Régimen de Cursada: *Semestral (primer semestre)*

Carácter: *Optativa*
Correlativas: *Proyecto de Software - Orientación a Objetos 2 - Ingeniería de Software 2*
Profesor: *Mariano Bibbo*
Hs semanales: *6 hs*

FUNDAMENTACIÓN

La Arquitectura de Software es el diseño de alto nivel de la estructura de un sistema. El diseño de aplicaciones empresariales, que dan soporte a los procesos de negocio de una organización (Aplicaciones Enterprise), suele ser una tarea muy compleja. Los diseñadores de este tipo de aplicaciones tienen que prestar especial atención a varios aspectos que intervienen en el producto a desarrollar, dado que no solo deberán atender los aspectos funcionales de las aplicaciones sino que tendrán que trabajar en cuestiones de eficiencia, escalabilidad, performance, seguridad, entre otras, que constituyen el diseño de las aplicaciones Enterprise. Por otro lado, los arquitectos de software suelen enfrentar problemas que ocurren una y otra vez en distintos proyectos de software; como dividir a la aplicación en capas, manejar la concurrencia de acceso a recursos o vincular la aplicación con la estructura de datos. En estos casos se recurre a soluciones que resultaron exitosas en otros casos similares. Al implementar esta modalidad el arquitecto o diseñador aplicó un Patrón de diseño.

OBJETIVOS GENERALES

A partir del dictado de esta materia se pretende que los alumnos:

- logren comprender los desafíos que aparecen en construcción de aplicaciones enterprise,
- que puedan conceptualizar los problemas que aparecen en este tipo de aplicaciones
- que obtengan estrategias para aplicar las soluciones más apropiadas a cada uno de esos problemas
- finalmente, que los alumnos estén en condiciones de evaluar alternativas de diseño y logren medir los costos y beneficios de cada alternativa.

CONTENIDOS MINIMOS

- Arquitectura
- Aplicaciones enterprise
- Patrones
- Layering (división en capas)
- Concurrencia
- Mapeo a base de datos relacional

PROGRAMA ANALÍTICO

Unidad 1: Introducción a las Aplicaciones Enterprise.

- Introducción. Definición de arquitectura, aplicaciones enterprise, tipos de aplicaciones enterprise, pensando en performance, patrones.
- Layering. La evolución de las capas en las aplicaciones enterprise, las tres capas principales, eligiendo donde correr las capas.
- Organizando la lógica de dominio. Capa de servicio.

Unidad 2: Mapeando a base de datos relacionales

- Patrones arquitectónicos. El problema del comportamiento. Leyendo datos.
- Patrones de mapeo de estructuras. Mapeando realaciones. Herencia
- Construyendo el mapeo. Mapeo doble.
- Utilizando metadata.
- Conecciones a la base de datos.

Unidad 3. Capa de presentación.

- Patrones de vista.
- Patron: Input Controller Pattern.

Unidad 4. Concurrencia.

- Problemas de concurrencia.
- Contextos de ejecución.
- Aislamiento e inmutabilidad.
- Control de concurrencia optimista y pesimista. Lecturas incosistentes.
- Transacciones. Recursos transaccionales. Transacciones de sistema y de negocio.
- Patrones para control de concurrencia offline.

Unidad 5. Estado de Sesión.

- El valor de statelessness
- Session state. Formas de almacenar el session state.

Unidad 6. Inversión de Control

- Inversión de control. Un ejemplo trivial.
- Formas de inyección de dependencias.
- Utiliizando el Service Locator.
- Decidiendo que opción utilizar.

Unidad 7. Plataforma de Desarrollo

- Definición de Plataforma de Desarrollo
- Que beneficios nos aporta una plataforma de desarrollo

BIBLIOGRAFÍA OBLIGATORIA

- Patterns of Enterprise Application Architecture. Martin Fowler. Addison-Wesley Professional, 2002, ISBN 0-321-12742-0.
- Software System Architecture: Working With Stakeholders Using Viewpoints and Perspectives. Nick Rozanski, Eoin Woods. Addison Wesley, 2005, ISBN 0-321-11229-6
- Software Architecture in Practice, Second Edition. Len Bass, Paul Clements, Rick Kazman. Addison Wesley, 2003, ISBN 0-321-15495-9.
- Documenting Software Architectures: Views and Beyond. Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford. Addison Wesley, 2002, ISBN 0-201-70482-X.
- *Design Patterns. Elements of Reusable Object-Oriented Software* - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides - Addison Wesley (GoF- Gang of Four)
- *A System of Patterns* - Buschmann, Meunier, Rohnert, Sommerlad, Stal - Wiley
- *UML y Patrones. Introducción al análisis y diseño orientado a objetos* - Larman - Prentice Hall
- Pattern-Oriented Software Architecture: A System of Patterns. Buschmann, F.,Meunier, R.,Rohnert, H.,Sommerlad, P.,Stal, M. 1996. Chichester: John Wiley and Sons, 1996, ISBN 0-47195889-7

METODOLOGÍA DE ENSEÑANZA

Se organizan clases teóricas en formato digital en las cuáles el docente, con participación de los alumnos, presentará los temas del programa. Además se establecerán distintos ejercicios a desarrollar que guiarán a los alumnos en la concreción de los objetivos anteriormente planteados para que los resuelvan durante el transcurso de la materia. Promediando la cursada se establecerá una modalidad de seminario donde los alumnos presentarán un conjunto de patrones y se discutirán la

posible utilidad de los mismos en distintas situaciones que se iran planteando. Los alumnos participarán activamente interactuando con los compañeros y el cuerpo docente. Se dispondrá de un entorno virtual para descargar los materiales de clase y hacer un seguimientos de las consultas de los alumnos.

EVALUACIÓN

La evaluación de la cursada comprende la aprobación de una serie de ejercicios obligatorios (2-3) o pruebas de concepto donde los alumnos implementarán algunos patrones vistos en las clases teóricas.

Para la evaluación final es requisito que los alumnos que aprobaron la cursada realicen un examen final escrito o la implementación de un ejercicio integrador visto durante el año. Como resultado de la evaluación final se les pasará una nota global que comprende la evaluación final de la materia.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas	Fechas
1	Exposición teórica de los siguientes temas: • Introducción • Aplicaciones enterprise • Arquitectura de software para aplicaciones Enterprise • Plataforma de desarrollo para aplicaciones Enterprise • Introducción a los temas a las que juntas deben dar respuesta	Ninguna	7/3/2018
2	Exposición teórica de los siguientes temas: • Definición de Layering • Capas más importantes en una arquitectura tradicional Enterprise		
3	Exposición teórica de los siguientes temas: • Persistencia de datos • Problemas y Patrones de solución para la persistencia de datos • Persistencia de mapeo objeto-relacional • Bases de datos orientada a objetos		
4	Exposición teórica de los siguientes temas: • Persistencia ◦ Hibernate Presentación del TP1: Persistencia de datos		
5	Revisión/Consulta del avance del ejercicio planteado.		
6	Entrega del TP1. Exposición teórica de los siguientes temas: • Persistencia ◦ Bases de datos en memoria ◦ NO-SQL	Evaluación del TP1 entregado	18/4/2018

7	Exposición teórica de los siguientes temas: • Problemas y Patrones de solución para la capa de la lógica de dominio Presentación de TP2: Implementación en Java de alguno de los patrones discutidos en clase, relacionados con la capa de lógica de dominio		
8	Consulta/Revisión del avance del TP2.		
9	Seminario: • Presentación de los grupos de alumnos sobre el TP2.		
10	Exposición teórica de los siguientes temas: • Capa de presentación ◦ Aplicaciones web ◦ Aplicaciones de escritorio ◦ Problemas y Patrones de solución para la capa de presentación.		
11	Exposición teórica de los siguientes temas: • Capa de presentación ◦ Tecnologías y frameworks ◦ Desarrollo orientado a componentes. Presentación TP3. Desarrollo del front-end de una aplicación para evaluar los conceptos tratados sobre la capa de presentación		
12	Revisión/Consulta del avance del ejercicio planteado TP3.		
13	Seminario: • Presentación de los grupos de alumnos sobre el TP3.	Evaluación del TP3 entregado	06/06/2018
14	Exposición teórica de los siguientes temas: • Temas avanzados ◦ Motor de reglas ◦ BPM		
15	Presentación / Discusión del trabajo final de la materia que involucra todos los conceptos tratados durante las clases teóricas, la bibliografía sugerida y en los trabajos prácticos realizados		
16	Revisión/Consulta del avance del ejercicio planteado como trabajo final.		
17	Revisión/Consulta del avance del ejercicio planteado como trabajo final.		
18	Revisión/Consulta del avance del ejercicio planteado como trabajo final.		
		Evaluación de los trabajos presentados	06/07/2018

BIBLIOGRAFÍA COMPLEMENTARIA

- Software Architecture, Perspectives on an Emerging Discipline. M. Shaw and D. Garlan. Prentice-Hall. 1996.

- Evaluating Software Architectures: Methods and Case Studies. Len Bass, Paul Clements, Rick Kazman. Addison Wesley
- The Art of Software Architecture: Design Methods and Techniques. Stephen T. Albin. John Wiley & Sons, 2002, ISBN 0-471-22886-9.

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

Mail del Profesor: lmbibbo@lifia.info.unlp.edu.ar

Firmas del/los profesores responsables: