

## PROGRAMA de Arquitectura de Software I

**Carrera:** Licenciatura en Informática

**Asignatura:** Arquitectura de Software 1

**Núcleo al que pertenece:** Avanzado

**Profesor:** Leonardo Volinier

**Asignaturas Correlativas:** Programación Concurrente, Práctica de Desarrollo de Software, Gestión de Proyectos de Software, Seguridad de la Información.

### Objetivos:

Que quienes cursan la materia,

- Aprendan a planificar y gestionar proyectos software.
- Conozcan cómo realizar estimaciones de tamaño, esfuerzo y costos en proyectos de software.
- Sepan identificar y gestionar riesgos en proyectos software.
- Sean capaz de gestionar un proyecto software mediante diferentes herramientas de soporte.

### Contenidos mínimos:

- Arquitectura de software y arquitectura de sistemas. Definición, objetivos y clasificación
- Actividades en un proyecto de software relacionadas con la arquitectura. Proceso de definición y evolución de una arquitectura en diferentes metodologías de desarrollo. Arquitectura en metodologías ágiles. Construcción de prototipos como herramienta de verificación y documentación de una arquitectura.
- Insumos para la definición de arquitectura: requerimientos funcionales y no funcionales, restricciones, influencias, entorno social y técnico, estándares, herramientas disponibles. Objetivos de una arquitectura: no intrusividad, no duplicación, separación de responsabilidades, garantía de atributos de calidad, robustez.
- Estilos arquitectónicos. Arquitecturas en capas, arquitecturas orientadas a servicios, arquitecturas orientadas a objetos, arquitecturas orientadas a procesos (BPM). Patrones arquitecturales. Limitaciones del paradigma de objetos y las arquitecturas tradicionales. Programación declarativa.
- Arquitectura de dominio. Modelado de la lógica de dominio. Patrones. Diseño guiado por el dominio (DDD). Motores de reglas. Workflows.
- Integración de los componentes de una arquitectura: lógica de dominio, interfaz de usuario, persistencia, seguridad, etc. Integración basada en aspectos (AOP). Acoplamiento y comunicación entre los componentes de la arquitectura. Transporte de la información. Manejo de transacciones. Tácticas para garantizar robustez y modificabilidad.
- Patrones arquitecturales para la interfaz de usuario. Integración con el dominio.

- Arquitectura de persistencia. Impacto de la persistencia sobre un diseño orientado a objetos. Bases de datos multidimensionales, datawarehouse, data mining.
- Integración de aplicaciones. Clasificación de los mecanismos de integración: base de datos, dominio, servicios, interfaz de usuario. Estrategias de integración apropiadas para ambientes compatibles e incompatibles entre sí. Integración sincrónica y asincrónica. Colas de mensajes. Callbacks. Arquitecturas orientadas a servicios. Web services. Integración con aplicaciones legacy. Patrones para la integración: punto a punto, middleware, Enterprise Service Bus. Definición de interfaces y conectores. Definición de procesos de negocio. Coreografía y orquestación. Manejo de transacciones y compensaciones. Servicios de directorio (JNDI, UDDI, etc).
- Configuración. Reemplazo de configuración por convenciones. Inyección de dependencias. Inversión de control. Contenedores y microcontenedores. Arquitecturas extensibles. Arquitecturas basadas en plugins. Lenguajes de scripting.
- Arquitecturas de seguridad. Integración de métodos de autenticación y autorización en una aplicación. Patrones. Single Sign-on. Acceso basado en roles. Perspectivas de seguridad de una aplicación: seguridad web, sistema operativo, base de datos, middleware.
- Diseño de APIs y Frameworks. Fluent interfaces.
- Estrategias de verificación de arquitecturas. Procesos formales de evaluación de la arquitectura y de los requerimientos no funcionales. Aseguramiento de la adecuación de un sistema a la arquitectura definida, automatización de aseguramiento. Herramientas arquitecturales para la automatización de pruebas de dominio.
- Arquitecturas concurrentes y distribuidas. Objetos distribuidos. Máquinas virtuales distribuidas. Programación orientada a agentes.
- Herramientas tecnológicas para soportar las decisiones arquitectónicas.
- Cuestiones organizacionales, humanas y sociales relativas a la arquitectura de software. Relación entre la arquitectura y el grupo de desarrollo. Comunicación de la arquitectura: modelos, vistas y perspectivas. Herramientas y prácticas que complementan una arquitectura para poder llevar adelante un desarrollo grande y/o complejo.

**Carga horaria semanal:** 6 hs

**Carga horaria total:** 108 horas (Presenciales: 68 horas / Semipresenciales: 40 horas)

**Programa analítico:**

Unidad 1: Arquitectura de software y arquitectura de sistemas. Definición, objetivos y clasificación

Unidad 2: Actividades en un proyecto de software relacionadas con la arquitectura. Proceso de definición y evolución de una arquitectura en diferentes metodologías de desarrollo. Arquitectura en metodologías ágiles. Construcción de prototipos como herramienta de verificación y documentación de una arquitectura.

Unidad 3: Insumos para la definición de arquitectura: requerimientos funcionales y no funcionales, restricciones, influencias, entorno social y técnico, estándares, herramientas disponibles. Objetivos de una arquitectura: no intrusividad, no duplicación, separación de responsabilidades, garantía de atributos de calidad, robustez.

Unidad 4: Estilos arquitectónicos. Arquitecturas en capas, arquitecturas orientadas a servicios, arquitecturas orientadas a objetos, arquitecturas orientadas a procesos (BPM). Patrones arquitecturales. Limitaciones del paradigma de objetos y las arquitecturas tradicionales. Programación declarativa.

Unidad 5: Arquitectura de dominio. Modelado de la lógica de dominio. Patrones. Diseño guiado por el dominio (DDD). Motores de reglas. Workflows.

Unidad 6: Integración de los componentes de una arquitectura: lógica de dominio, interfaz de usuario, persistencia, seguridad, etc. Integración basada en aspectos (AOP). Acoplamiento y comunicación entre los componentes de la arquitectura. Transporte de la información. Manejo de transacciones. Tácticas para garantizar robustez y modificabilidad.

Unidad 7: Patrones arquitecturales para la interfaz de usuario. Integración con el dominio.

Unidad 8: Arquitectura de persistencia. Impacto de la persistencia sobre un diseño orientado a objetos. Bases de datos multidimensionales, datawarehouse, data mining.

Unidad 9: Integración de aplicaciones. Clasificación de los mecanismos de integración: base de datos, dominio, servicios, interfaz de usuario. Estrategias de integración apropiadas para ambientes compatibles e incompatibles entre sí Integración sincrónica y asincrónica. Colas de mensajes. Callbacks. Arquitecturas orientadas a servicios. Web services. Integración con aplicaciones legacy. Patrones para la integración: punto a punto, middleware, Enterprise Service Bus. Definición de interfaces y conectores. Definición de procesos de negocio. Coreografía y orquestación. Manejo de transacciones y compensaciones. Servicios de directorio (JNDI, UDDI, etc).

Unidad 10: Configuración. Reemplazo de configuración por convenciones. Inyección de dependencias. Inversión de control. Contenedores y microcontenedores. Arquitecturas extensibles. Arquitecturas basadas en plugins. Lenguajes de scripting.

Unidad 11: Arquitecturas de seguridad. Integración de métodos de autenticación y autorización en una aplicación. Patrones. Single Sign-on. Acceso basado en roles. Perspectivas de seguridad de una aplicación: seguridad web, sistema operativo, base de datos, middleware.

Unidad 12: Diseño de APIs y Frameworks. Fluent interfaces.

Unidad 13: Estrategias de verificación de arquitecturas. Procesos formales de evaluación de la arquitectura y de los requerimientos no funcionales. Aseguramiento de la adecuación de un sistema a la arquitectura definida, automatización de aseguramiento. Herramientas arquitecturales para la automatización de pruebas de dominio.

Unidad 14: Arquitecturas concurrentes y distribuidas. Objetos distribuidos. Máquinas virtuales distribuidas. Programación orientada a agentes.

Unidad 15: Herramientas tecnológicas para soportar las decisiones arquitectónicas.

Unidad 16: Cuestiones organizacionales, humanas y sociales relativas a la arquitectura de software. Relación entre la arquitectura y el grupo de desarrollo. Comunicación de la arquitectura: modelos, vistas y perspectivas. Herramientas y prácticas que complementan una arquitectura para poder llevar adelante un desarrollo grande y/o complejo.

#### **Bibliografía obligatoria:**

- Fowler, M., Patterns of Enterprise Application Architecture, Addison Wesley, 1ra. Edición, 2002.
- Evans, E., Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison Wesley, 1ra. Edición, 2003.
- Hohpe, G., Woolf, B., Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions, Addison Wesley, 1ra. Edición, 2003.

#### **Bibliografía de consulta**

- Beck, K., Andres, C., Extreme Programming Explained: Embrace Change, Addison Wesley, 2da. Edición, 2004.
- DeMarco, T., Lister, T., Peopleware: Productive Projects and Teams, Addison Wesley, 3ra. Edición, 2013.
- Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D., Refactoring: Improving the Design of Existing Code, Addison Wesley, 1ra. Edición, 1999.

#### **Organización de las clases:**

El proceso de Enseñanza – Aprendizaje se desarrollará a través de los siguientes métodos:

- Clases teórico-prácticas en las que se incentivará la participación activa de quienes cursen.
- Proyecto final de aplicación de los conceptos aprendidos en clase.
- Prácticas de Laboratorio donde se enfoca la consulta acerca del proyecto.
- Seguimiento de proyecto a través de herramientas de aplicación de arquitectura.

#### **Trabajos Prácticos**

Durante las clases prácticas, la principal tarea es el desarrollo de un sistema similar al de un caso del mercado laboral en el cual se insertarán quienes cursan la materia. El cuerpo docente cumplirá el rol de cliente para establecer los requerimientos que debe tener el sistema.

El objetivo es el que el trabajo sea en grupos de estudiantes y que en el mismo, se apliquen las tecnologías que deben adquirirse en la materia, de tal manera que los contenidos sean aprendidos en la aplicación de los mismos en un proyecto.

El trabajo tiene varias etapas. En cada una de las etapas, el cuerpo docente indicará cuál es la evolución esperada. El resultado de cada una deberá ser un producto de software, que debe contener varias partes:

1. El código en sí, versionado sobre un repositorio accesible por el cuerpo docente.
2. El producto funcionando sobre un servidor accesible en Internet.
3. Herramientas de monitoreo y administración del producto (por ejemplo logging, entre otras).
4. Documentación del producto, incluyendo funcionamiento interno, las formas de instalación y ejecución y las justificaciones de las decisiones que se fueron tomando, sobretudo en lo que implican el uso de arquitecturas.

La evaluación de la materia involucra todos estos aspectos, no sólo en el resultado final sino en el desarrollo a lo largo del cuatrimestre. Con respecto al proceso, se espera que quienes cursen la materia cuenten con herramientas para administrar y planificar las tareas pendientes, involucrando al cuerpo docente.

*Parte de la carga horaria de esta asignatura forma parte del grupo de espacios curriculares de la carrera que implementan horas de Instancias de Formación de Prácticas Profesionales Supervisadas (IFPPS), que se regulan a través del Reglamento aprobado por Res. (CDCyT) N°034/21, o cualquier otra resolución que la modifique o la reemplace. En estas instancias quienes cursen deben cumplir con la evaluación por la aplicación de los conceptos teóricos/prácticos aprendidos en relación con los objetivos de aprendizaje. Así mismo se formulan otros objetivos vinculados a la solución de un problema del mercado laboral -en el cual se insertarán como profesionales. De esta manera, en lo que respecta a las horas totales de la materia, se dedican 32 horas totales para las IFPPS y las 76 horas restantes para clases teóricas y prácticas (trabajos prácticos usuales) necesarias para el aprendizaje de los conceptos.*

*Esta asignatura forma parte del grupo de las asignaturas con horas presenciales y semipresenciales que están reguladas a través de la Resolución del CS 052/21, o cualquier otra resolución que la modifique o la reemplace.*

*Las asignaturas con esta modalidad se desarrollan con horas presenciales y encuentros virtuales en base a las necesidades de las asignaturas, sin alterar la*

*metodología de evaluación de las materias en modalidad presencial. Los encuentros virtuales proveen contenidos, materiales y ejercicios de distintos formatos que incorporen las ventajas de la plataforma virtual indicada por la Universidad.*

**Modalidad de evaluación:**

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18).

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

### CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*			Evaluación
		Teórico	Práctico		
			Res Prob	Lab.	
1	Arquitectura de software y arquitectura de sistemas.	x			
2	Actividades en un proyecto de software relacionadas con la arquitectura.	x		x	
3	Insumos para la definición de arquitectura.	x		x	
4	Estilos arquitectónicos.	x		x	x
5	Arquitectura de dominio.	x		x	
6	Integración de los componentes de una arquitectura.	x		x	
7	Patrones arquitecturales para la interfaz de usuario.	x		x	
8	Arquitectura de persistencia.	x		x	x
9	Integración de aplicaciones.	x		x	

10	Configuración.	x		x		
11	Arquitecturas de seguridad.	x		x		
12	Diseño de APIs y Frameworks.	x		x		x
13	Estrategias de verificación de arquitecturas.	x		x		
14	Arquitecturas concurrentes y distribuidas.	x		x		
15	Herramientas tecnológicas para soportar las decisiones arquitectónicas.	x		x		
16	Cuestiones organizacionales, humanas y sociales relativas a la arquitectura de software.	x		x		x

**\*indique con una cruz la modalidad**