



ASIGNATURA:	TECNOLOGÍAS AVANZADAS EN LA CONSTRUCCIÓN DE SOFTWARE
DEPARTAMENTO:	ING. EN SIST. DE INFORMACION
AREA:	ELECTIVA
BLOQUE	TECNOLOGÍAS APLICADAS

MODALIDAD:	Cuatrimestral
HORAS SEM.:	6 horas
HORAS/AÑO:	96 horas
HORAS RELOJ	72
NIVEL:	4°
AÑO DE DICTADO:	2018

Objetivos

- Conocer y dominar las distintas tecnologías comúnmente utilizadas en la actualidad para la construcción de software, como así también aquellas que podrían ser utilizadas en el futuro, sentando las bases para la investigación posterior por parte del alumno devenido en profesional de la materia.
- Comprender las tendencias actuales y futuras del mercado de desarrollo, fomentando a la vez la formación de ideas propias que le permitan aprovechar las herramientas y conceptos más útiles en cada caso, independientemente de la herramienta “de moda”.
- Adquirir la capacidad de incorporar distintos tipos de tecnologías o frameworks en su proceso de desarrollo, a partir de:
 - o Identificar problemas que podrían ser resueltos por herramientas preexistentes
 - o Investigar los distintos tipos de herramientas comerciales y código abierto existentes en el mercado
 - o Seleccionar una herramienta a partir de la comprensión de sus ventajas y desventajas y la relación de dichas características con los requerimientos funcionales y no funcionales del producto de software que está desarrollando
 - o Aprender por sus propios medios a utilizar las nuevas herramientas y frameworks seleccionados, combinando distintas formas de análisis de las mismas
 - o Diferenciar entre cada aplicación que se construye y las tecnologías y frameworks utilizados en la construcción, comprendiendo la conveniencia de mantener independencia respecto de decisiones tecnológicas puntuales, y conociendo formas de separar responsabilidades entre esos componentes en una pieza compleja de software.



- Comprender las bases del funcionamiento técnico y conceptual de herramientas de construcción de software, con el objetivo de optimizar su valor, entender e incorporar los conceptos de diseño utilizados en la construcción de distintas herramientas aprovechando las bases conceptuales que guían una arquitectura de software construida sobre un conjunto de herramientas dado, extender y/o modificar las herramientas que así lo permitan y sentar las bases para la construcción de sus propias herramientas y frameworks.
- Acercarse a la complejidad de las herramientas más utilizadas, comprendiendo su filosofía de funcionamiento y arquitectura, con el objetivo de abolir la distancia ficticia que se suele establecer entre el software que se puede adquirir en el mercado internacional y el que el propio profesional puede construir con un equipo dedicado a tal fin.

Contenidos Mínimos (Programa Sintético).

- Tecnologías y frameworks en el desarrollo de software: Conceptos de frameworks, arquitectura y contenedor. Clasificaciones.
- Tecnologías de Persistencia: Mecanismos de persistencia, problemas típicos.
- Tecnologías de Interfaz de usuario: Tecnologías utilizadas y conceptos de diseño de software aplicados a la construcción de interfaces de usuarios
- Programación concurrente y tecnologías de integración: Conceptos. Características. Implementaciones. Ventajas. Problemáticas.
- Metaprogramación: Conceptos herramientas y tecnologías de programación dinámica, y metaprogramación.
- Herramientas de configuración, administración y despliegue de aplicaciones: Conceptos. Características. Implementaciones. Ventajas. Problemáticas.
- Incorporación de Seguridad: Impacto en la selección de estrategias, granularidad de los elementos a incorporarlos seguridad.
- Integración de tecnologías y paradigmas en el desarrollo de software: Conceptos y herramientas necesarios para la construcción de software multilinguaje.

Contenidos Analíticos

Unidad I: Tecnologías y frameworks en el desarrollo de software.

Concepto de framework y de arquitectura. Frameworks de caja blanca y caja negra. Concepto y técnicas en el diseño de frameworks. Concepto de contenedor. Contenedores pesados y livianos. Metodologías de interacción con frameworks comerciales y de código fuente abierto. Cualidades de Software. Desarrollo Guiado por Tests

Unidad II: Tecnologías de persistencia.

Introducción a los mecanismos de persistencia en los sistemas de software. Problemas técnicos de la persistencia. Selección y combinación de mecanismos de persistencia. Mapeo O/R. Bases de datos orientadas a objetos.



Unidad III: Tecnologías de Interfaz de usuario

Tecnologías de presentación basadas en formularios. Tecnologías de presentación orientadas a eventos. Diseño de Interfaces Web. Clientes pesados y livianos. Clientes RIA.

Unidad IV: Programación concurrente y tecnologías de integración.

Conceptos de programación concurrente y mecanismos de sincronización. Integración por mensajes y colas. Integración por Web Services.

Unidad V: Metaprogramación:

Capacidades reflexivas y uso de meta-datos en la construcción del software. Utilización y construcción de herramientas declarativas en el desarrollo de sistemas de software. Herramientas habituales de metaprogramación: Annotations, XML, Reflection, Conceptos de mirroring y reificación.

Unidad VI: Herramientas de configuración, administración y despliegue de aplicaciones.

XML. Herramientas de versionado y administración de código fuente, entregables y dependencias. Introducción a los mecanismos de configuración y despliegue sobre distintos tipos de servidores.

Integración continua. Inyección de dependencias.

Unidad VII: Incorporación de Seguridad:

Estrategias comúnmente utilizadas para la incorporación de seguridad en una aplicación

Unidad VIII: Integración de tecnologías y paradigmas en el desarrollo de software:

Introducción de mecanismos declarativos en aplicaciones empresariales. Utilización de lenguajes dinámicos y distintos mecanismos de integración. Programación orientada a aspectos. Test-driven development. Extensiones al paradigma orientado a objetos.

Bibliografía

- BAUER, C., KING G., *Java Persistence with Hibernate*, Manning (2006), ISBN: 1932394885
- CASEY, J., MASSOL, V., PORTER, B. SANCHEZ, C., *Better Builds with Maven*, DevZuz (2006), <http://www.devzuz.com/web/guest/products/resources#BBWM>
- ECKEL, B., *Thinking in Java*, Prentice Hall (2006); 4th Edition, ISBN: 0131872486
- FOWLER, M., *Analysis Patterns: Reusable Object Models*, Addison-Wesley (1996), ISBN: 0201895420



- FOWLER, M., *Patterns of Enterprise Application Architecture*, Addison-Wesley (2002), ISBN: 0321127420
- GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J. M., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1994), ISBN: 0201633612
- HORSTMANN, C. S. y CORNELL, G., *Core Java(TM), Volume I—Fundamentals*, Prentice Hall (2007); 8th Edition, ISBN: 0132354764
- HORSTMANN, C. S. y CORNELL, G., *Core Java, Vol. 2: Advanced Features*, Prentice Hall (2008); 8th Edition, ISBN: 0132354799
- HUNT, A. y THOMAS D., *The Pragmatic Programmer: From Journeyman to Master*, Addison-Wesley (1999), ISBN: 020161622X
- JOHNSON, R. E. *How to Design Frameworks*. University of Illinois at Urbana-Champaign. 1993. <http://st-www.cs.uiuc.edu/users/cs497/notes/day20.pdf>
- JOHNSON, R. y HOELLER, J., *Expert One-on-One J2EE Development without EJB*, Wrox (2004), ISBN: 0764558315
- KAN, S. H., *Metrics and Models in Software Quality Engineering*, Addison-Wesley (2002); 2nd Edition, Capítulos 1, 2, 11, 12, ISBN: 0201729156
- LEE, K. A., *The Buildmeister's Guide - How to design and implement the right software build and release process for your environment*, Lulu.com (2006); 1st Edition, ISBN: 184728373X
- MASSOL, V., O'BRIEN, M. T., *Maven: A Developer's Notebook*, O'Reilly (2005); 1st Edition, ISBN: 0596007507
- PAWSON, R., MATTHEWS R., *Naked Objects*, Wiley, 1st edition (2002), ISBN: 0470844205
- SRIGANESH, R. P., BROSE G., SILVERMAN M., *Mastering Enterprise JavaBeans 3.0*, Wiley (2006), ISBN: 0471785415
- *Software Configuration Management (SCM)*, Software Engineering Institute (SEI) - Carnegie Mellon University, <http://www.sei.cmu.edu/legacy/scm/>
- TATE, B., *Beyond Java*, O'Reilly (2005); 1st Edition, ISBN: 0596100949

Correlativas

Para Cursar:

Cursadas:

- Diseño de Sistemas
- Sistemas Operativos
- Gestión de Datos

Aprobadas:

- Análisis de Sistemas
- Sintaxis y Semántica del Lenguaje

- Paradigmas de Programación

Para rendir:

Aprobadas:

- Diseño de Sistemas
- Sistemas Operativos
- Gestión de Datos