



INSTITUTO/S: Tecnología e Ingeniería

CARRERA/S: Licenciatura en Informática

MATERIA: Arquitectura de software II

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Lic. Ricardo Brea

EQUIPO DOCENTE: Lic. Ricardo Brea

CUATRIMESTRE: 2^{do}

AÑO: 5^{to}

PROGRAMA N°: 40

(Aprob. Por Cons.Directivo fecha XX)

Instituto/s: Tecnología e Ingeniería

Carrera/s: Licenciatura en Informática

Nombre de la materia: Arquitectura de software II

Responsable de la asignatura y equipo docente: Ricardo Brea

Cuatrimestre y año: 2^{do} cuat. del 5^{to} año

Carga horaria semanal: 4 hs

Programa N°: 40

Código de la materia en SIU: 781

Arquitectura de software II

1. Fundamentación

La arquitectura de software es una disciplina que integra los conocimientos adquiridos en muchas de las asignaturas de la carrera. Aportando sinergia a los distintos componentes de un sistema.

La materia se centra en los requerimientos no funcionales, principalmente Eficiencia y Eficacia, como ejes conductores de los distintos temas. Incorporando ejemplos prácticos utilizando tecnologías actuales como Microservicios y servicios en la nube. Con el objetivo de aportar herramientas para un análisis crítico entre distintas alternativas y opciones de acuerdo a posibles escenarios que se pueden encontrar en la práctica profesional.

2. Propósitos y objetivos

Objetivos

Son objetivos de esta materia que las/os estudiantes

- Puedan realizar un análisis crítico de distintas alternativas tecnológicas basado en los requerimientos y evitando sesgos.
- Comprendan el concepto de eficiencia y eficacia dentro de la norma ISO25010
- Identifiquen problemas de performance y posibles soluciones en sistemas ya implementados.
- Conozcan tecnologías actuales, sus ventajas y desventajas, para diseñar, desarrollar e implementas un sistema teniendo en cuenta metodologías ágiles

3. Programa sintético:

Escalabilidad, eficiencia y efectividad. Dimensionamiento de los requerimientos de hardware y de las necesidades de red de un sistema de software. Técnicas para escalamiento vertical y horizontal. *Clustering*, balanceo de carga, afinidad, *sharding*. Estrategias de particionamiento de bases de datos. Tolerancia a fallos. Estrategias de cache de datos. Nociones de minería de datos. Hardware específico para sistemas de gran envergadura. Virtualización. Software y hardware como servicios. Verificación del cumplimiento de los requerimientos no funcionales: performance, tolerancia a fallos, carga. Operación y monitoreo de sistemas. Estrategias de *logging* para sistemas de gran envergadura. Herramientas para medición de performance. *Profiling*. Información caliente e información de ciclo de vida largo. Análisis de servicios en red, análisis de tráfico. Herramientas de monitoreo de fallas.

4. Programa analítico

4.1 Organización del contenido:

Unidad 1: Eficiencia, Eficacia

Definición de eficiencia y eficacia. Requerimientos no funcionales dentro de la norma ISO25010. Comparación de alternativas tecnológicas utilizando Matriz de Pugh.

Unidad 2: Microservicios

Docker, definición de imagen, contenedor, registro de imágenes. Arquitectura interna de Docker. *Docker Compose*. Declaración imperativa y declarativa de recursos. Kubernetes, historia, arquitectura interna.

Práctica: Ejemplo prácticos de Microservicios utilizando *Docker Compose* para implementar soluciones multi contenedor.

Unidad 3: Escalabilidad

Definición de escalabilidad. Escalabilidad vertical, ventajas, desventajas y casos de uso. Escalabilidad horizontal ventajas, desventajas y casos de uso. Paralelismo, utilización del CPU multi núcleo. Balanceo de carga, algoritmos de balanceo de carga. Métodos de particionamiento de sistemas de base de datos. Data *Sharding*.

Práctica: YAML y recursos en Kubernetes. Escalabilidad horizontal, definición de *Pod* y recursos. Escalabilidad vertical, definición de *Deployment* y *Service*.

Unidad 4: Confiabilidad, Tolerancia a fallos y alta disponibilidad

Definición del requerimiento no funcional Confiabilidad. Definición de Tolerancia a Fallos (*Fault Tolerance*) y alta disponibilidad (*High Availability*). Único punto de falla, redundancia y detección de fallos. Medición de la disponibilidad. Recuperación de Desastres. Buenas prácticas. Tipos de despliegue para alta disponibilidad. *Clustering* en Kubernetes. “.

Práctica: Tolerancia a fallo en Kubernetes (*Rediness Probe, Liveness Probe*)

Unidad 5: Dimensionamiento y Monitoreo

Concepto de dimensionamiento. Recursos a dimensionar. Metodologías de dimensionamiento. Observabilidad y su rol fundamental en la mejora continua. Identificación de “cuellos de botella”. Bases de datos de series de tiempo y visualización de los datos. Logging. Estructura de un Log. Diferencias con Auditoria (ISO25010 Responsabilidad / *Accountability*). Tecnologías de logging. Integración y agregación de logs. Id de correlación de logs. Buenas practicas

Práctica: *Kubernetes Monitoring Operator. Kubernetes Logging Operator*

Unidad 6: Nube

Concepto de Nube (Cloud Computing) y sus características. Modelos de implementación. SaaS, PaaS y IaaS. Casos de uso. Almacenamiento en la nube (Caliente, de ciclo de vida largo). Datamining y análisis de grandes volúmenes de dato.

Práctica: Cálculo de costos en Azure y AWS.

Unidad 7: SSL / TLS / HTTPS

Concepto de proxy reverso (Reverse Proxy). Repaso modelo OSI. Funcionamiento de *Secure Socket Layer, Transport Layer Security*, y HTTPS. *IngressController* en Kubernetes.

Práctica: Implementación básica en Kubernetes con *Ingress Controller, Certificate Manager, Lets encrypt*.

Unidad 8: Diseño de Software como servicio

Se describen las características/factores de una solución para que sea implementable como servicio. Código base. Dependencias. Configuración, Servicios externos, Compilar/empaquetar/ implementar. Estado de una aplicación. Asignación de Puertos. Concurrencia. Terminación de procesos y liberación de recursos. Homogeneidad de ambientes.

Práctica: Manejo del estado de un sistema en Kubernetes utilizando *ConfigMaps, Secrets* y *Volumes*

Unidad 9: DevOps e Integración continua

Historia del concepto *DevOps*. Relación con la metodología ágiles y las problemáticas de su implementación. Soluciones propuestas. Integración Continua, control de versiones y estrategias de *branching* en GIT. Concepto *DevSecOps* y la seguridad desde el inicio del desarrollo. Análisis estático de código. Mejora continua. OWASP. Entrega continua,

Práctica: Integración continua y entrega continua con microservicios

Unidad 10: Implementación Continua

Problemas de implementación manual en sistemas complejos o de gran envergadura. Implementación continua. Infraestructura como código. Criterios de aceptación y test automáticos de aceptación. Verificación de requerimientos no funcionales

Práctica: Implementación continua con Microservicios

4.2 Bibliografía y recursos obligatorios:

Len Bass, Paul Clements, Rick Kazma. (2021). *"Software Architecture in Practice"*. 4^{ta} Edición. USA: Addison-Wesley Professional.

David Farley, Jez Humble. (2010). *"Continuous delivery: reliable software releases through build, test, and deployment automation"*. USA: Addison-Wesley Professional.

Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. (2002). *"Patterns of Enterprise Application Architecture"*. USA: Addison-Wesley.

C.J. Date. (2001). *"Introducción a los sistemas de base de datos"*. México: Pearson Educación

Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy. (2016). *"Site Reliability Engineering"*. USA: O'Reilly Media, Inc.

Betsy Beyer, Niall Richard Murphy, David K. Rensin, Kent Kawahara, Stephen Thorne. (2016). *"The Site Reliability Workbook"*. USA: O'Reilly Media, Inc.

Peter Mell, Timothy Grance. (Septiembre 2011). "The NIST Definition of Cloud Computing". NIST Technical Series Publications. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

Andrej Dyck; Ralf Penners; Horst Lichter. (2015). *"Towards Definitions for Release Engineering and DevOps"*. (Mayo 2015). Institute of Electrical and Electronics Engineers. https://www2.swc.rwth-aachen.de/docs/2015_RELENG_Dyck.pdf

Documentación de Kubernetes. Sitio oficial de Kubernetes. <https://kubernetes.io/es/docs/home/>

4.3 Bibliografía optativa:

No aplica.

5. Metodología de enseñanza:

Las clases están divididas en dos mitades. Una mitad teórica y la otra mitad práctica. Durante la mitad teórica se presenta el contenido formal del próximo tema práctico a realizar. Durante

las porciones práctica se muestran ejemplos prácticos de la teoría que luego deberán ser aplicación en el trabajo practico final.

Al finalizar cada clase las/os alumnos disponen de un cuestionario de auto evaluación. En base a estos cuestionarios se hará la evaluación teórica

Plan de trabajo en el campus:

En el aula virtual se propondrá material educativo;

- Clases grabadas
- Presentaciones que el/la docente emplea durante las clases
- Vinculo al repositorio de código (abierto) con los ejemplos prácticos presentados en clase
- Videos y textos introductorios o ampliatorios de los temas tratados.
- Guías de trabajos prácticos y todo material que el/la alumno/a deba entregar.
- También incluirá un foro de consultas, el programa, el cronograma de la asignatura y cualquier tipo de información adicional que sea necesaria.

6. Actividades de investigación y extensión (si hubiera)

No aplica.

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a las cursadas de materias virtuales se requerirá que el estudiante ingrese al aula virtual como mínimo una vez por semana.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial

menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente.

8. Cronograma

CLASE	UNIDAD	TIPO DE CLASE	MODALIDAD	ACTIVIDAD
1	-	INTRODUCCION A LA MATERIA	VIRTUAL	-
	1	TEORICA		CUESTIONARIO AUTO EVAVALUACION
	2	TEORICA Y PRACTICA		CUESTIONARIO AUTO EVAVALUACION
2		FERIADO		
3	3	TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
4	4	TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
5	5	TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
6		TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
7	6	TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
8	-	PARCIAL TEORICO	PRESENCIAL	-
9	-	PRESENTACION TRABAJO PRACTICO	VIRTUAL	-
	7	TEORICA Y PRACTICA	VIRTUAL	CUESTIONARIO AUTO EVAVALUACION
10	-	FERIADO		
11	8	TEORICA Y PRACTICA	VIRTUAL	OPCIONAL ETREGA / CORRECCION TP CUESTIONARIO AUTO EVAVALUACION

12	9	TEORICA Y PRACTICA	VIRTUAL	OPCIONAL ETREGA / CORRECCION TP CUESTIONARIO AUTO EVAVALUACION
13	10	TEORICA Y PRACTICA	VIRTUAL	OPCIONAL ETREGA / CORRECCION TP CUESTIONARIO AUTO EVAVALUACION
14	-	PARCIAL	PRESENCIAL	FECHA FINAL ENTREGA TRABAJO PRACTICO
15	-	RECUPERATORIO	PRESENCIAL	
16		FIRMA FEEDBACK DE LOS ALUMNOS FIRMA	PRESENCIAL	