

Carrera: Ingeniería en Sistemas de Información**Asignatura:** Backend de Aplicaciones**Planificación a partir del Ciclo Lectivo 2025****1. Datos administrativos de la asignatura**

Nivel en la carrera	3	Duración	Cuatrimestral
Plan	2023		
Bloque curricular:	Asignatura Electiva		
Carga horaria presencial semanal (hs. cátedra):	8	Carga Horaria total (hs. reloj):	96
Carga horaria no presencial semanal (hs. reloj) (si correspondiese)		% horas no presenciales (hs. reloj) (si correspondiese)	

2. Presentación, Fundamentación**Introducción**

Es un hecho que las soluciones de software se construyen en base a la integración de un conjunto de componentes que se conectan para lograr cubrir las necesidades del usuario y, que en la actualidad, en ese conjunto de componentes conviven: frameworks de desarrollo, componentes desarrollados para la solución y componentes que existían previamente.

Por otro lado, una buena porción de las soluciones actuales, tienen múltiples vías de interacción con el usuario, sin embargo, en general hay una única capa denominada backend, a la que las diferentes alternativas de interfaz se conectan para enviar peticiones y recibir respuestas con datos o información. Y, que en esta capa, conviven distintos componentes para cubrir un amplio abanico de funciones específicas, que van desde la identificación del usuario y sus permisos de acceso, a la mantención del estado y la implementación de la lógica propia de la solución, soportadas además por una infraestructura capaz de contener estos componentes y las herramientas de monitoreo y administración de dicha infraestructura.

La presente asignatura tiene por función, revisar las diferentes alternativas existentes para soportar los roles mencionados e implementar una solución que integre la utilización e interacción con frameworks y componentes existentes con la codificación de los componentes necesarios que en su conjunto dan vida a la lógica de la solución, desplegar dicha solución en un entorno de

infraestructura y ponerlo a prueba de forma tal que pueda dar servicio a las interfaces de usuario que la requieran.

En la asignatura Desarrollo de Software, el estudiante se enfrenta por primera vez con los distintos componentes de la capa de backend de una solución y el obtiene una visión general de la solución y específica de las problemáticas del backend y sus dependencias, sus componentes y las distintas herramientas de soporte, que hacen falta para su construcción.

En esta asignatura el estudiante se enfocará en la capa Backend y profundizará acerca de los desafíos que se enfrentan al construir esta capa y las alternativas de solución existentes para la misma en el mercado. Comprenderá las ventajas y desventajas de cada alternativa y será capaz de identificar estas características en los componentes que utilizará para la construcción del Backend de la solución.

Al finalizar la materia el estudiante tendrá una idea clara de los distintos elementos que componen el backend de una solución de software, las herramientas transversales requeridas para su construcción, las herramientas y alternativas de prueba del propio backend y la infraestructura necesaria para su despliegue y puesta en marcha.

En resumen, al finalizar la materia el estudiante estará capacitado para enfrentar la construcción del Backend de una solución o formar parte de un equipo de desarrollo de software cumpliendo dicha función, tanto para un trabajo final de carrera, como profesionalmente en el ámbito laboral.

3. Relación de la asignatura con las competencias de egreso de la carrera

En la tabla siguiente se establece la relación de la asignatura con las competencias de egreso: Específicas, Genéricas Tecnológicas y Genéricas Sociales, Políticas y Actitudinales de la carrera. Se incluyen las competencias de egreso a las que tributa, aportes reales y significativos de la asignatura, y en qué nivel (no aporta, bajo, medio, alto).

Competencias	Nivel
Competencias genéricas tecnológicas (CG):	
CG.1. Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.	Bajo
CG.2. Concepción, diseño y desarrollo de proyectos de Ingeniería en Sistemas de Información/Informática	No aporta

CG.3. Gestión, planificación, ejecución y control de proyectos de Ingeniería en Sistemas de Información/Informática.	No aporta
CG.4. Utilización de técnicas y herramientas de aplicación de Ingeniería en Sistemas de Información/Informática.	Bajo
CG.5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	No aporta
Competencias genéricas sociales, políticas y actitudinales (CG)	
CG.6. Fundamentos para el desempeño en equipos de trabajo.	Bajo
CG.7. Fundamentos para una comunicación efectiva.	No aporta
CG.8. Fundamentos para una actuación profesional ética y responsable.	No aporta
CG.9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad profesional en el contexto global y local.	No aporta
CG.10. Aprender en forma continua y autónoma.	Bajo
CG.11. Fundamentos para el desarrollo de una actitud profesional emprendedora	No aporta
Competencias Específicas de la carrera	
CE1.1. Especificar, proyectar y desarrollar sistemas de información para concebir soluciones tecnológicas que permitan resolver situaciones en las organizaciones mediante el empleo de metodologías de sistemas y tecnologías asociadas a los sistemas de información.	Bajo
CE1.2. Especificar, proyectar y desarrollar sistemas de comunicación de datos, evaluando posibles soluciones tecnológicas disponibles para dar soporte a los sistemas de información en lo referido al procesamiento y comunicación de datos.	No aporta
CE1.3. Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.	Medio
CE2.1. Proyectar y dirigir lo referido a seguridad informática para seleccionar y aplicar técnicas, herramientas, métodos y normas, garantizando la seguridad y privacidad de la información procesada y generada por los sistemas de información.	Bajo
CE.3.1. Establecer métricas y normas de calidad de software para medir, evaluar, controlar y monitorear el rendimiento, impulsando mejoras de acuerdo a técnicas y normas vigentes definidas por los organismos de estandarización.	No aporta

CE.4.1. Certificar el funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software para asegurar la generación de los resultados deseados en función de restricciones de tiempo y recursos establecidos.	No aporta
CE.5.1. Dirigir y controlar la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de alcanzar los objetivos fijados por la organización.	No aporta
CE.6.1. Asesorar y capacitar a organizaciones, empresas, organismos públicos o privados en la adquisición, instalación y uso, en lo que respecta a sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software, a los fines de un uso correcto de los sistemas intervinientes.	No aporta
CE.7.1. Realizar pericias, tasaciones y arbitrajes relacionados con su actividad profesional, respetando marcos normativos y jurídicos con el objeto de asesorar a las partes o a los tribunales de Justicia.	No aporta

4. Contenidos Mínimos

No corresponde.

5. Objetivos establecidos en el DC

No corresponde.

6. Resultados de aprendizaje

Los siguientes resultados de aprendizaje se promueven en el desarrollo de la asignatura

Identificador de RA	Redacción
RA1	Aplicar un lenguaje de programación actual utilizando buenas prácticas, elementos de código seguro, validaciones de estructuras y herramientas de soporte para construir los componentes codificables del Backend de una aplicación.
RA 2	Examinar los diferentes Frameworks involucrados en el back en de una aplicación, con base en las especificaciones existentes de los mismos y con la finalidad de decidir cuál resuelve mejor las necesidades de la aplicación con los recursos disponibles.
RA 3	Construir el backend de una solución de software, para dar servicio a las distintas interfaces de dicha solución, utilizando gateway, ruteo, autenticación, autorización y lógica de aplicación y teniendo en cuenta buenas prácticas y herramientas transversales de soporte al proceso de desarrollo.

RA 4	Diferenciar los distintos aspectos de la seguridad del backend de una aplicación para tener conocimiento de los mecanismos que permiten construir software con alto nivel de seguridad informática, en función de las necesidades y criticidad del sistema a desarrollar
RA 5	Reconocer herramientas de monitoreo de aplicaciones, centralización de logs y automatización de despliegue, con la finalidad de visualizar los aspectos transversales a la construcción del software y necesarios para su puesta en marcha.

7. Relación de los RA y las competencias

En la tabla siguiente se indica con X la tributación de cada Resultado de Aprendizaje con las competencias de egreso: específicas, genéricas tecnológicas, sociales, políticas y actitudinales de la carrera.

RA	CE1.1	CE1.2	CE1.3	CE2.1	CE3.1	CE4.1	CE5.1	CE6.1	CE7.1	CG1	CG2	CG3	CG4	CG5	CG6	CG7	CG8	CG9	CG10	CG11
RA1			X							X									X	
RA 2	X		X										X		X				X	
RA 3	X		X							X			X						X	
RA 4	X			X									X						X	
RA 5	X												X							

8. Asignaturas correlativas previas

Para cursar y rendir debe tener cursadas:

- Asignatura/s:
Paradigmas de Programación
Sintaxis y Semántica de los Lenguajes

Para cursar y rendir debe tener aprobada:

- Asignatura/s:
Algoritmos y Estructuras de Datos

9. Asignaturas correlativas posteriores

Indicar las asignaturas correlativas posteriores:

- Asignatura/s: algunas asignaturas del conjunto definido a continuación.
Arquitectura de Software
Gestión Industrial de la Producción
Gestión de la Mejora de los Procesos
Comunicación Multimedial en el desarrollo de Sistemas de Información
Desarrollo y Operaciones DevOps
Ingeniería de Software de Fuentes Abiertas/Libres
Integración de Aplicaciones en Entorno Web
Seguridad en el Desarrollo de Sistemas
Desarrollo con Tecnologías Blockchain

10. Programa analítico

Este programa analítico contempla los contenidos mínimos, previstos en el DC vigente, y aquellos que se consideran necesarios para desarrollar los resultados de aprendizaje propuestos.

Unidad N°: 1

Título: **Introducción al Backend de Aplicaciones**

Contenidos:

- Estructura general de las Arquitecturas de Microservicios (Gateway/Proxy, Ruteo, Intercepción, Autenticación, Autorización, Orquestación y Logueo de aplicaciones)
- Introducción al modelado de Arquitectura (C4 Model)
- Definición de API Rest (a partir de los conceptos de DDS)
- Java Microprofile (implementaciones de referencia / frameworks alternativos)

Carga horaria por Unidad: 8 h

Unidad N°: 2

Título: **Lenguaje de Programación Java**

Contenidos:

- Elementos fundamentales de Java
- Sintaxis del Lenguaje de programación Java
- Programación Orientada a Objetos en Java
- Template de proyectos y gestión de dependencias (Maven)
- Principales APIs incluidas en Java
- Reducción de Boiler Plate
- Testing Unitario en Java

Carga horaria por Unidad: 32 h

Unidad N°: 3

Título: **Microservicios en Java**

Contenidos:

- Framework Spring, implementación con Spring boot. Alternativas y comparación con otras plataformas.
- Implementación y prueba de Servicios básicos.
- Estructura de contenedores y despliegue en entornos de prueba.
- Acceso a base de datos y ORM
- Gateway de servicios, Ruteo e Intercepción
- Patrones de Implementación
- Principios SOLID, Orquestación de Servicios

Carga horaria por Unidad: 44 h

Unidad N°: 4

Título: **Seguridad en Backend de Aplicaciones**

Contenidos:

- Autenticación de usuarios.
- Autenticación interna vs Identidad Federada, Consumo de servicios de autenticación.
- Diferencia entre Autenticación y Autorización
- Esquemas de Autorización (Confidencialidad, Políticas de mínimos privilegios y Segregación de funciones)
- Patrones de Seguridad del Servidor

- JWT
- Configuración de pruebas de Seguridad en Backend.

Carga horaria por Unidad: 28 h

Unidad N°: 5

Título: **Seguimiento, Infraestructura y Despliegue de Aplicaciones**

Contenidos:

- Nociones básicas IC, Pipelines de Despliegue
- Elementos fundamentales del control de calidad del código fuente (Sonarqube)
- Mantenimiento y configuración básica de redes de contenedores.
- Manejo de excepciones. Log de aplicaciones.
- Centralización de Log de aplicaciones (Stack ELK)

Carga horaria por Unidad: 16 h

Carga horaria por tipo de formación práctica de toda la asignatura

Tipo de formación práctica	Horas reloj
Formación experimental	36 h
Análisis y resolución de problemas de ingeniería y estudios de casos	24 h
Formulación, análisis y desarrollo de proyectos.	12 h

Bibliografía Obligatoria:

- Newman S. (2015). Building Microservices – Designing Fine-Grained Systems. O’Reilly.
- Sharma S. (2016). Mastering Microservices with Java. Packt

Bibliografía optativa y otros materiales a utilizar en la asignatura:

- Gosling J., Joy B., Steele G., Bracha G., Buckley A., Smith D. y Bierman G. (22 de noviembre de 2022). The Java Language Specification – Java SE 17 Edition. <https://docs.oracle.com/javase/specs/jls/se17/html/index.html>
- Spring. (22 de noviembre de 2022). Spring Boot – Overview.
- Martin R. (2008). Clean Code. Pearson
- Howard M. y LeBlanc D. (2003). Writing Secure Code 2nd. Edition. Microsoft Press
- Evans E. (2003). Domain Driven Design. Addison Wesley

11. Metodología de enseñanza

Clases Presenciales

Se prevé que la asignatura tendrá 2 focos bien definidos, por un lado la revisión, estudio y debate sobre alternativas de implementación para cada uno de los tópicos principales. Y, por el otro, la implementación de cada uno de los tópicos aplicando la alternativa elegida. Esta realidad plantea que el seguimiento de la materia va a estar marcado por dos esquemas, cada uno de ellos enfocado en cada uno de los aspectos mencionados.

En cuanto al análisis de alternativas, se implementará

- Revisión guiada de alternativas y estudio de casos.
- Clases de debate y contraposición de cada una de las alternativas con objetivo de selección de la alternativa a implementar.
- Se fomentará que los estudiantes propongan alternativas propias, más allá de las propuestas por la cátedra para su análisis y oposición a las estudiadas.

En cuanto a las clases de implementación

- Los estudiantes realizarán talleres paso a paso de implementación de cada esquema propuesto con el soporte de los docentes.
- Los estudiantes pondrán a prueba los componentes desarrollados mediante las pruebas provistas en base a los elementos provistos en el taller.
- Luego de implementado el taller los estudiantes aplicarán los conceptos del taller en un práctico integrador.

Aula virtual

La asignatura estará guiada por una única aula virtual para toda la cátedra, la misma será implementada en el moodle provisto por la universidad, además la asignatura mantendrá un esquema de nube con las herramientas principales requeridas en los talleres y actividades. En este sentido se implementa un repositorio local de componentes en la universidad, se implementa un servidor de código fuente también dentro de la universidad.

En el aula virtual los estudiantes contarán con:

- Videos explicativos
- Materiales de lectura obligatoria y opcional
- Los materiales de los talleres a resolver
- Otros Ejemplos resueltos

En el aula virtual, además, los estudiantes deberán:

- Responder cuestionarios de autoevaluación
- Participar en foros de discusión
- Comunicarse con los profesores y compañeros

Herramientas de digitales

Por otro lado, se propondrá una herramienta común de comunicación fuera del momento de clase de forma que los estudiantes tengan acceso a comunicación con sus profesores cuando estos lo establezcan.

Finalmente, todos los proyectos básicos propuestos por la cátedra podrán ser descargados del repositorio de la cátedra y todos los servicios a consumir por los estudiantes también podrán ser accedidos en la plataforma que la cátedra proveerá.

12. Recomendaciones para el estudio

La asignatura intentará llevar al estudiante a un entorno de realidad cercana a la de un equipo de desarrollo de backend de una aplicación. En este camino será requerido que el estudiante demuestre las habilidades que se esperarían de él en un equipo de trabajo y que son transversales a los contenidos de la materia como, por ejemplo:

- Proactividad
- Autonomía
- Juicio crítico

Se espera que el estudiante invierta tiempo fuera de las horas de cursado, para enriquecer su propio conocimiento sobre los temas que se desarrollen en la asignatura, y contraste este conocimiento con las ideas abordadas en las clases.

La asignatura prevé que el estudio se enfocará en conocer herramientas existentes, ventajas y desventajas de estas, y en aplicar algunas de ellas para cumplir con el objetivo de construcción de software requerido.

Antes de cada clase será fundamental que el estudiante haya revisado los elementos a debatir o implementar en la clase y/o haya realizado los talleres o actividades previstas. Durante la clase se espera que el estudiante se enfoque en hacer su juicio crítico de los elementos revisados o implemente los talleres paso a paso propuestos y luego de la clase se espera que el estudiante lleve a cabo las actividades de seguimiento de la asignatura.

13. Metodología de evaluación

El modelo de enseñanza basado en competencias implica la aplicación de metodologías e instrumentos de evaluación que permiten conocer, a docentes y estudiantes, el nivel de desarrollo de las competencias que aborda la asignatura.

La evaluación ocurre:

A través de una instancia de evaluación individual y su instancia de recuperación se miden aspectos como la interpretación de consignas, resolución de problemas usando lógica de programación, integración de componentes, implementación de validaciones, conexión a servicios remotos y actualización de datos en una base de datos relacional.

A través de la realización de cuestionarios de autoevaluación donde los estudiantes pueden evaluar el grado de avance personal sobre los contenidos propuestos y dejar registro del mismo.

A través de la observación de las / los estudiantes en clase durante los debates.

A través del seguimiento del trabajo grupal, de las entregas pactadas y de los entregables planificados.

Los trabajos grupales y su documentación deben almacenarse en el repositorio git creado a tal efecto.

Serán considerados los siguientes aspectos al evaluar:

- Cumplimiento de plazos de entrega en evaluaciones.
- Dominio técnico del tema evaluado.
- Capacidad de integrar conceptos y aplicaciones.
- Dominio de terminología técnica.
- Expresión clara y técnicamente correcta.

A continuación, se detallan todos los Resultados de Aprendizajes con sus contenidos a desarrollar para alcanzarlos, la mediación pedagógica, metodologías y estrategias de evaluación, tiempo en horas reloj.

Resultados de Aprendizaje	Contenidos según programa	Mediación Pedagógica	Metodología y Estrategias de Evaluación	Tiempos en hora reloj
RA 1	Lenguaje Java, POO. Principales APIs. Tests Unitarios.	<p>Estrategias:</p> <p>Revisión guiada de materiales. Formación Experimental. Talleres dirigidos. Resolución de Ejercicios</p> <p>Actividades:</p> <p>Cuestionarios de autoevaluación como evidencia del avance y seguimiento de los contenidos. Resolución de Ejercicios de forma autónoma aplicando los conceptos vistos.</p>	<p>Instrumentos:</p> <p>Cuestionarios de Autoevaluación. Desafíos de Programación.</p> <p>Criterios:</p> <p>Utiliza correctamente los elementos del Lenguaje de Programación Java. Es capaz de escribir clases y las pruebas de las mismas. Entiende las ventajas y desventajas de los componentes predefinidos y los utiliza correctamente.</p>	<p>Horas teoría: 8</p> <p>Horas laboratorio: 16</p>

RA 2	Arquitectura de Microservicios Frameworks Herramientas de Modelado	Estrategias: Aprendizaje basado en investigación. Trabajo Autónomo. Seminario. (Debate y Oposición). Actividades: Producción audiovisual presentando unos de los frameworks disponibles. Análisis y valoración de las presentaciones de las presentaciones de los demás grupos.	Cuestionarios de Autoevaluación. Instrumentos: Presentación oral. Avance de TPI. Criterios: Conoce los diferentes Frameworks de Backend existentes y tiene una visión crítica acerca de las ventajas y desventajas de cada uno. Prepara el entorno de trabajo para el TPI con el Framework propuesto por la cátedra.	Horas teoría: 6 Horas laboratorio: 0
RA 3	Implementación y prueba de Servicios básicos. Estructura de contenedores y despliegue en entornos de prueba. Acceso a base de datos y ORM. Gateway de servicios, Ruteo e Intercepción. Contenedores.	Estrategias: Lección Magistral Participativa. Revisión guiada de materiales. Talleres dirigidos. Aprendizaje Basado en Proyectos aplicado a la implementación de un Práctico Integrador. Actividades: Construcción de modelos iniciales a partir de los Talleres propuestos y generando paulatinamente mayor nivel de autonomía. Construcción del Backend del Trabajo práctico integrador en	Instrumentos: Cuestionarios de Autoevaluación. Avance de TPI. Instancia de Evaluación Individual. Criterios: Es capaz de programar y probar los componentes de backend de una aplicación. Se conecta a una base de datos para consumir y registrar los datos necesarios.	Horas teoría: 14 Horas laboratorio: 16

		base a las consignas propuestas por la cátedra.	Despliega la aplicación de forma que pueda ser puesta a prueba.	
RA 4	Autenticación. Autorización. JWT Patrones de Seguridad	<p>Estrategias: Revisión guiada de Materiales. Taller dirigido. Implementación de entorno guiado.</p> <p>Actividades: Refactorización de la solución del TPI agregando Autenticación y Autorización.</p>	<p>Instrumentos: Cuestionarios de Autoevaluación. Entrega de TPI.</p> <p>Criterios: Conoce y es capaz de implementar un mecanismo de autenticación. Agrega capacidad de autorización y logra administrar el resultado exitoso y fallido de las peticiones.</p>	<p>Horas teoría: 14 Horas laboratorio: 12</p>
RA 5	Monitoreo. Centralización de Logs CI Despliegue	<p>Estrategias: Lección Magistral Participativa. Estudio de casos. Taller dirigido.</p> <p>Actividades: Comprensión y prueba de los elementos del taller para comprobar el funcionamiento de las herramientas propuestas.</p>	<p>Instrumento: Cuestionarios de Autoevaluación.</p> <p>Criterio: Comprende el problema de logs en las aplicaciones con más de un componente y conoce los elementos principales de las herramientas que resuelven el problema.</p>	<p>Horas teoría: 4 Horas laboratorio: 6</p>

14. Condiciones de aprobación

Al ser una asignatura electiva, se opta por seguir la recomendación del Departamento de Ingeniería en Sistemas de Información para las materias electivas, de forma tal que todo alumno que cumple las condiciones de Regular en la materia alcanza también la condición de Aprobación Directa.

Por esto, a continuación, vamos a plantear las condiciones de Regularidad/Aprobación Directa de la materia:

- a) Obtener 6 o más como Nota de Seguimiento, esta estará compuesta por los cuestionarios de autoevaluación, desafíos de programación, exposición/generación de material audiovisual y participación en foros de discusión, la nota se calculará como un promedio ponderado de las instancias mencionadas
- b) Obtener 6 o más como Nota en la instancia de evaluación Individual o la instancia de recuperación
- c) Obtener 6 o más en el Trabajo Práctico Integrador grupal o su reentrega

El estudiante debe cumplir con todas las entregas de las instancias de evaluación propuesta, a excepción de los cuestionarios de autoevaluación, todas las demás evaluaciones deben ser entregadas.

Finalmente se transcribe la escala de notas propuesta por el Departamento de Ingeniería en Sistemas de Información:

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: Se utiliza la escala de exámenes finales ya que como se expresó anteriormente todo estudiante que regulariza la materia obtiene en el mismo acto la aprobación directa de la misma.

15. Modalidad de examen

Al plantearse como una materia donde todo aquel que regulariza, le corresponde la aprobación directa, de acuerdo con la recomendación del Departamento de Ingeniería en Sistemas de Información, no habrá evaluación de tipo examen final sino solo transcribir la nota final de los que regularizaron/aprobaron la asignatura a las actas, en base a las notas obtenidas en las actividades de cursado y firmar libretas.

A los efectos de completar la documentación y afirmar las condiciones transcribo las condiciones de regularización / aprobación del apartado anterior, que finalmente son las instancias a cumplir por los alumnos para aprobar la asignatura.

A continuación enumeramos las condiciones de Regularización/Aprobación de la Asignatura:

- a) Nota de Seguimiento: Obtener 6 o más como Nota de Seguimiento, esta estará compuesta por los cuestionarios de autoevaluación, desafíos de programación, exposición/generación de material audiovisual y participación en foros de discusión, la nota se calculará como un promedio ponderado de las instancias mencionadas.
- b) Evaluación Individual: Obtener 6 o más como Nota en la instancia de evaluación individual o su instancia de recuperación.
- c) Trabajo Práctico Integrador Grupal: Obtener 6 o más en el trabajo práctico integrador grupal o su reentrega

El estudiante debe cumplir con todas las entregas de las instancias de evaluación propuesta, a excepción de los cuestionarios de autoevaluación, todas las demás evaluaciones deben ser entregadas.

Como nota final de la asignatura se consignará el promedio directo de estas 3 notas.

16. Recursos necesarios

Las clases se llevan a cabo en laboratorio, donde se requiere:

- IDE IntelliJ Idea Community Edition y Visual Studio Code
- Java JDK 17
- Maven
- Capacidad de Acceso a Terminal
- Postman
- Curl
- Gitlab

- Docker (En caso de equipos Windows, se requiere WSL2 instalado y configurado con alguna distribución que soporte APT y Snap)
- Guía de trabajo: Publicada en gitlab
- Proyector de alta definición.

Nota: Se precisa el mismo software para el desarrollo del trabajo grupal.

Anexo I: Plantel docente de la asignatura			
Titular	Felipe Steffolani	Dedicación:	Especifique la cantidad de dedicaciones.
Asociado	Especifique Nombre y Apellido completo.	Dedicación:	Especifique la cantidad de dedicaciones.
Adjunto:	Diego Serrano Federico Bett Jorge Harach Ruben Romero	Dedicación:	Especifique la cantidad de dedicaciones.
Jefe de Trabajos Prácticos	German Romani Ivo Parun Rua	Dedicación:	Especifique la cantidad de dedicaciones.
Auxiliar de 1ra.	Rodrigo Díaz Mc Williams Leonel Bendezú	Dedicación:	Especifique la cantidad de dedicaciones.
Auxiliar de 2da.	Franco Pesci	Dedicación:	Especifique la cantidad de dedicaciones.

FIRMA (Jefe o encargado de cátedra).

Anexo II: Cronograma de clases/trabajos prácticos/evaluaciones (por comisión)

COMISIÓN: Indique la comisión.

Nro. de Semana	Fecha	Tema	Tipo de Actividad
1	Indique la fecha	Describe el tema trabajado	Seleccione el tipo de actividad.

FIRMA (de cada docente que conforman la comisión).