

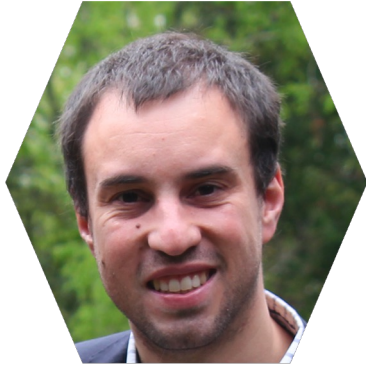


PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración

Profesores



Arturo Tagle



Daniel Darritchon

Ayudantes



**Martín
Illanes**



**Teresita
López**



**Francesco
Rodríguez**

Arturo Tagle

Ingeniero Civil Industrial,
mención Tecnologías de
Información.

Magíster en Ciencias de la
Computación.



Experiencia

CTO en Mineral Forecast, IA para
descubrimiento de minerales.

Director de Tecnología en Betterfly, el primer
unicornio social de latinoamérica.

Socio y fundador de Kunder, empresa de
desarrollo a medida y creación de soluciones
digitales para la industria financiera.

Director y asesor tecnológico de empresas
de la industria fintech: Racional, Heypay,
Nesto, Percus.

Profesor Taller de Integración desde 2013.



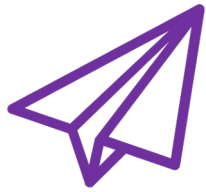


**KEEP
CALM
IT'S
ACTIVITY
TIME!!!**

Conociendo a tus
compañeros

Programa del curso

El programa completo del curso [se subirá a Canvas](#), sección “Programa del curso”.



Objetivos del curso

Al finalizar el curso, el alumno será capaz de:

1. Entender diferentes tecnologías y técnicas de integración según las tecnologías y técnicas actuales.
2. Describir las dependencias entre sistemas y sus interacciones.
3. Implementar capas de software que permitan la interoperabilidad de sistemas de información o herramientas, en su totalidad o en parte.
4. Discutir sobre las consecuencias tecnológicas y de negocios al integrar sistemas, considerando las variables económicas, tecnológicas y legales asociadas.
5. Diseñar estrategias para la interoperación de sistemas de información con un objetivo específico.
6. Analizar los costos, potenciales beneficios y potenciales problemas que implica la integración de sistemas de información y herramientas.
7. Identificar problemáticas relacionadas a la falta de integración de sistemas, así como los desafíos para lograr una correcta integración.

Este curso busca mostrar diferentes soluciones y mitigaciones para los problemas originados por la falta de integración de sistemas en las organizaciones. Además, el curso enseña diferentes técnicas de integración de sistemas ofrecidas por la tecnología vigente.

Durante el desarrollo del curso, se buscará potenciar el pensamiento de los alumnos para que sean capaces de reconocer beneficios, limitaciones y consecuencias de diferentes soluciones posibles a los problemas relacionados con la integración de sistemas.



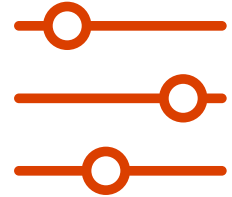
Aprendizajes
esperados



Contenidos

1. Introducción a la integración
 - a. Técnicas, conceptos y tecnologías de integración
2. Integración por servicios
 - a. Web: Web API, Rest API, Webhooks, otros
 - b. SOAP
3. Integración por eventos
 - a. Características de sistemas de eventos
 - b. Patrones de manejo de eventos
4. Patrones de arquitectura de integración
 - a. Arquitecturas orientadas a integración por datos
 - b. Arquitecturas orientadas a servicios (SOA)
 - c. Arquitecturas orientadas a eventos
5. *Systems thinking*
 - a. Principios de diseño de sistemas robustos
 - b. Seguridad en la integración de sistemas
 - c. Principios de escalabilidad de sistemas
6. **Módulo transversal:** Integración de sistemas en las organizaciones: Experiencias de invitados, casos de estudio y aplicación práctica
 - a. Casos de estudio
 - b. Implicancias, complejidad y consecuencias de la integración
 - c. Experiencia práctica en organizaciones (presentaciones realizadas por invitados)
 - d. Proyectos de integración: aplicaciones prácticas

1. Proyecto grupal
2. Tareas
3. Actividades en clases
4. Examen



Evaluaciones

	Ponderación	Requisitos de aprobación	Otros
Promedio actividades en clase	10%		Se borra la peor nota
Promedio tareas	35%	≥ 4.0	3 tareas en el semestre
Nota proyecto	35%	≥ 4.0	Condiciones adicionales según evaluación de pares
Examen	20%		



Fechas

- [PENDIENTE] Tareas
- [PENDIENTE] Proyecto
- Examen:
 - 6 de diciembre, 17:30

Política de Integridad Académica (extracto)

Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho individualmente por el/la estudiante, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Política de Integridad Académica (completa)

Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho individualmente por el/la estudiante, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse como el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente.

Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (<https://registrosacademicos.uc.cl/reglamentos/estudiantiles/>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

Compromiso del Código de Honor

Este curso suscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso que exista colaboración permitida con otros/as estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es un debe conocer el Código de Honor (<https://www.uc.cl/codigo-de-honor/>)

Medios de comunicación



Clases



**Sitio web
del curso
(CANVAS)**



Correo Electrónico

Arturo Tagle

ajtagle@uc.cl

Daniel Darritchon

dedarritchon@uc.cl



**Atención a
alumnos**

Pedir hora en

[calendario](#)



Foros del curso

Disponibles en

Canvas

Recomendaciones para tener buenas clases

Participa en clases

La clase es más entretenida y dinámica si todos comentan y hacen preguntas.

Feedback continuo

Si alguna actividad resulta muy bien o no tiene buen resultado, háganme los comentarios, para repetir actividades similares o sacarlas del repertorio.

Respeto

Todas las opiniones y comentarios son válidos.

Expectativas

Qué se puede esperar de este curso

Qué no esperar de este curso

BREAK



Descanso de 10 minutos y volvemos.

Integración de sistemas

Introducción





¿Qué es un Sistema?



¿Qué es un Sistema?

Entidades o componentes interrelacionados que permiten una nueva función.

Un sistema hace algo que no puede hacer un componente por sí solo.



¿Qué es un Sistema?

Función del sistema:

Lo que el sistema hace.

Forma del sistema:

Lo que el sistema es.

Múltiples formas pueden realizar una misma función



Integración de sistemas

DEFINICIÓN

El proceso de crear un sistema de información complejo que puede incluir el diseño y/o construcción de una arquitectura o aplicación personalizada, integrándola con hardware, software o capas de comunicación existentes o nuevas.

Gartner, disponible en

<https://www.gartner.com/en/information-technology/glossary/system-integration>

Imaginemos una cervecería

Desafío:

Una cervecería tiene varios sistemas: CRM para manejar datos de clientes, sistemas de facturación, gestión de la producción, logística, etc. Sin embargo, estos sistemas no están integrados, resultando en silos de información e ineficiencias.

El sueño:

Esta cervecería quiere mejorar sus sistemas para mejorar la atención a clientes, tener información y analítica en línea, reducir costos y optimizar procesos.

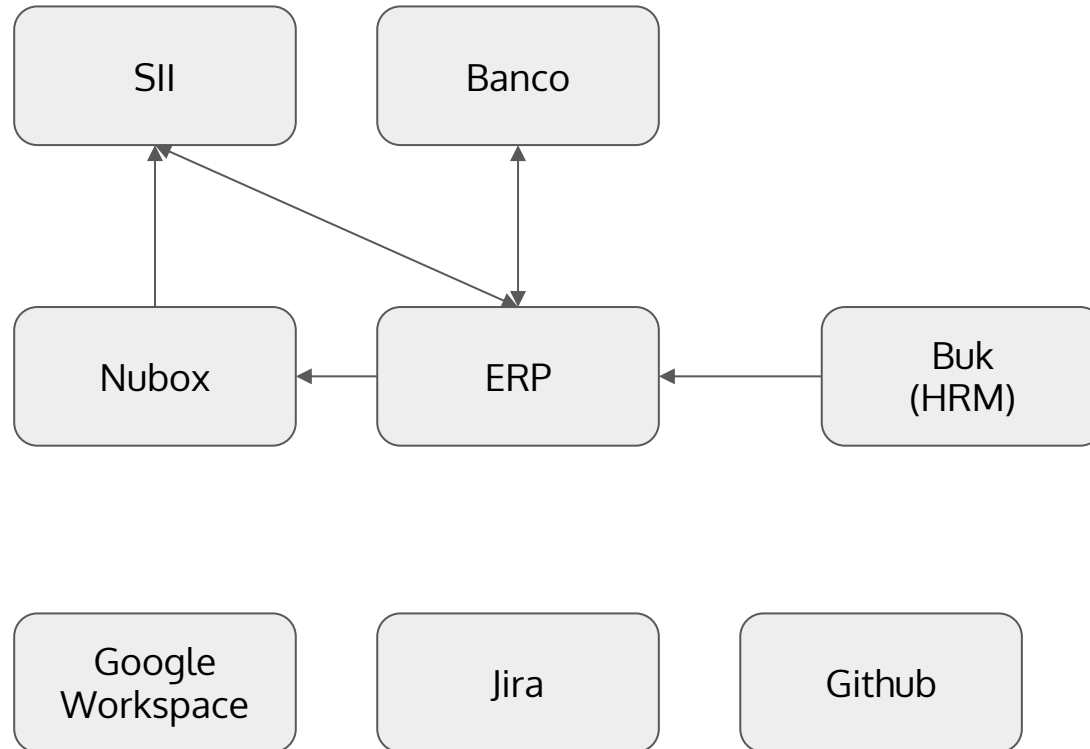
**¿POR QUÉ
QUEREMOS
INTEGRAR
SISTEMAS?**



Algunas razones

- Procesos transparentes para la comunicación e intercambio de datos entre sistemas
- Interoperabilidad de sistemas (dependencias)
- Automatización de procesos
- Disponibilidad de información
- Uso más eficiente del software
- Comunicación de sistemas nuevos con sistemas existentes

Stack tecnológico Kunder - flujo de información



Más razones para no crear un nuevo sistema con ambas funcionalidades

- Muy caro reprogramar todo para hacer un nuevo sistema
 - Además que conlleva el riesgo que el nuevo sistema no funcione.
- Es ineficiente reinventar la rueda
 - Sistemas actuales resuelven el problema en forma satisfactoria.
- Costo de migración es muy alto
 - Y la retrocompatibilidad es siempre necesaria.
- Velocidad de desarrollo < Velocidad del negocio
 - El negocio pide características más rápido de lo que lleva desarrollarlas de cero
 - Cuando el software está listo ya no se necesita

¿Se puede crear una
empresa dedicada a la
integración?



Dos opciones

Soluciones de integración

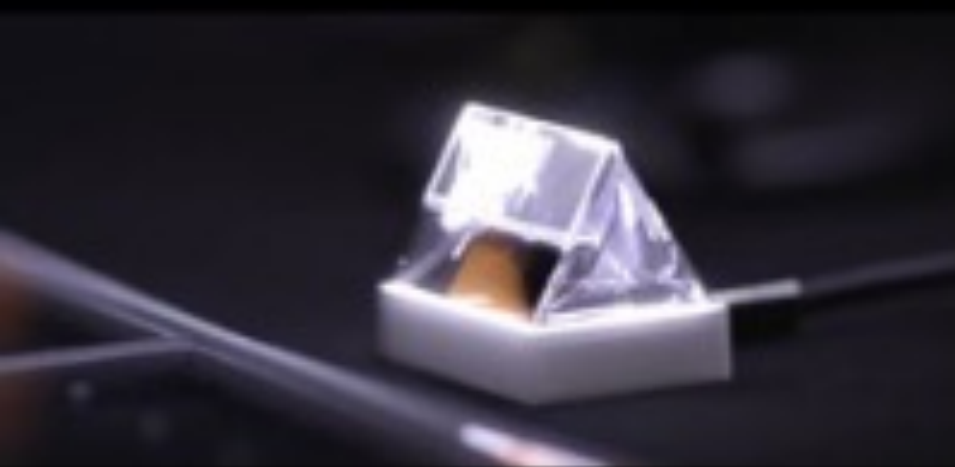
Generar herramientas que faciliten la integración entre sistemas.



Integración como modelo de negocio

Integraciones como medio para generar un nuevo modelo de negocios.





Diferentes formas de integrar sistemas

Algunas estrategias



Transferencia de archivos

Un sistema envía datos a una segunda que lee el archivo generado.

Los sistemas deben “ponerse de acuerdo” en el nombre del archivo, ubicación, formato, tiempos de escritura y lectura, y la delegación de responsabilidad sobre quién deberá borrarlo (si fuera necesario).

Ejemplos

- Descarga de un archivo de un sistema para cargar en un segundo sistema
- Envío de archivos por correo electrónico
- Transferencia por FTP/sFTP.

Base de datos compartida

Múltiples sistemas comparten una base de datos común.

Como todos los datos se encuentran en el mismo lugar, no se debe transferir datos entre aplicaciones.

Desventajas

La mayor dificultad es encontrar un modelo de datos que sea útil para todas los sistemas.

La base de datos se convierte en un punto único de falla y en un cuello de botella cuando los sistemas escalan.

RPC

Remote procedure call

Un sistema expone su funcionalidad (funciones del código) para que pueda ser accedido remotamente por otro sistema mediante un procedimiento remoto.

La comunicación es en tiempo real y síncrona.

Ejemplos:

- Comunicación entre apps móviles

Mensajería

Un sistema publica un mensaje a otro mediante un canal común de comunicación. El segundo sistema responde “cuando puede” el mensaje y procesa el contenido. Esto puede o no generar una respuesta.

Los sistemas deben ponerse de acuerdo en el canal y formato del mensaje.

Ejemplos

- API Rest
- GraphQL

2

Soluciones posibles al “problema de la integración”

Cambiar sistemas actuales por una suite “completamente integrada”.

Un sólo sistema que ofrece todas las funcionalidades requeridas.

Integrar sistemas actuales para hacerlos interoperar.

Las funcionalidades requeridas se logran por el uso de todos los sistemas (existentes o nuevos).

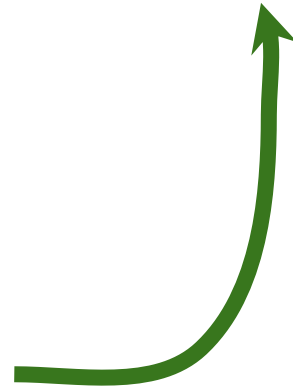
Cambiar sistemas actuales por una suite “completamente integrada”.

Un sólo sistema que ofrece todas las funcionalidades requeridas.

¡OBJETIVO DEL CURSO!

Integrar sistemas actuales para hacerlos interoperar.

Las funcionalidades requeridas se logran por el uso de todos los sistemas (existentes o nuevos).



Que significa “comprar un software integrado”?

Ejemplo de SAP Business One



0:04 / 1:19



What is SAP Business One?



SAP Business One
22.9 k suscriptores

Suscribirse

209



Compartir


Descargar

Guardar



Desafíos que nos encontraremos al integrar sistemas





Toda integración de sistemas se verá
enfrentada a alguno de estos problemas:

1. Todos los sistemas son diferentes
2. El cambio es inevitable
3. Redes son poco confiables
4. Redes son lentas

Todos los sistemas son diferentes



La integración entre dos sistemas deberá solucionar el hecho que probablemente ambos sistemas estén escritos en lenguajes diferentes, utilizarán distintos sistemas operativos e incluso diferentes arquitecturas.

Una solución de integración debe ser capaz de interactuar con todo tipo de sistemas.

El cambio es inevitable



Los sistemas no son estáticos, cambian con el tiempo.

Una solución de integración debe ser capaz de adaptarse a los cambios y mantener el ritmo de los sistemas que interconecta. Si un sistema cambia, todos los otros se verán afectados.

Se debe buscar minimizar las dependencias de forma de tener siempre un bajo acoplamiento.

Redes poco confiables



Normalmente, las soluciones de integración deben transportar datos desde un sistema a otro a través de una red.

Un sistema distribuido, o sistemas integrados a través de una red, se ven enfrentados con muchos más problemas que cuando todo corre en una misma máquina.

Redes son lentas



Enviar datos a través de una red es varios órdenes de magnitud que consultar datos localmente.

Toda integración que utilice una red deberá enfrentar un cambio en el *performance* de la solución.



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración



Bibliografía

Desafíos y formas de integración

Hohpe, G. (2002, July). Enterprise integration patterns. In 9th conference on pattern language of programs (pp. 1-9).

Hohpe, G., Woolf, B. (2015). Introduction. In *Enterprise integration patterns: Designing, building and deploying messaging solutions* (pp. xxix - li). Boston, Massachusetts: Addison-Wesley.



Artículos relacionados

Desafíos y formas de integración - REST vs RPC

Nally, M. (2018, November 15). *REST vs. RPC: What problems are you trying to solve with your apis?*
Retrieved March 15, 2021, from <https://cloud.google.com/blog/products/application-development/rest-vs-rpc-what-problems-are-you-trying-to-solve-with-your-apis>