



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración

Profesores

Arturo Tagle / Daniel Darritchon

Profesores



Arturo Tagle



Daniel Darritchon

Ayudantes



**Martín
Illanes**



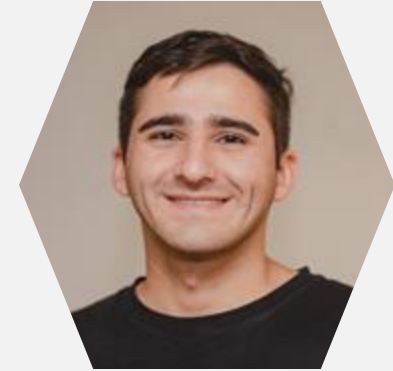
**Teresita
López**



**Francesco
Rodríguez**

Daniel Darritchon

Ingeniero Civil Industrial,
mención Tecnologías de
Información.



Experiencia

Ayudante IIC1103: 2019-2

Ayudante IIC3103: 2020-1, 2021-1, 2022-1

Profesor IIC3103: desde 2022-2

Software Engineer @ adere.so

- Nov 2020 -> Sept 2024

Software Engineer II @ front.com

- Sept 2024 -> ongoing

dedarri.com



129

Instagram

12/09/23, 15:49

08/05/24, 14:07

Xime Huidobro

ximehuidobro

adereso...

: iHasta pronto!

✓

Reabrir

825433

CD

17/04/24, 11:44

17/04/24, 11:44

Discovery-A

discovery_a_scouting

adereso...

: UNHANDLED_ERROR

✓

Designorar

826184

Sin ejecutivo

12/04/24, 14:55

12/04/24, 14:58

Daniel Darritchon

dedarritc

adereso...

: Hola Con saltos de linea

✓

Reabrir

826173

DD

12/04/24, 14:45

12/04/24, 14:47

Xime Huidobro

ximehuidobro

adereso...

: Hola 1. primero 2. punto 2 3. hola

✓

Designorar

826170

DS

20/03/24, 17:50

12/04/24, 14:35

Daniel Darritchon

dedarritc

Daniel

: Holaaa

999+

Email

27/06/24, 14:38

25/07/24, 12:29

FP

Francisco Pinto

fpinto@adere.so

hola

"Twitter": te contacto

Reabrir

826307

FP

10/06/24, 13:15

10/06/24, 13:15

V

"Twitter" via Demo Post...

demo@adere.so

New feedback this week for Adereso

"Twitter": 1 new post

Ignorar

826294

V

11/11/19, 03:11

15/05/24, 13:51

PS

Postcenter Soporte

soporte@postcenter.io

[Postcenter] - Notificación Canal

Postce... : |||| soporte@postcenter.io --- --- Estim...

Designorar

810354

DS

Sin ejecutivo

11/11/19, 03:08

15/05/24, 13:51

PS

Postcenter Soporte

soporte@postcenter.io

[Postcenter] - Notificación Canal

Postce... : |||| soporte@postcenter.io --- --- Estim...

Designorar

810353

DS

Sin ejecutivo

11/11/19, 03:01

15/05/24, 13:51

PS

Postcenter Soporte

soporte@postcenter.io

[Postcenter] - Notificación Canal

clientes

+150

Cliente más grande



100 usuarios
usando la app
(daily)



Administrar integraciones

[Mira la ayuda para esta sección](#)

Integraciones habilitadas

**BotCenter**

Inactivo

Integra chatbots para mejorar tu atención al cliente.

**HubSpot**

Activo

Obtén y envía información de metadatos entre HubSpot y Desk.

**Shopify**

Activo

Visualiza las órdenes y contacta con los clientes de tu tienda de Shopify.

Tiendas vinculadas: `darritchon-travel-co.myshopify.com`**Webhook de notificaciones**

Activo

Recibe en tus sistemas una notificación que contiene el ID del ticket, basado en las reglas que definas.



Integraciones disponibles

**Salesforce**

Obtén y envía información de metadatos entre Salesforce y Desk.

3 integraciones

Integraciones

**Shopify**

Conecta con tu e-commerce

**HubSpot**

Conecta a tu CRM


**Salesforce**


Conecta a tu CRM









Compose


- Me ▾
-  My Inbox


 Assigned to me


 Shared with me
- Team Inboxes ▾
-  Priority Accounts



 orders@acme.com

 SMS

 Live chat

 Twitter

 WhatsApp

 NPS Surveys
- Teammates ▾
-  Marisa Jones

- Mia Cartwright

Analytics question

Good afternoon, I'd like to...
- Kaitlin Lamas

New account setup

Hello there! My wife and I...
- Sabrina Calhoun

Shipment status

Hello, I'm checking in on th...
- Cal Weaver

Bulk fulfillment help

My team needs options for...
- Tim Riggins



New quote request

Hi! Thanks for your help wi...
- Carly Scott


Issue with tracking

Good morning, I'm unable...

INBOX



Charlie Reyes

 Mia Cartwright

Subject: Analytics question


⋮

↩


Good afternoon,

I'd like to explore the analytics and API capabilities to do an ROI analysis, can you please send along any relevant resources?

Best,
Mia


 @charlie

can you take care of this?

 @marisa

I'm on it!

Shared with 2 participants


 Charlie Reyes

Subject: Analytics question





⋮

↩

Hi Mia,



Add comment...



- Conversations
- Last 7 days
- Front Chat - Support T1

Updating billing

● support@ - Support T1

Can't upload customs forms

● Facebook - Comments

It's about time we got a pr...

● WhatsApp - Fullfillment

Can you share an update o...

● Front Chat - Support T1





Where do I go for monthly...

● support@ - Support T1

The form keeps making me...

● Front Chat - Sales

Request for pricing quote
- 31



+

clientes

+8500

Cliente más grande

?

10.000 usuarios
usando la app
(daily)

Manage all of your favorite apps, right within Front



+100 integraciones

FILTERS

Explore -

- ☒ All
- ☐ Popular
- ☐ New
- ☐ Channels

Categories +

Plans +

🔍 Search for an app...

All apps



AMGiNE

Bots & Workflow Automation



Aide

Bots & Workflow Automation



Aircall

Voice & Telephony

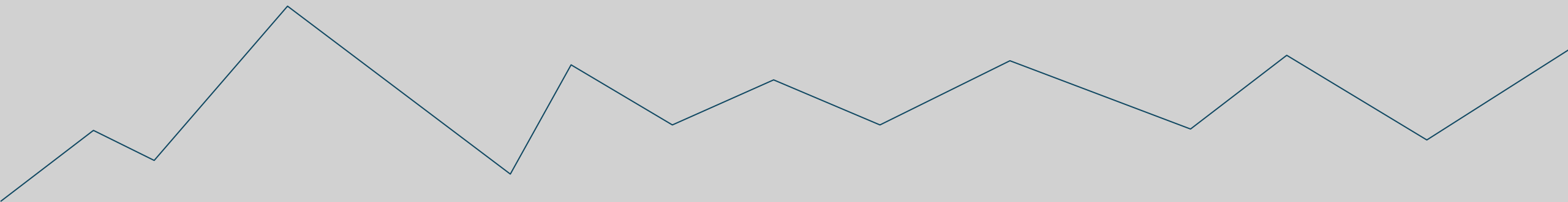




Integración por servicios

Módulo 2 - Parte 1

¿Qué es un servicio?



DEFINICIÓN

El término servicio se refiere a un set de funciones de software relacionadas entre sí que pueden ser reutilizadas para diferentes propósitos.

El acceso se realiza a través de una interfaz establecida (canal de comunicación) bajo reglas de acceso y de uso.



Ejemplo 1

Consumiendo un
servicio





¿Quiénes son los principales actores en un servicio?

Características de los servicios

1. Bajo acoplamiento

2. Comunicación punto a punto

3. Comunicación iniciada por el cliente

4. Comunicación síncrona

5. Dos mensajes: solicitud y respuesta

Bajo acoplamiento

Permite hacer modificaciones del funcionamiento interno del servicio:

- Los sistemas consumidores no se ven impactados.
- Los parámetros y respuestas del servicio son los mismos.

Permite hacer modificaciones en el proveedor del servicio

- Se puede cambiar al proveedor del servicio en su totalidad si los contratos de comunicación no cambian.

Bajo acoplamiento

Desacoplamiento tecnológico

- El servicio publicado puede comunicarse con un cliente con cualquier tecnología, y viceversa. No importa la tecnología en que está construido el servicio para poder comunicarse con él.
- El servidor es una “caja negra” para el cliente. El cliente sólo espera una respuesta según el contrato (documentación).

La comunicación es punto a punto

Sólo hay dos actores:

- Servidor: proveedor del servicio
- Cliente: consumidor del servicio

La comunicación es directa entre estos dos actores.

La comunicación es iniciada por el cliente

La comunicación se inicia cuando el cliente solicita información (realiza un *request*).

La comunicación termina cuando el servidor ha terminado de entregar la información solicitada.

El servidor **nunca** inicia la comunicación. Está esperando a recibir una solicitud por parte de un cliente.

La comunicación es síncrona

El cliente realiza una solicitud y espera hasta obtener una respuesta o detener la espera por *timeout*.

El cliente queda en un estado de “*busy waiting*” a la espera de la respuesta del proveedor del servicio.

**La
comunicación
sólo tiene dos
mensajes:
solicitud y
respuesta.**

El cliente realiza una solicitud (*request*) y el servidor entrega una respuesta (*response*).

La comunicación finaliza cuando el proveedor del servicio envía la respuesta al cliente. No hay más mensajes, hasta que haya una nueva solicitud.

Ejemplo 2

Poniendo en práctica
las características

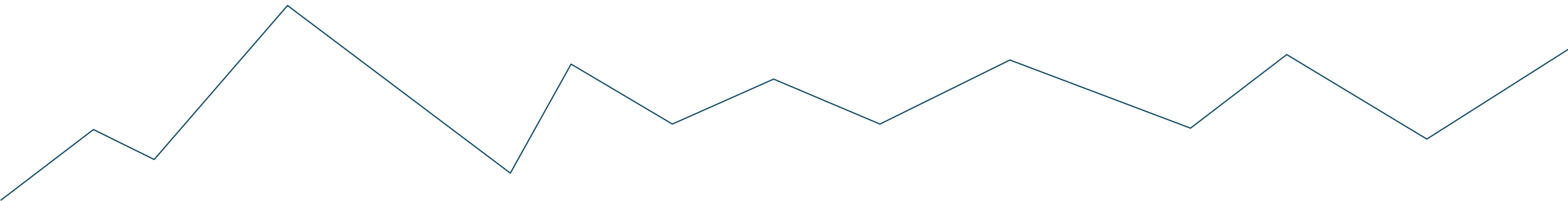




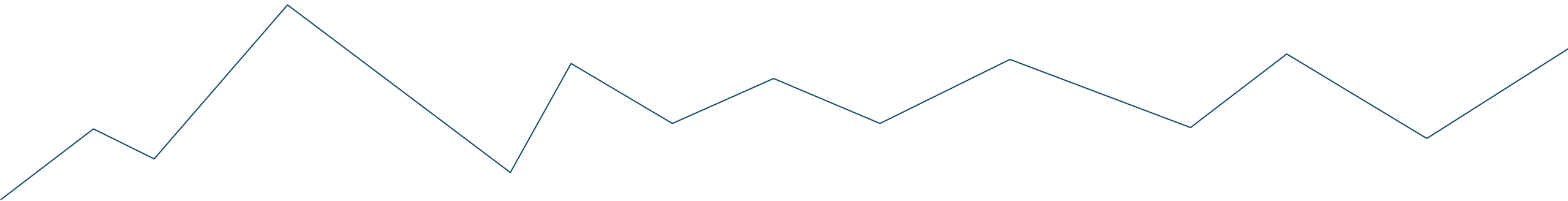
¿Por qué queremos
usar servicios?

Un servicio provee un medio para exponer funcionalidad a clientes

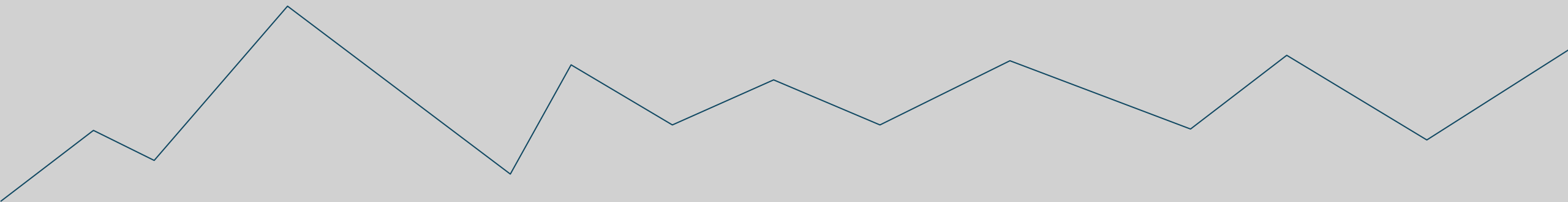
En aplicaciones más grandes, se busca también proveer acceso a otros desarrolladores a la plataforma mediante el uso de servicios



Desde otro punto de vista, los servicios nos ayudan a descomponer un sistema en muchas unidades lógicas diferentes, que interactúan entre ellas para producir un resultado.



Servicios web



Servicios: línea de tiempo



Servicios: línea de tiempo



SERVICIOS WEB Y REST

Servicios web

Usados en ambientes/sistemas web

Son independientes del lenguaje en que se implemente

La capa de transporte siempre es sobre HTTP

REST

Primera implementación en el año 2000

Principalmente basados en JSON



CONCEPTOS BÁSICOS

- Servidor: Quién entrega el servicio. Puede ser implementando de miles de maneras diferentes: Framework MVC, Serverless, etc.
- Cliente: Página web, aplicación móvil, otro sistema, etc.
- Protocolo de comunicación: HTTP
 - HTTP es una serie de guías que describen cómo se comunican los datos, validaciones, formatos y otros
 - La ventaja de usar HTTP es que ya tiene todas las bases para implementar cosas sobre este

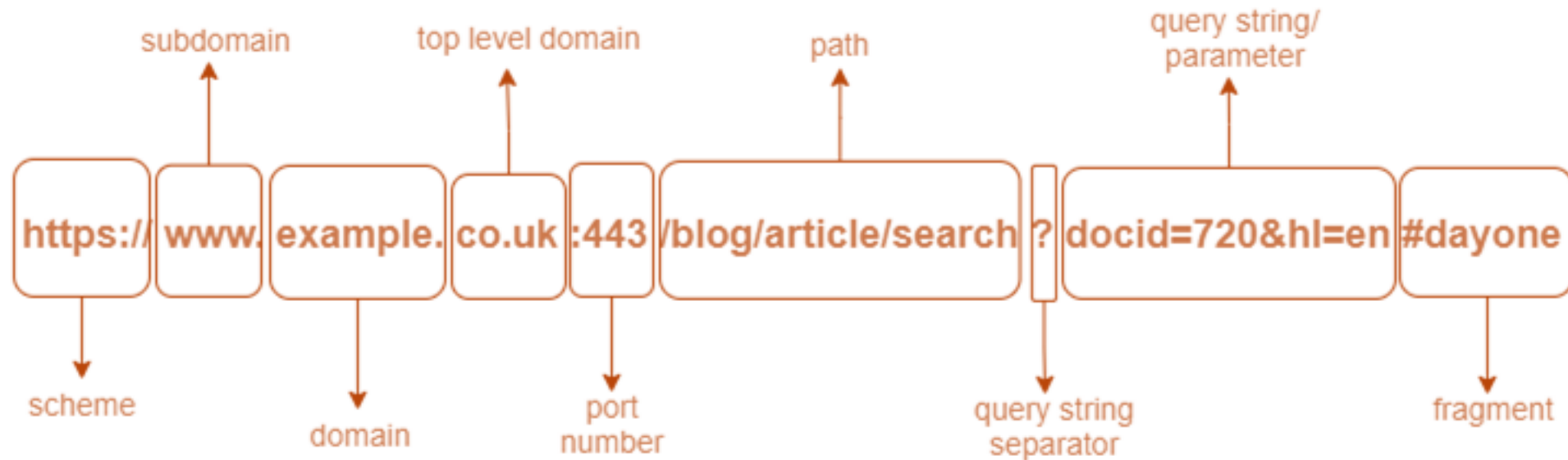




¿Cómo invoco un
servicio en la web?

¿Cómo se construye una URL?

URL : <https://www.example.co.uk:443/blog/article/search?docid=720&hl=en#dayone>



Métodos HTTP

Un servicio debe ser llamado mediante una IP o URL con un método o verbo para diferenciar distintas acciones que se pueden realizar con un mismo servicio.

Métodos principales

- GET
- DELETE
- POST
- PUT
- PATCH

Otros métodos menos utilizados

- HEAD, OPTIONS, TRACE, MOVE, MERGE, UPDATE, LABEL, PROPFIND y otros

Respuesta de servicios

La respuesta de un servicio contiene un código de estatus, encabezados, y el cuerpo de la respuesta.

Códigos de estatus

- 2XX: éxito en la operación. 200 es el más común
- 3XX: redirección temporal o permanente (el recurso ya no se encuentra en este lugar)
- 4XX: Error en la llamada. Típicamente 400 (*Bad Request*) o 404 (*Not Found*)
- 5XX: Error de servidor

Encabezados (*headers*)

Corresponden a metadata del mensaje, y no la carga del mensaje en sí.

Agregan información adicional al *request* y/o *response*, tales como:

- *Request*: Credenciales de autorización, tipo del contenido del mensaje, control de caché, etc.
- *Response*: Cantidad de resultados, tipo del contenido del mensaje, cookies, control de caché, settings, etc.

Mensajes (*body*)

El cuerpo del mensaje es donde se encuentra el contenido que se está enviando y recibiendo.

GET y DELETE no incluyen cuerpos de mensaje.

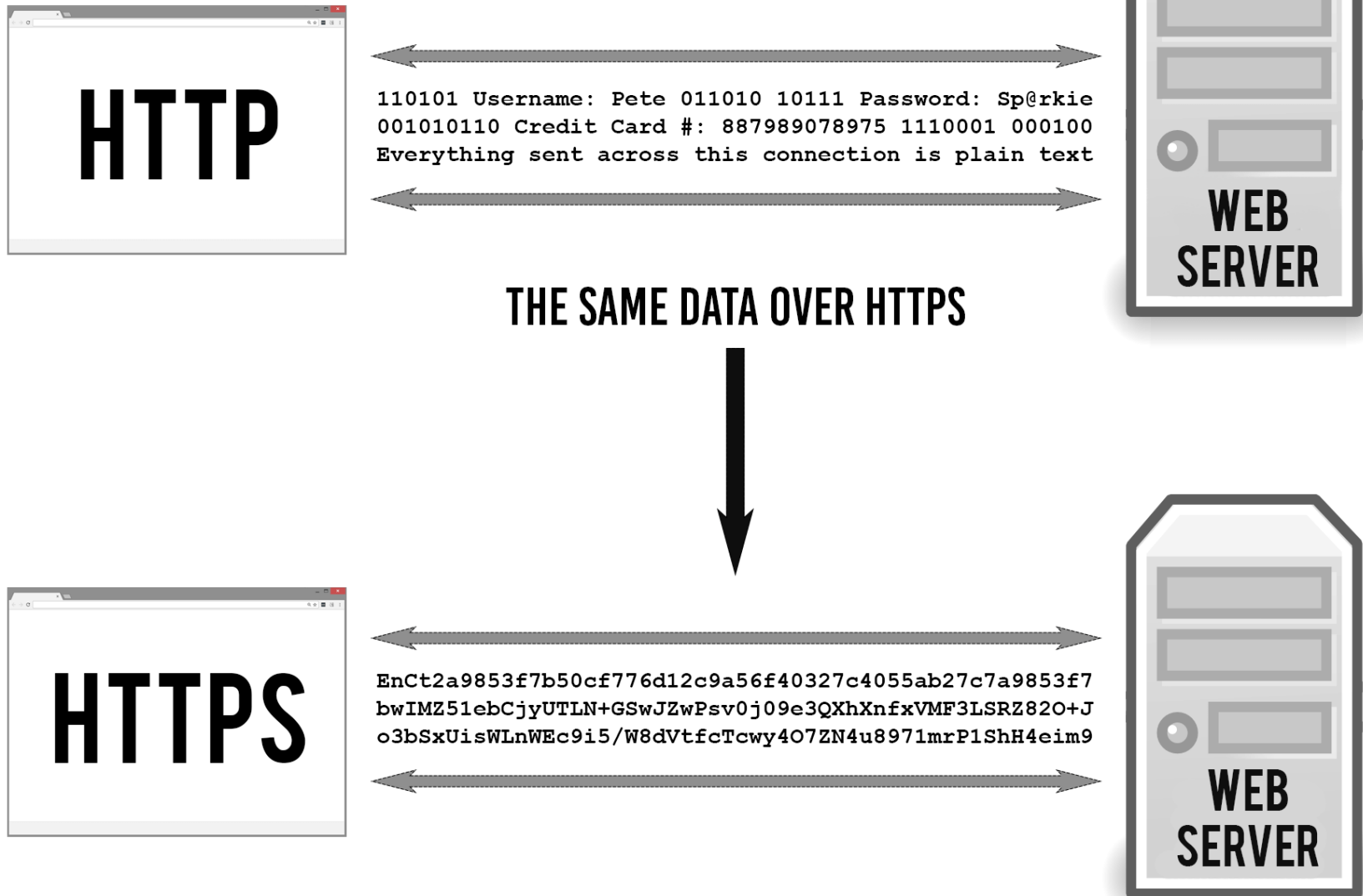
La respuesta siempre tiene mensajes completos: *headers* y *body*. El *body* puede ser vacío.

El cuerpo puede tener cualquier formato: texto plano, HTML, JSON, XML, CSV, etc. Normalmente un *header* determina el tipo de contenido.

HTTP(s)

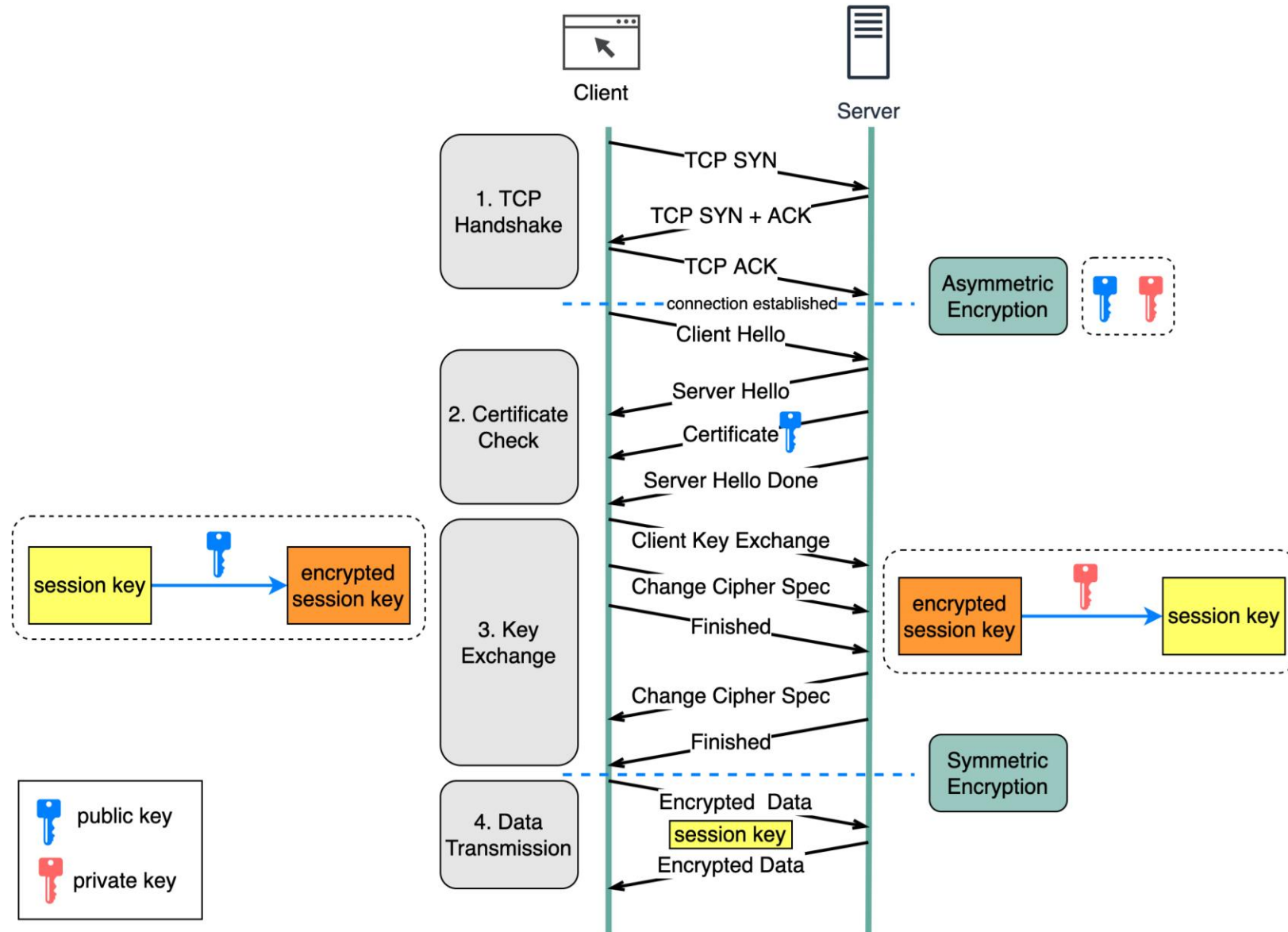
Hypertext Transfer Protocol (secure)

HTTP vs HTTPS



Cómo funciona HTTPs?

- Basado en **TCP**
- Servidor comparte su **llave pública**
- Cliente genera **secreto** y la envía al servidor encriptada con su **llave pública**
- Servidor desencripta el **secreto** usando su **llave privada**
- Ahora ambos comparten un **secreto** con el que pueden encriptar/desencriptar los siguientes mensajes



Ejemplo 3

Construyendo un
servicio web





PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

IIC3103

Taller de Integración

Profesores

Arturo Tagle / Daniel Darritchon

BIBLIOGRAFÍA

Service-oriented architectures

Microsoft. (2010, July 14). *Service-Oriented integration*. Retrieved March 17, 2024, from [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff647687\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff647687(v=pandp.10))

Ilustraciones

Diseñadas por freepik.es



BIBLIOGRAFÍA

Web services

The evolution of web services. Tech Machina. (2007, August 14).

<http://www.techmachina.com/2007/08/evolution-of-web-services.html>

Skrew Everything. (2019, February 10). *Http server: Everything you need to know to build a simple http server from scratch.* Medium.

<https://medium.com/from-the-scratch/http-server-what-do-you-need-to-know-to-build-a-simple-http-server-from-scratch-d1ef8945e4fa>

Mozilla. (2021, June 1). *What is a URL? - learn web development: MDN.*

Learn web development. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL



BIBLIOGRAFÍA

Web services

Rest worst practices, <https://jacobian.org/writing/rest-worst-practices/>

Is REST better than SOAP? Yes, in Some Use Cases,

<https://nordicapis.com/rest-better-than-soap-yes-use-cases/>

Comparing API Architectural Styles: SOAP vs REST vs GraphQL vs RPC,

<https://levelup.gitconnected.com/comparing-api-architectural-styles-soap-vs-rest-vs-graphql-vs-rpc-84a3720adefa>

geeksforgeeks.org. (2021, June 29). *Components of a URL.*

<https://www.geeksforgeeks.org/components-of-a-url/>

