

Programación Avanzada

IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha



Anuncios

23 de octubre de 2025



1. Hoy tendremos la Experiencia 4.
2. Recuerden que la Tarea 3 se entrega la próxima semana.
3. Respondan la ECA.
4. Hicimos unos cambios en la planificación de las últimas 2 semanas de clase.

Serialización y JSON



Serialización y JSON

Pensando en la serialización de forma general:

- ¿Por qué es necesario serializar?
- Cuando escribimos un archivo, ¿se serializa lo que escribimos?

Pensa específicamente en JSON:

- ¿Es *human-readable*?
- ¿Qué tipos de datos acepta?
- Si queremos serializar algo que no es aceptado por JSON, ¿qué debemos hacer?

Networking

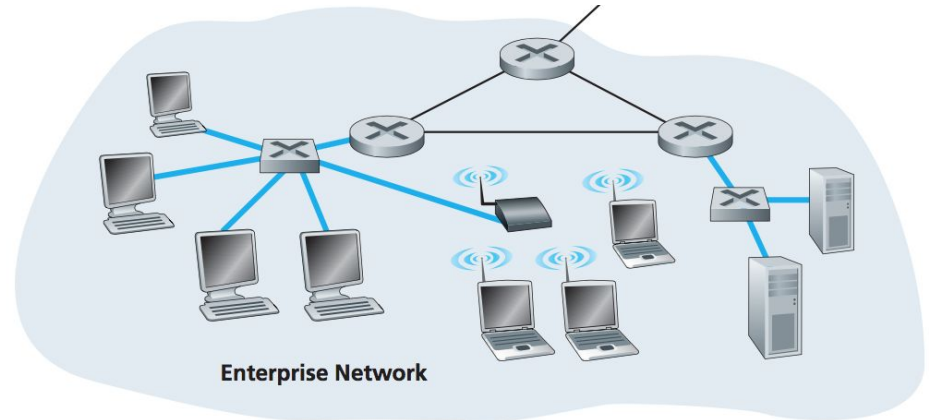


Networking

En la arquitectura cliente-servidor, ¿cuál es el rol del cliente?

¿Para qué se usa la dirección IP y el puerto?

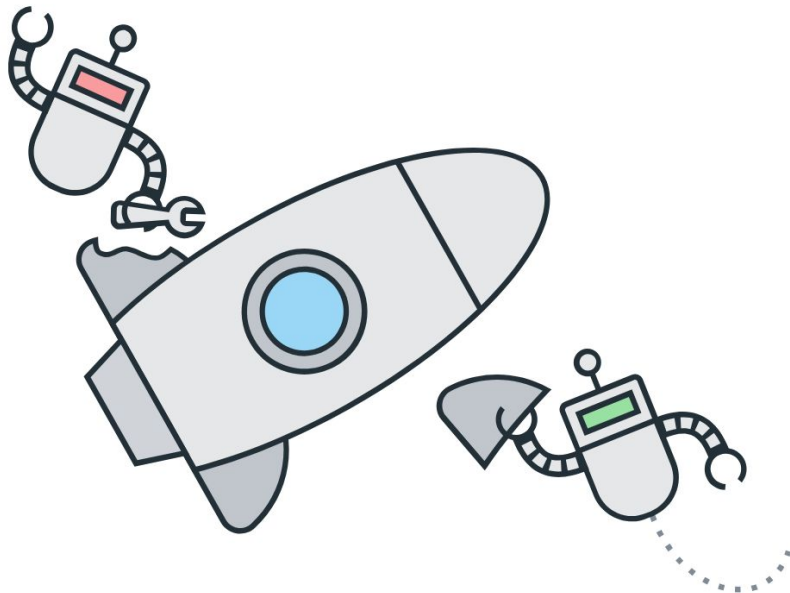
¿Cuáles son ejemplos de uso de TCP y UDP?



Hablando de *networking*...

El pasado lunes 20 de octubre,
¿tuvieron **problemas para acceder a
algunas páginas web**?

¿Cómo se relaciona esto con el
contenido de esta semana?



Canvas is experiencing issues due to an ongoing AWS incident.

We are working on recovering as soon as possible.

You can follow the status of the issue at <https://health.aws.amazon.com/health/status> and <https://status.instructure.com/>

Webservices

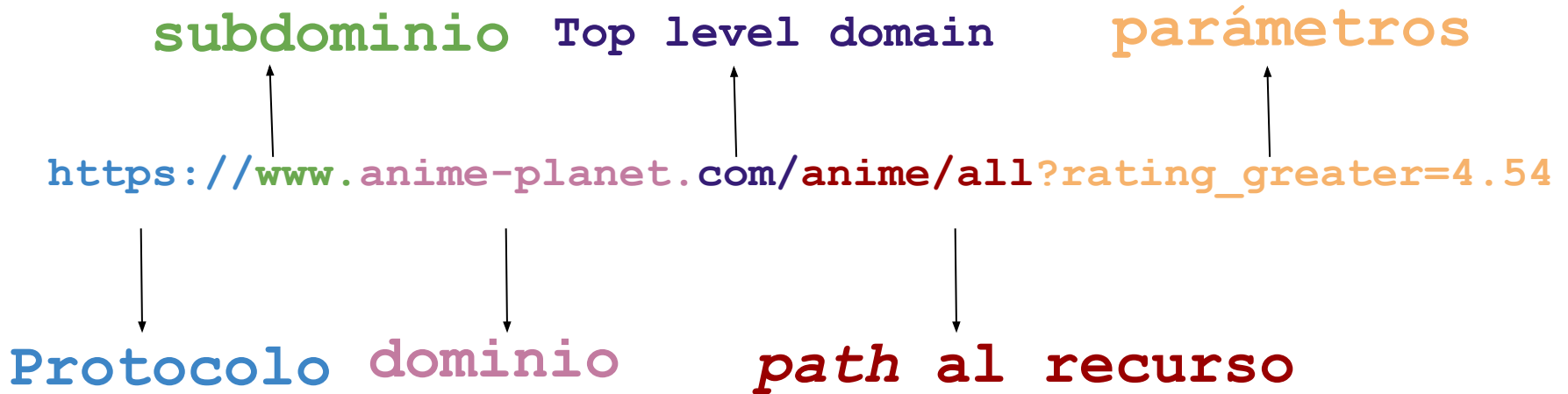


URL y HTTP

- ¿Qué compone una URL?
- ¿Cuáles son los métodos de HTTP y qué hace cada uno?
- ¿Cuáles son los *status codes* más comunes?

URL (*Uniform Resource Locator*)

Se definió un **formato que deben cumplir las URLs** para encontrar a lo que sea que deseen buscar en internet.



URL (*Uniform Resource Locator*)

Todo enlace, por intimidante que se vea, lo cumple.

https://es.aliexpress.com/item/1005006792016183?spm=a2g0o.productlist.main.1.3d1aeGRMeGRMwJ&algo_pvid=4a7a04ee-41fe-457b-8b5a-8e46d7554e61&algo_exp_id=4a7a04ee-41fe-457b-8b5a-8e46d7554e61-0&pdp_npi=4%40dis%21CLP%212457%211899%21%21%212.60%212.01%21%40210318c317174331085181675e3076%2112000038316839442%21sea%21CL%21171744362%21&curPageLogUid=ZC08JbSCPYCs&utparam-url=scene%3Asearch%7Cquery_from%3A

Ejemplo: Verbos HTTP





Supongamos que tenemos la página web <http://www.paginaic2233.com/>, que tiene un recurso estudiantes. Para poder interactuar con los datos de la página, se habilitaron las siguientes URLs:

- <http://www.paginaic2233.com/estudiantes>
- <http://www.paginaic2233.com/estudiante/<id>>

donde el **id** es un número.

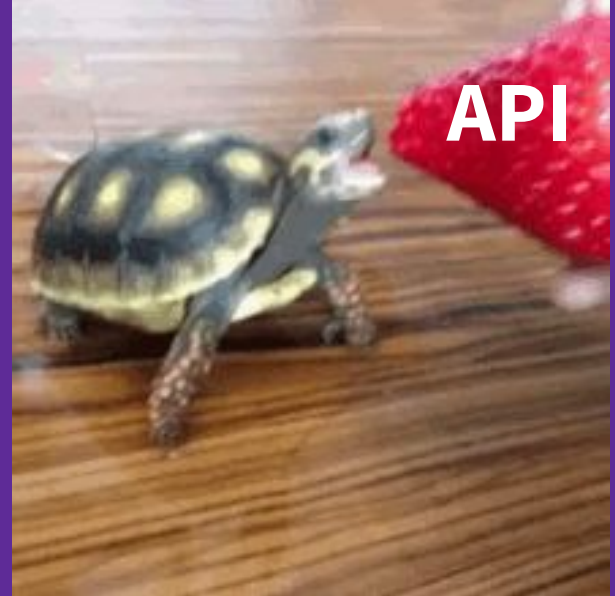
Considerando esto, podríamos pensar que dicha página tiene las siguientes acciones:

Ejemplo: Verbos HTTP

URL	GET	POST	PUT	PATCH	DELETE
<u>/estudiantes</u>	Obtiene una lista de todos los estudiantes	Crea un nuevo estudiante			
<u>/estudiante/1</u>	Obtiene los datos del estudiante con id 1		Reemplaza todos los datos del estudiante con id 1	Reemplaza algunos atributos del estudiante con id 1	Elimina al estudiante con id 1

*Esto es solo un ejemplo, las funcionalidades dependerán de lo que decida quien implementó el *webservice*.

Consumo y creación de APIs



Consumo de APIs

¿Qué herramientas que usamos todos los días creen que utilizan APIs?

¿Qué información se puede mandar a través de APIs?

Pensando tanto en las *requests* como las *responses*:

- ¿Qué información siempre trae cada una? ¿Qué información es opcional?

Servidor
HTTP

Cliente
HTTP

Solicitud (*request*)

Headers

Body (opcional)

Respuesta (*response*)

Headers

Body (opcional)

Temas adicionales de interés



Temas adicionales de interés

- Hay tokens que incluso pueden almacenar información: [JWT.io](https://jwt.io)
- Existen muchos más códigos HTTP
 - [Códigos de estado de respuesta HTTP](#)
 - Existe su explicación gatuna: [HTTP Cats](#)
 - El [código 418](#) es cuando el servidor se rehúsa a preparar café porque es una tetera.
- Existen otras librerías para programar rápidamente una API como [FastApi](#).

Pregunta Evaluación Escrita



Veamos una pregunta de Evaluación Escrita

(Examen 2024-1)

23. Desde el cliente, se necesita consultar una API, que cumple el protocolo HTTP, para actualizar **únicamente** el número de teléfono del usuario. Con este objetivo, el cliente accede al *endpoint* "`/ {nombre_usuario} / datos-personales`".

¿Cuál de los siguientes métodos deberá estar diseñado desde el servidor de dicha API para realizar esa acción?

- A) POST
- B) PUT
- C) DELETE
- D) PATCH
- E) GET

Veamos una pregunta de Evaluación Escrita

(Examen 2024-1)

23. Desde el cliente, se necesita consultar una API, que cumple el protocolo HTTP, para actualizar **únicamente** el número de teléfono del usuario. Con este objetivo, el cliente accede al *endpoint* `" / {nombre_usuario} / datos-personales "`.

¿Cuál de los siguientes métodos deberá estar diseñado desde el servidor de dicha API para realizar esa acción?

- A) POST
- B) PUT
- C) DELETE
- D) PATCH**
- E) GET

Veamos una pregunta de Evaluación Escrita

(Examen 2024-1)

15. ¿Cuál de las siguientes afirmaciones sobre el uso de APIs es correcta?

- A) Los códigos de respuesta siempre se envían en el *body* de la respuesta de la API al cliente.
- B) En cualquier *endpoint* que reciba parámetros, estos siempre deben enviarse en el *body* de la solicitud.
- C) En un *endpoint* de una API cualquiera, el método POST siempre se usará para crear un recurso nuevo en el servidor.
- D) Todas las APIs usan el formato JSON en sus respuestas a solicitudes de clientes.
- E) Ninguna de las afirmaciones anteriores es correcta.

Veamos una pregunta de Evaluación Escrita

(Examen 2024-1)

15. ¿Cuál de las siguientes afirmaciones sobre el uso de APIs es correcta?

- A) Los códigos de respuesta siempre se envían en el *body* de la respuesta de la API al cliente.
- B) En cualquier *endpoint* que reciba parámetros, estos siempre deben enviarse en el *body* de la solicitud.
- C) En un *endpoint* de una API cualquiera, el método POST siempre se usará para crear un recurso nuevo en el servidor.
- D) Todas las APIs usan el formato JSON en sus respuestas a solicitudes de clientes.
- E) Ninguna de las afirmaciones anteriores es correcta.**

Programación Avanzada

IIC2233 2025-2

Cristian Ruz - Pablo Araneda - Francisca Ibarra - Tamara Vidal - Daniela Concha

