Publicación de cartografía para la web

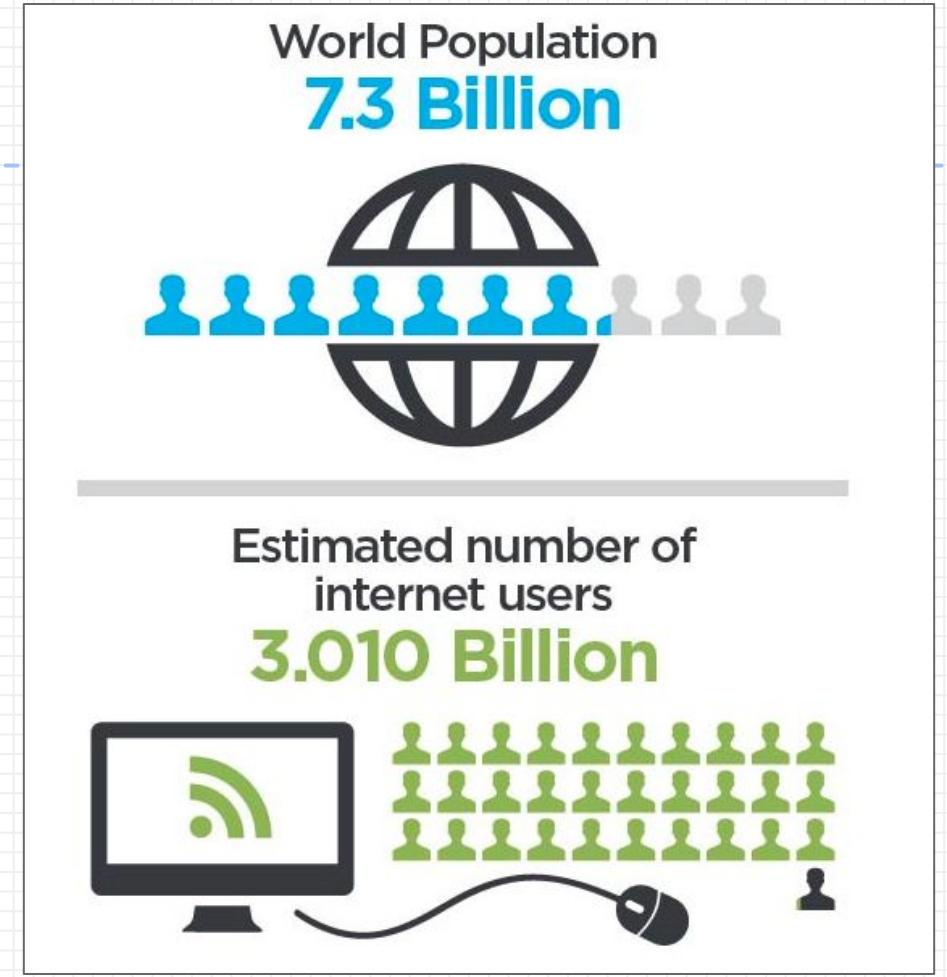
Juan Carlos Méndez
juan@gkudos.com

Especializacion en Geomática
Universidad Militar Nueva Granada

1. Fundamentos de Internet

Objetivo

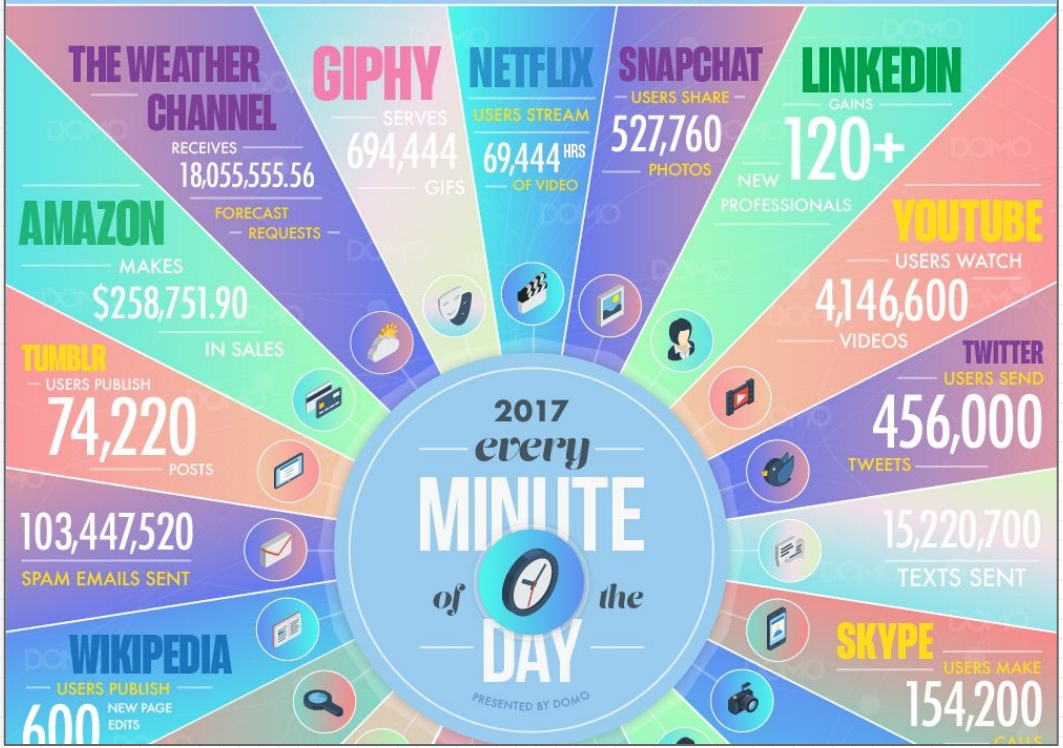
Conocer los conceptos y tecnologías básicas utilizadas para la publicación de contenido y servicios en internet


World Population
7.3 Billion
Estimated number of
internet users
3.010 Billion

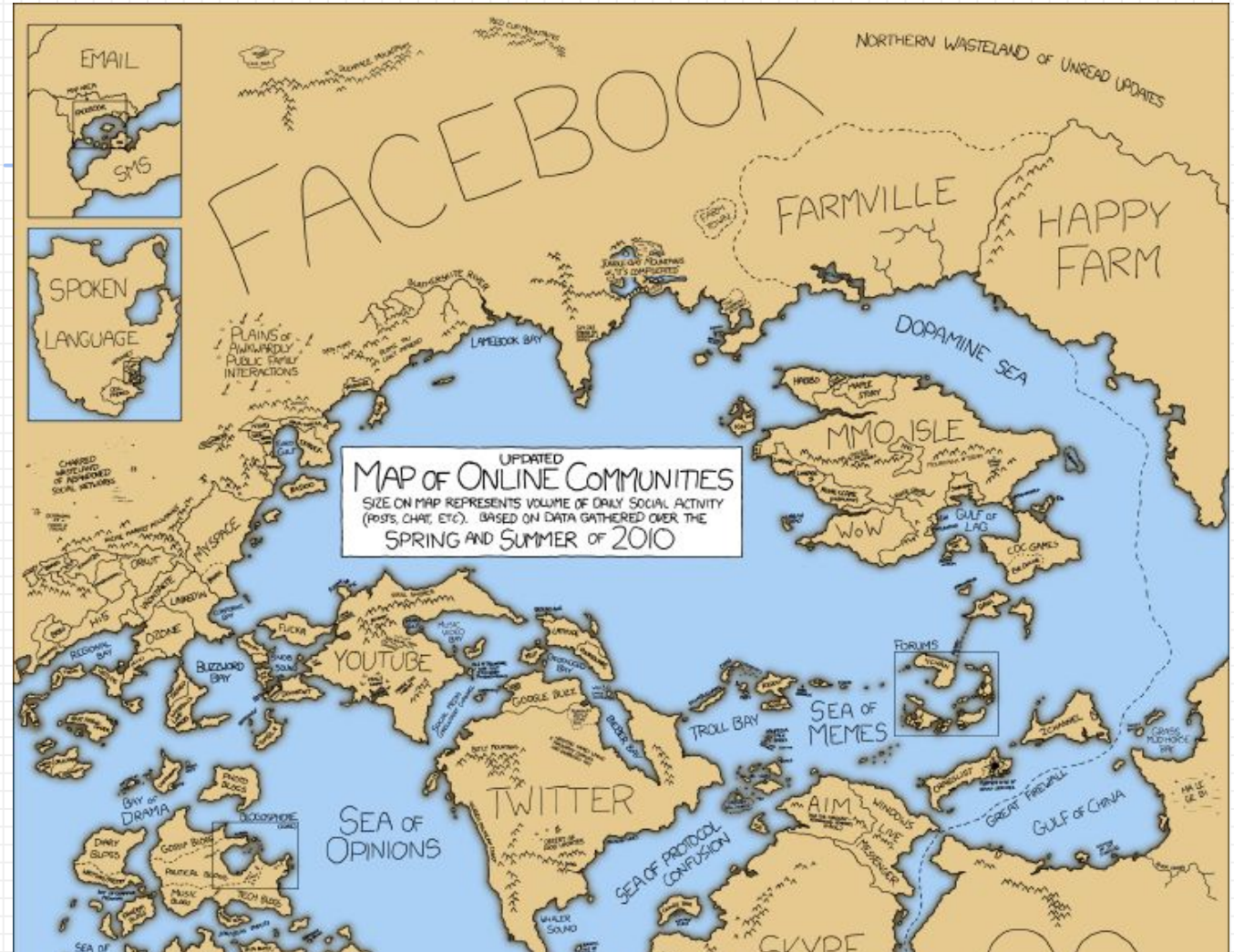
DATA NEVER SLEEPS 5.0

How much data is generated *every minute?*

90% of all data today was created in the last two years—that's 2.5 quintillion bytes of data per day. In our 5th edition of Data Never Sleeps, we bring you the latest stats on just how much data is being created in the digital sphere—and the numbers are staggering.



Pero qué es la Internet?


EMAIL
SMS
SPOKEN LANGUAGE
FACEBOOK
NORTHERN WASTELAND OF UNREAD UPDATES
FARMVILLE
HAPPY FARM
PLAINS OF AWKWARDLY PUBLIC FAMILY INTERACTIONS
LAMEBOOK BAY
DOPAMINE SEA
MMO ISLE
WoW
GULF OF LAG
UPDATED
MAP OF ONLINE COMMUNITIES
SIZE ON MAP REPRESENTS VOLUME OF DAILY SOCIAL ACTIVITY (POSTS, CHAT, ETC). BASED ON DATA GATHERED OVER THE SPRING AND SUMMER OF 2010
MYSPACE
ORKUT
LINKEDIN
HI5
ORDUM
FLICKR
YOUTUBE
BLIZZARD BAY
FORUMS
TROLL BAY
SEA OF MEMES
GOOGLE BUZZ
TWITTER
BAY OF DRAMA
SEA OF OPINIONS
AIM
WINDOWS LIVE MESSENGER
GREAT FIREWALL
GULF OF CHINA
SEA OF PROTOCOL CONFUSION
WHALER SOUND
SKYPE


THIS JEN, IS THE INTERNET.

Qué es Internet?

- ✘ Internet es una red internacional de computadoras conectadas.
- ✘ Ninguna empresa es propietaria de internet;
- ✘ Es un esfuerzo cooperativo regido por un estándares.
- ✘ Propósito: conectar computadoras entre sí para compartir información.

Fuente: https://github.com/jakobzhao/geog371/tree/master/lectures/lec00

Qué es Internet?

- Los métodos estandarizados para transferir datos o documentos a través de una red se conocen como protocolos.
- Ejemplos de protocolos: correo electrónico (POP3 / IMAP / SMTP), transferencia de archivos (FTP), shell seguro (SSH), Protocolo de transferencia de hipertexto (HTTP), etc.

Qué es la Web?

- ✘ La web *(originalmente llamada World Wide Web, la "www" en las direcciones de los sitios)* es un subconjunto de internet.
- ✘ La web es una de las formas en que la información se puede compartir a través de Internet.
- ✘ Permite que los documentos se unan entre sí a través de enlaces de hipertexto, formando así una gran "red" de información conectada.

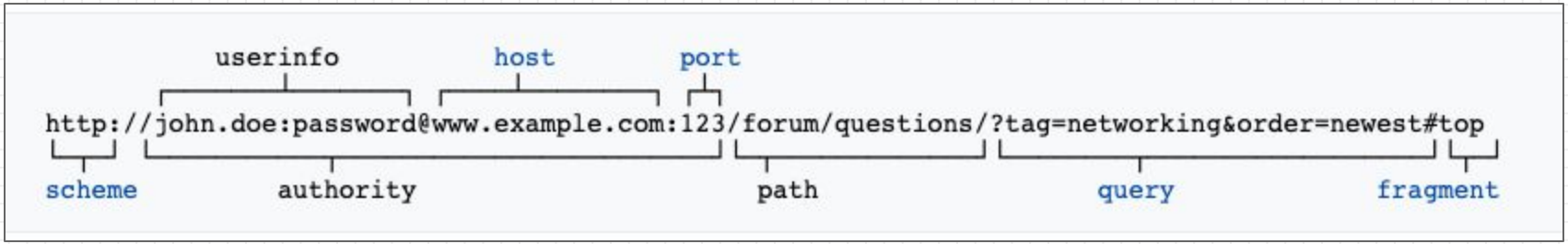
Qué es la Web?

- La web utiliza un protocolo llamado HTTP (Protocolo de transferencia de hipertexto)

http://www.elespectador.com/

- Este protocolo define las reglas para transmitir hipertexto entre computadores.
- Los recursos en la web se ubican a través de "*direcciones*" denominadas "URL's"

Qué es un URL



Fuente: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Qué es el hipertexto?

- ✘ *"...sistema que permite enlazar fragmentos de textos entre sí, lo que permite al usuario acceder a la información a través de los ítems relacionados en vez de hacerlo de forma secuencial..."*
- ✘ *"El hipertexto es texto que contiene enlaces a otros textos (hipervínculos)."*

Principales hitos en la historia y desarrollo			
Año	**Sistema**	**Autor**	**Hito**
1945	Memex	Vannevar Bush	Dispositivo basado en microfichas.
1965	Xanadú	Ted Nelson	Primero en nombrar el término "hypertext".
1967	Hypertext Editing System	Andy van Dam	Primer sistema de hipertexto en funcionamiento.
1968	On Line System	Douglas Engelbart	Sistema de hipertexto con manipulación directa con utilización de ratón.
1978	Aspen Movie Map	Andrew Lippman	Primer sistema hipermedio en funcionamiento.
1985	Intermedia	Yankelovich et al	Se empieza a utilizar el concepto de ancla y red.
1986	GUIDE	OWL	Primer producto para autoría de hiperdocumentos.
1987	HyperCard	Apple	Producto entregado con cada Macintosh.
1987	Hypertext ´87	University of North Carolina	Primera conferencia en la que se trata la tecnología de hipertexto.
1991	World Wide Web	Tim Berners-Lee	Proyecto para llevar la tecnología hipermedial en Internet.
1993	Mosaic	NCSA	Navegador gráfico para el WWW.

Fuente: https://www.significados.com/hipertexto/ https://es.wikipedia.org/wiki/Hipertexto

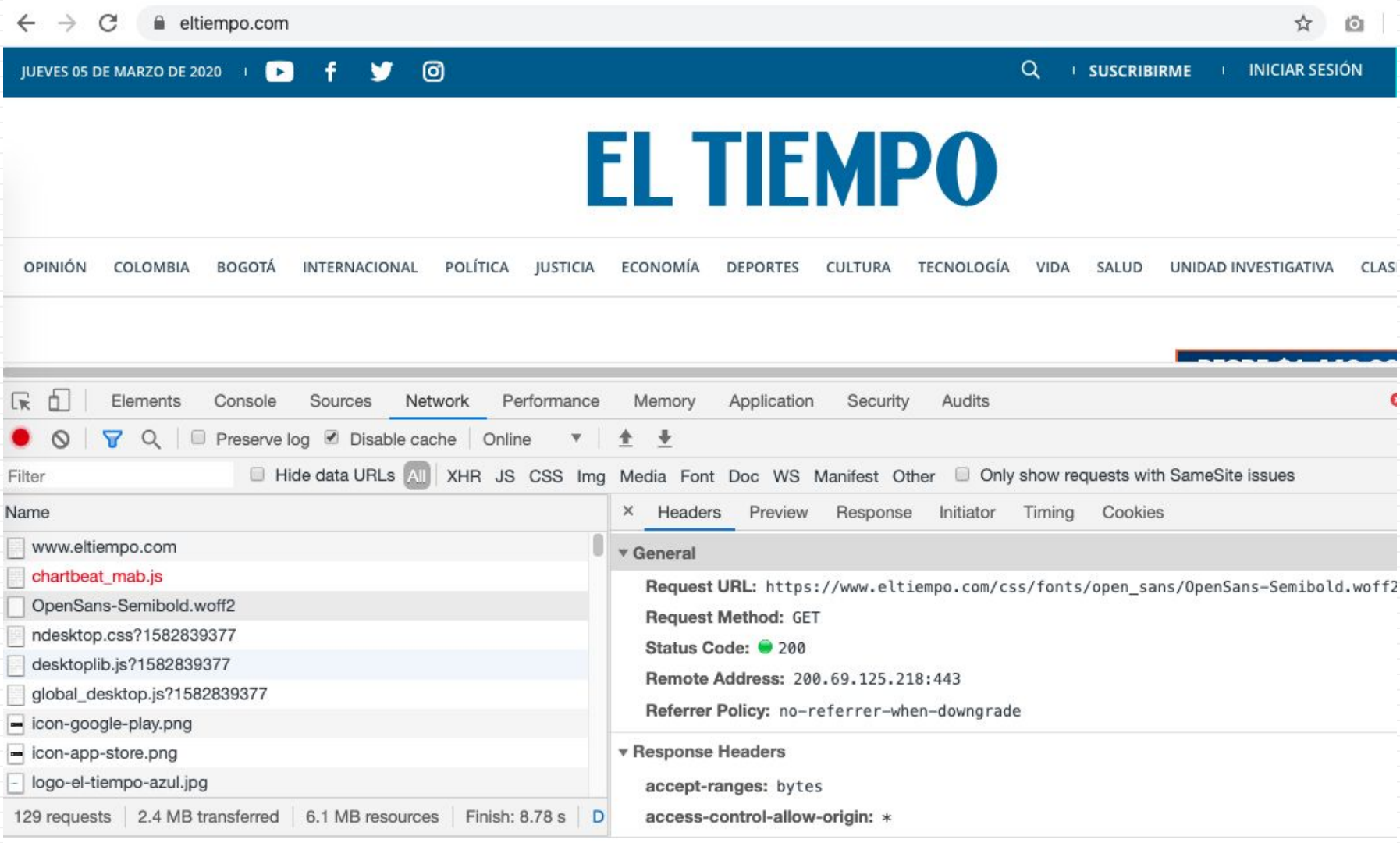
Qué es la Web?

- ✘ HTTPS: Extensión de Http para facilitar la comunicación segura entre computadores

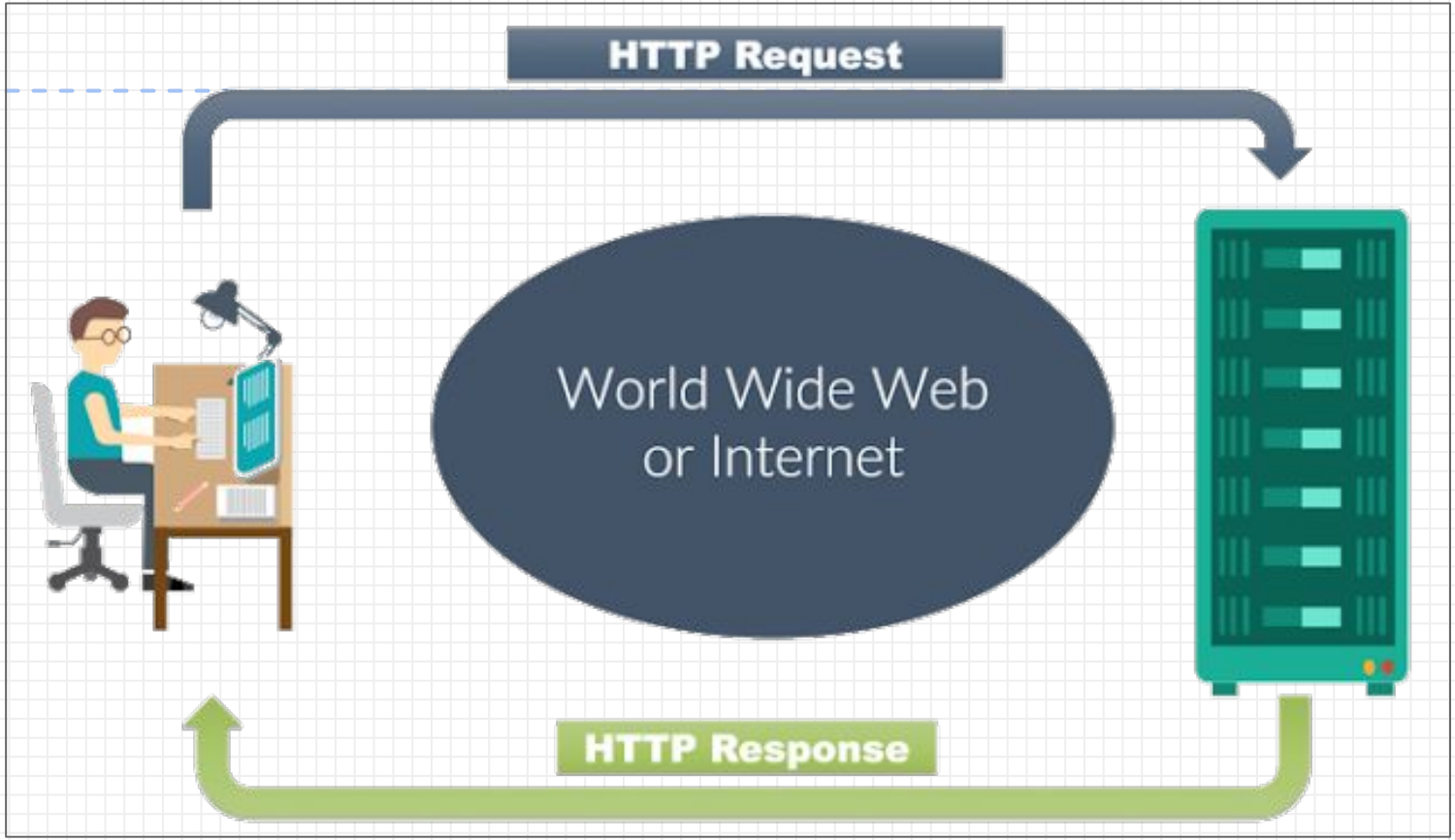
https://www.elespectador.com/

Qué es un navegador web? (Browser)

- ✘ Aplicación que permite acceder a información publicada en la web.
- ✘ El navegador se encarga de descargar los recursos publicados a través de URL’s


eltiempo.com
JUEVES 05 DE MARZO DE 2020
SUSCRIBIRME
INICIAR SESIÓN
EL TIEMPO
OPINIÓN COLOMBIA BOGOTÁ INTERNACIONAL POLÍTICA JUSTICIA ECONOMÍA DEPORTES CULTURA TECNOLOGÍA VIDA SALUD UNIDAD INVESTIGATIVA
Elements Console Sources Network Performance Memory Application Security Audits
Preserve log Disable cache Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Only show requests with SameSite issues
Name
www.eltiempo.com
chartbeat_mab.js
OpenSans-Semibold.woff2
ndesktop.css?1582839377
desktoplib.js?1582839377
global_desktop.js?1582839377
icon-google-play.png
icon-app-store.png
logo-el-tiempo-azul.jpg
129 requests 2.4 MB transferred 6.1 MB resources Finish: 8.78 s
Headers Preview Response Initiator Timing Cookies
General
Request URL: https://www.eltiempo.com/css/fonts/open_sans/OpenSans-Semibold.woff2
Request Method: GET
Status Code: 200
Remote Address: 200.69.125.218:443
Referrer Policy: no-referrer-when-downgrade
Response Headers
accept-ranges: bytes
access-control-allow-origin: *

Modelo Cliente Servidor



Proceso Consulta Página Web

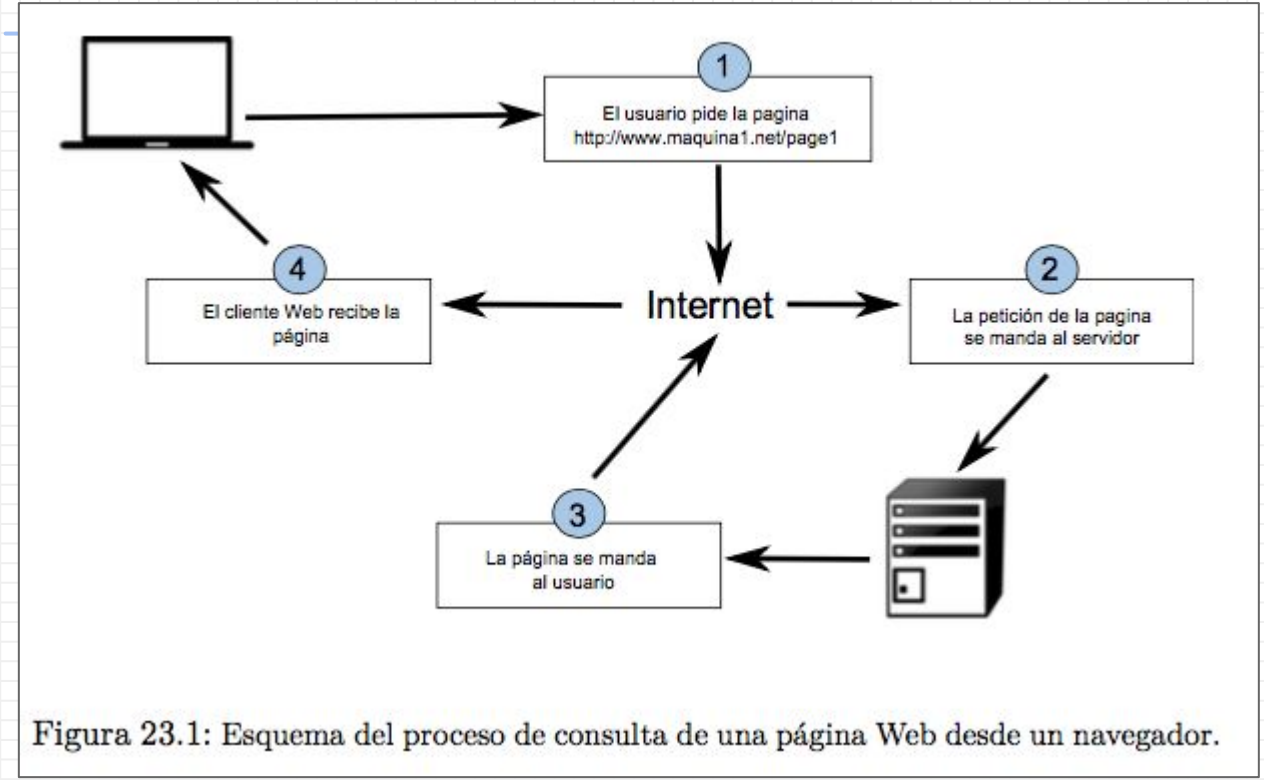
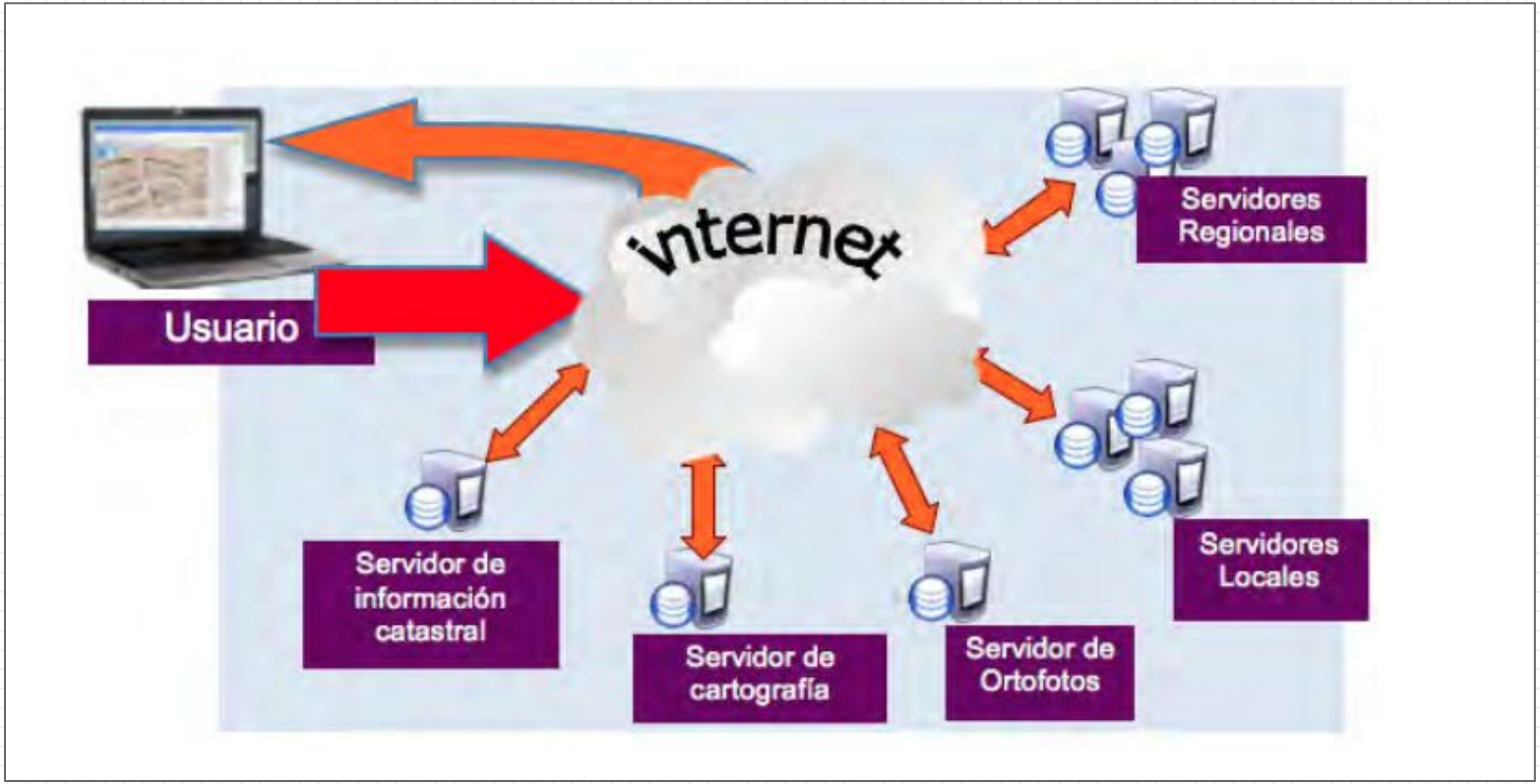


Figura 23.1: Esquema del proceso de consulta de una página Web desde un navegador.

IDE - Infraestructura de datos Espaciales



Arquitectura Cliente Servidor

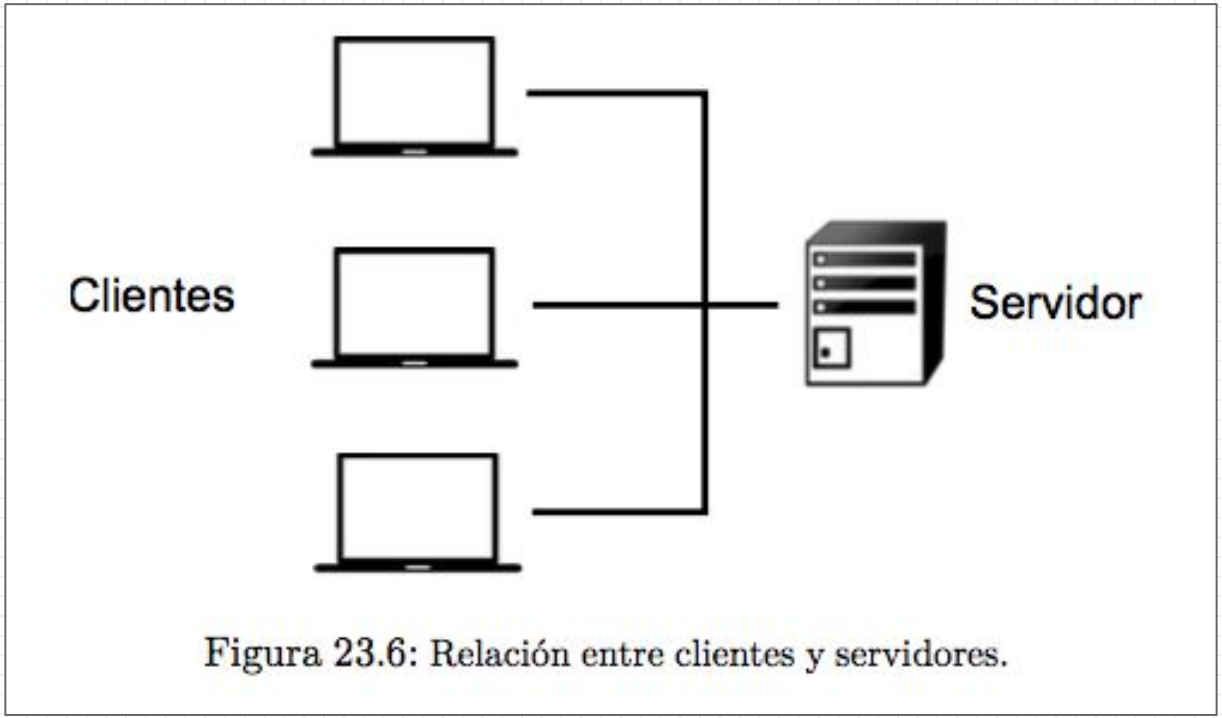


Figura 23.6: Relación entre clientes y servidores.

Sistemas de Información Geográfica - Un libro libre de Víctor Olaya

Arquitectura Servidor

- ✘ Entrega datos (Ejm. Shp, geojson)
- ✘ Entrega representaciones de los datos (Ejm: Imágenes)
- ✘ Sirve consultas (Filtros)
- ✘ Sirve procesamiento. (Ejm. Generar polígono de influencia a x distancia de un objeto)

Clientes

- ✘ Ligeros: Navegadores Web, Aplicaciones móviles
- ✘ Pesados: Aplicaciones Desktop

Anatomía de una página web

Página Html Simple

```
<html>
  <head>
    <title>Your title goes here</title>
  </head>
  <body bgcolor="white" text="red">
    <h1>My first Web page</h1>
    This is my first web page!
  </body>
</html>
```

Página Html Simple: Ejercicio

- ✘ Utilizando un editor de texto (*ejm: notepad, visual studio code*) cree un archivo *“hello.html”* con el contenido de la página web simple.
- ✘ Abra el archivo *“hello.html”* con su navegador web

Publicar Página Html Simple

Qué debo hacer para que cualquier persona en el mundo pueda ver mi página web “*hello.html*“ ?

Página Html Simple

Para permitir el acceso a los usuarios debe publicar su archivo en un servidor web.

Ejemplo:

https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/hello.html

Tecnologías para la construcción de páginas web

- Html 5 / Javascript / CSS

HTML5

*"HTML (HyperText Markup Language / **Lenguaje de Marcas de Hipertexto**) es un lenguaje de marcado que se utiliza para la creación de **páginas de Internet.***

HTML sirve para indicar la estructura de una página web. Esto lo hace por medio de las marcas de hipertexto las cuales son etiquetas conocidas en inglés con el nombre de ***tags***.

https://codigofacilito.com/articulos/que-es-html

HTML5


```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/ejemplo_html.html

CSS

*CSS “**Cascade Style Sheets / Hojas de estilo en cascada**) es un lenguaje que describe el estilo de los elementos de un documento Html.*

CSS describe como deben ser desplegados los elementos HTML

CSS

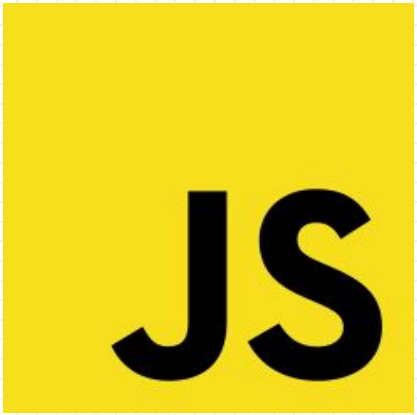

```
<html>
<head>
<style>
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/ejemplo_html_css.html

Javascript

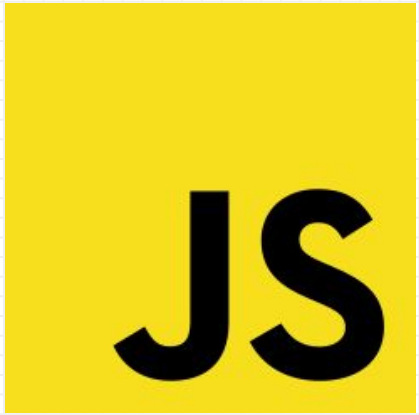
Javascript es un lenguaje de programación que permite a los desarrolladores crear interacciones en páginas web html.

Ejm: Actualizar contenido de la página, cambiar estilos, cargar imágenes, etc



Javascript

```
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button"
onclick='document.getElementById("demo").innerHTML = "Hello
JavaScript!"'>Click Me!</button>

</body>
</html>
```



https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/ejemplo_html_js.html

Ejemplo Html + CSS + JS

Vista página web:

https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/ejemplo_html_css_js.html

Código:
https://github.com/dersteppenwolf/cartografia_web/blob/master/01_Fundamentos/ejemplo_html_css_js.html

Ejercicio 1: Publicar página web a través de servidor de Github pages

Ejercicio 2: Mi primer mapa web

Ejercicio 3: Publique contenido web utilizando Markdown y Github

Otros conceptos

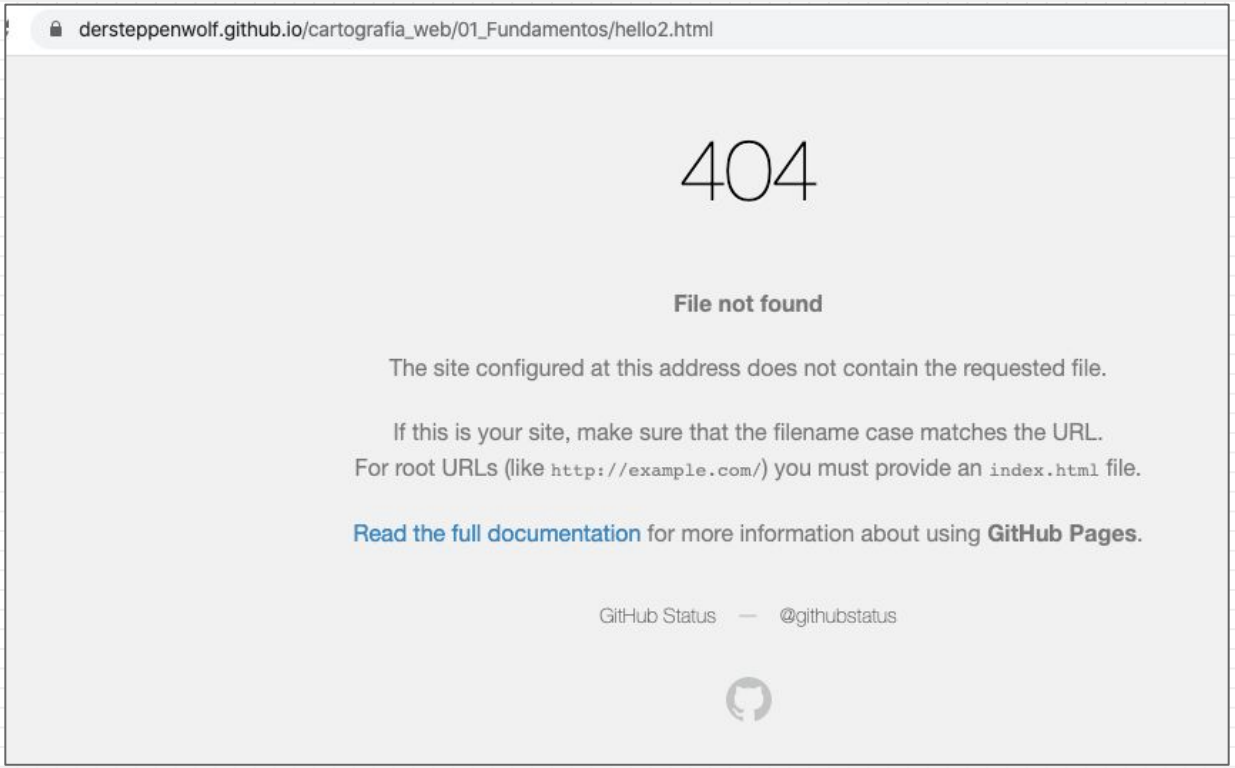
HTTP

- El Protocolo de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol o HTTP) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. (http://bit.ly/2nxS4c1)
- Maneja códigos de estado según la respuesta que el servidor envía al cliente https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

SORRY

PAGE NOT FOUND

https://dersteppenwolf.github.io/cartografia_web/01_Fundamentos/hello2.html



HTTP

Summary table

HTTP Method	RFC	Request Has Body	Response Has Body	Safe	Idempotent	Cacheable
GET	RFC 7231	Optional	Yes	Yes	Yes	Yes
HEAD	RFC 7231	No	No	Yes	Yes	Yes
POST	RFC 7231	Yes	Yes	No	No	Yes
PUT	RFC 7231	Yes	Yes	No	Yes	No
DELETE	RFC 7231	No	Yes	No	Yes	No
CONNECT	RFC 7231	Yes	Yes	No	No	No
OPTIONS	RFC 7231	Optional	Yes	Yes	Yes	No
TRACE	RFC 7231	No	Yes	Yes	Yes	No
PATCH	RFC 5789	Yes	Yes	No	No	No

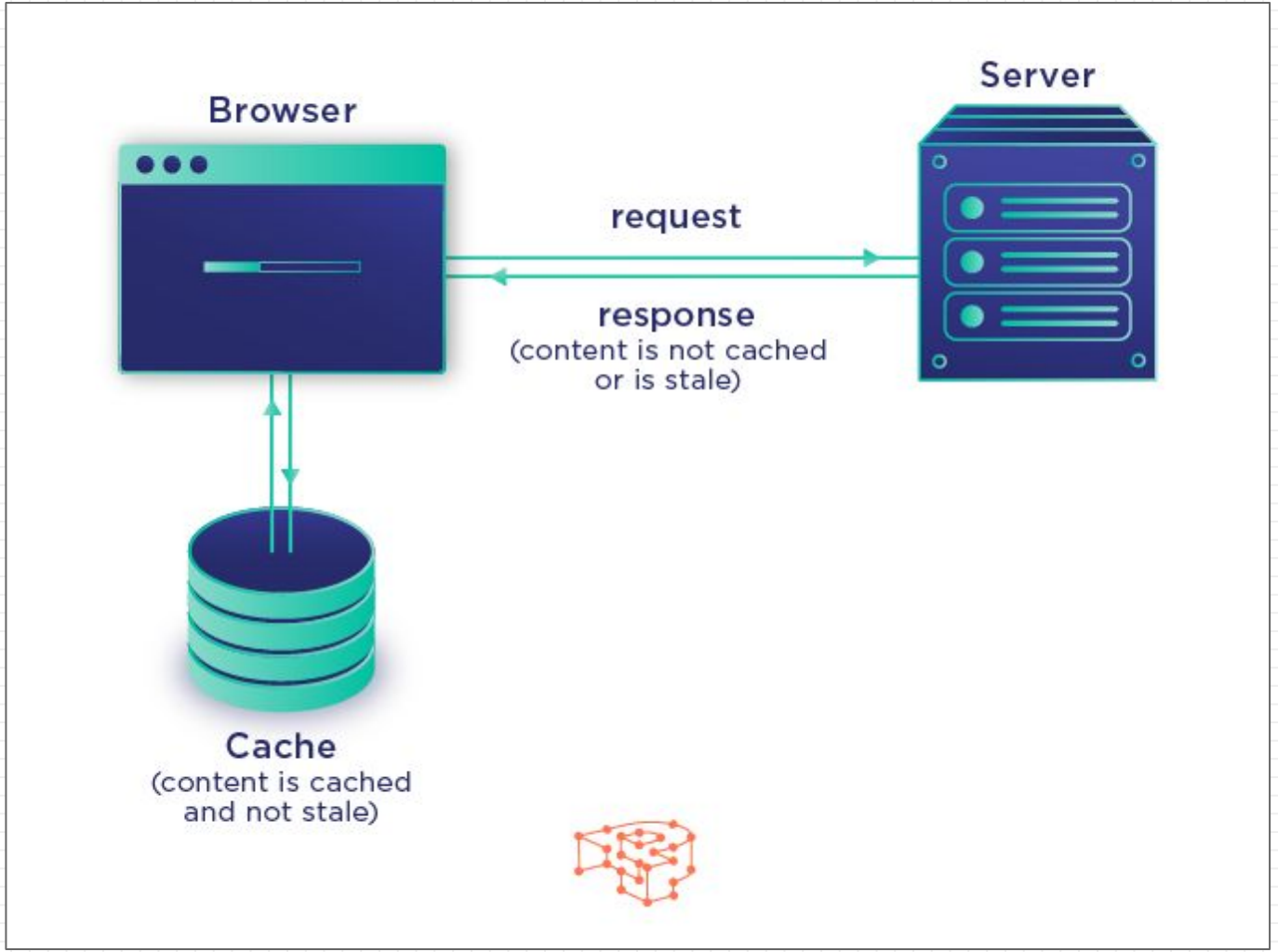
RPC

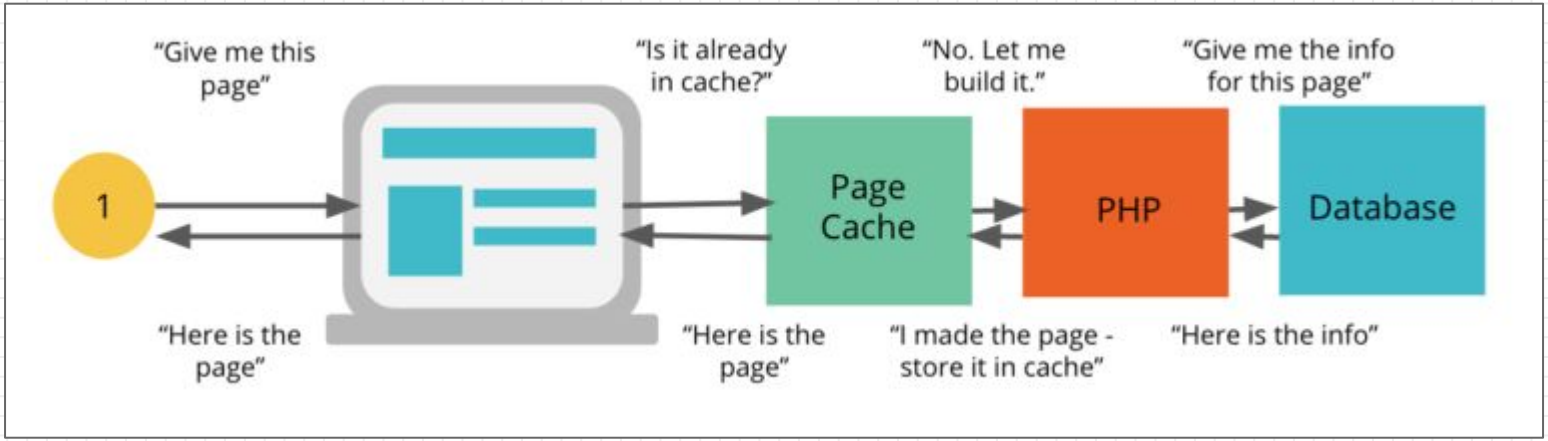
- Remote Procedure Call
- Invocación de procedimiento remoto
- Cuando un programa cliente ejecuta un procedimiento en una máquina remota
- Retos
  - Diversidad de sistemas operativos, hardware, codificación de datos, tiempos de respuesta, interoperabilidad, etc

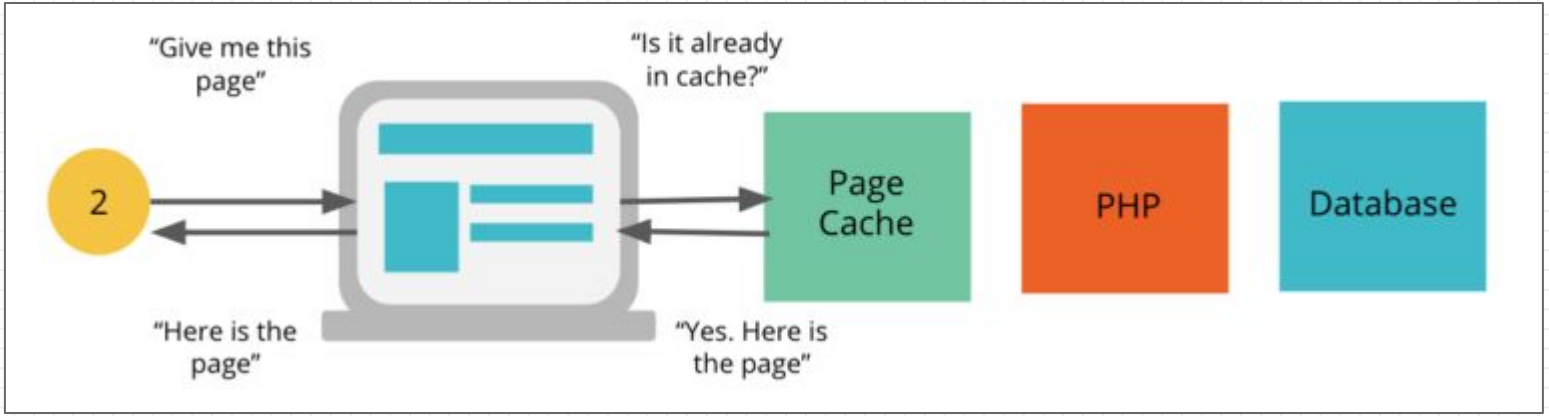
Caché

- En el cliente: Tecnología que permite el almacenamiento temporal local en el cliente de recursos como páginas web o imágenes

- En el servidor: Permite la optimización del manejo de los recursos de los servidores a través del almacenamiento temporal de los resultados de la ejecución de un proceso (Ejm: generación de una página web o una imagen)


Server
Browser
request
response
(content is not cached
or is stale)
Cache
(content is cached
and not stale)


"Give me this page"
"Is it already in cache?"
"No. Let me build it."
"Give me the info for this page"
1
Page Cache
PHP
Database
"Here is the page"
"Here is the page"
"I made the page - store it in cache"
"Here is the info"


"Give me this page"
"Is it already in cache?"
2
Page Cache
PHP
Database
"Here is the page"
"Yes. Here is the page"

XML

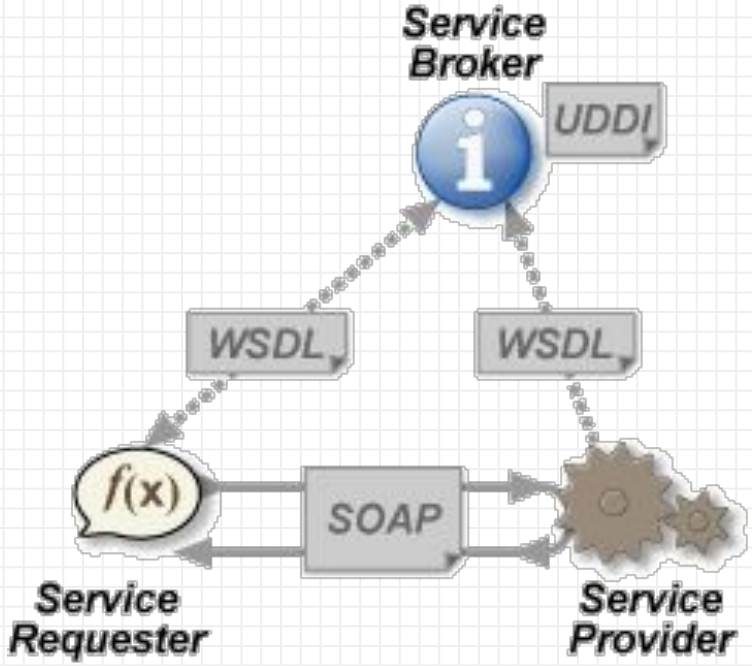
- Lenguaje que permite describir la estructura y datos particulares de alguna aplicación utilizando texto plano

XML - Ejemplo respuesta api last.fm

```
<artist>
  <name>Cher</name>
  <mbid>bfcc6d75-a6a5-4bc6-8282-47aec8531818</mbid>
  <url>http://www.last.fm/music/Cher</url>
  <image size="small">http://userserve-ak.last.fm/serve/50/285717.jpg</image>
  <image size="medium">http://userserve-ak.last.fm/serve/85/285717.jpg</image>
  <image size="large">http://userserve-ak.last.fm/serve/160/285717.jpg</image>
  <streamable>1</streamable>
  <stats>
    <listeners>196440</listeners>
    <plays>1599101</plays>
  </stats>
  <similar>
    <artist>
      <name>Madonna</name>
      <url>http://www.last.fm/music/Madonna</url>
      <image size="small">http://userserve-ak.last.fm/serve/50/5112299.jpg</image>
      <image size="medium">http://userserve-ak.last.fm/serve/85/5112299.jpg></image>
      <image size="large">http://userserve-ak.last.fm/serve/160/5112299.jpg</image>
    </artist>
    ...
  </similar>
```

Web Service

- Un servicio web (en inglés, web service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones a través de la web.
- Generalmente basado en el protocolo SOAP (XML)



API

- ✘ Application Programming Interface
- ✘ Interfaz de programación de aplicaciones
- ✘ conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software de forma remota
- ✘ Por lo general entrega información en formatos simples como xml o json

API

Last.fm Web Services

artist.getInfo

Get the metadata for an artist. Includes biography, truncated at 300 characters.

Example URLs

JSON: /2.0/?method=artist.getinfo&artist=Cher&api_key=YOUR_API_KEY&format=json
XML: /2.0/?method=artist.getinfo&artist=Cher&api_key=YOUR_API_KEY

Params

artist (Required (unless mbid)] : The artist name
mbid (Optional) : The musicbrainz id for the artist
lang (Optional) : The language to return the biography in, expressed as an ISO 639 alpha-2 code.
autocorrect[0|1] (Optional) : Transform misspelled artist names into correct artist names, returning the correct version instead. The corrected artist name will be returned in the response.
username (Optional) : The username for the context of the request. If supplied, the user's playcount for this artist is included in the response.
api_key (Required) : A Last.fm API key.

Auth

This service does **not** require authentication.

REST

- La transferencia de estado representacional (en inglés representational state transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
- Cliente Servidor
- Operaciones bien definidas: POST, GET, PUT y DELETE.
- Direccionable por el URL
- Simplicidad, uso de caché, desempeño
- Transferencia de datos en formatos simples como json

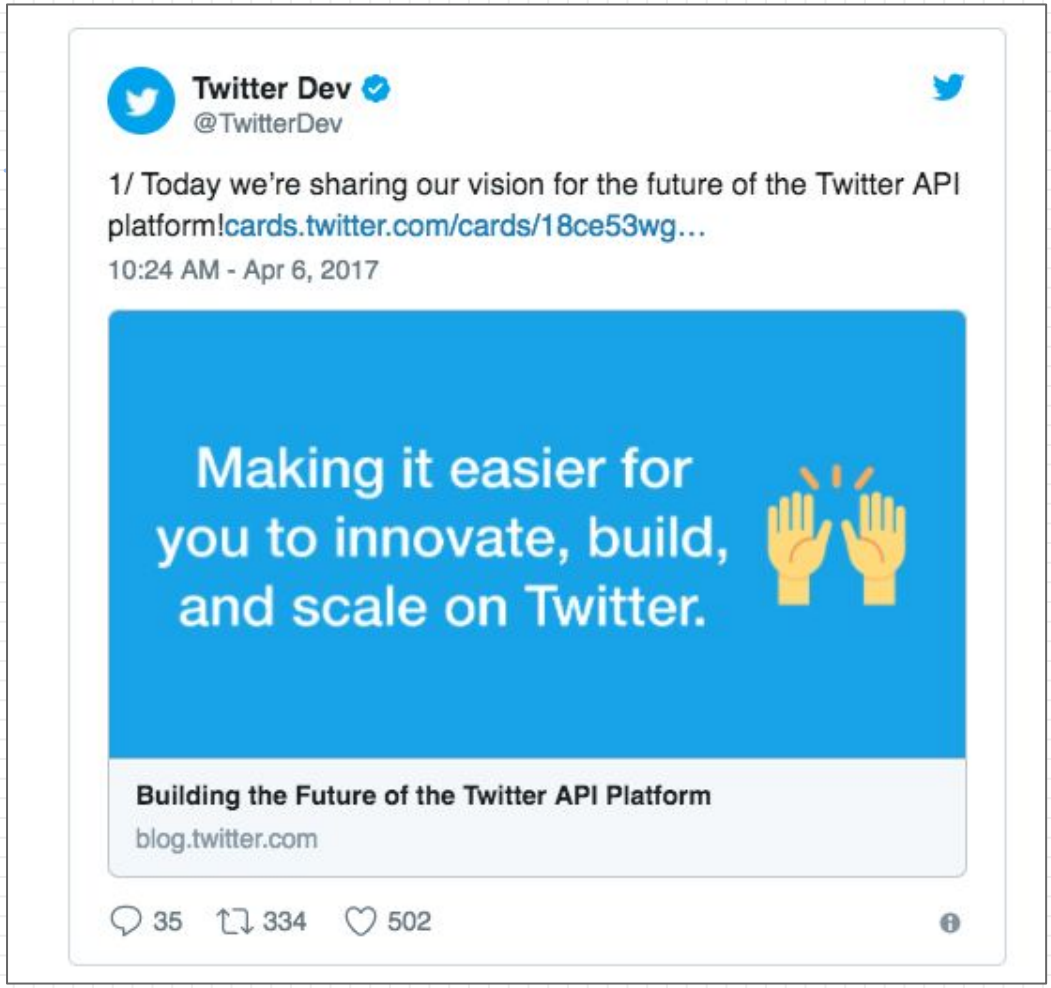
REST

Uniform Resource Locator (URL)	HTTP methods: GET	PUT	PATCH	POST	DELETE
Collection, such as `https://api.example.com/resources/`	**List** the URIs and perhaps other details of the collection's members.	**Replace** the entire collection with another collection.	Not generally used	**Create** a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.[17]	**Delete** the entire collection.
Element, such as `https://api.example.com/resources/item17`	**Retrieve** a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	**Replace** the addressed member of the collection, or if it does not exist, **create** it.	**Update** the addressed member of the collection.	Not generally used. Treat the addressed member as a collection in its own right and **create** a new entry within it.[17]	**Delete** the addressed member of the collection.

JSON

- ✘ JavaScript Object Notation
- ✘ Formato de texto ligero para el intercambio de datos
- ✘ Independiente de lenguaje de programación
- ✘ Mensajes más pequeños que SOAP

JSON


Twitter Dev
@TwitterDev
1/ Today we're sharing our vision for the future of the Twitter API platform!cards.twitter.com/cards/18ce53wg…
10:24 AM - Apr 6, 2017
Making it easier for you to innovate, build, and scale on Twitter.
Building the Future of the Twitter API Platform
blog.twitter.com
35
334
502

JSON


```
{
  "tweet": {
    "created_at": "Thu Apr 06 15:24:15 +0000 2017",
    "id_str": "850006245121695744",
    "text": "1\/ Today we\u2019re sharing our vision for the future o
    "user": {
      "id": 2244994945,
      "name": "Twitter Dev",
      "screen_name": "TwitterDev",
      "location": "Internet",
      "url": "https:\/\/dev.twitter.com\/",
      "description": "Your official source for Twitter Platform news,
    },
    "place": {

    },
    "entities": {
      "hashtags": [
```

GML

- ✘ Geography Markup Language
- ✘ GML está basado en la gramática XML y creado con el objetivo de describir objetos geográficos para su fácil intercambio sobre Internet.
- ✘ Estándar OGC

```
<gml:Point gml:id="p21" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
   <gml:coordinates>45.67, 88.56</gml:coordinates>
</gml:Point>
```

GML

```
<gml:Polygon>
    <gml:outerBoundaryIs>
            <gml:LinearRing>
                    <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
            </gml:LinearRing>
   </gml:outerBoundaryIs>
</gml:Polygon>
<gml:Point>
   <gml:coordinates>100,200</gml:coordinates>
</gml:Point>
<gml:LineString>
   <gml:coordinates>100,200 150,300</gml:coordinates>
</gml:LineString>
```

KML

- El lenguaje KML es parecido al lenguaje GML.
- Basado en la gramática XML
- permite definir objetos geográficos y cómo realizar su intercambio por medio de Internet.
- A diferencia de GML, KML tiene algunas limitaciones en la definición de estos objetos y adicionalmente incluye la parte referente a las características de visualización de estos objetos.

KML

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0"> <Document>

<Placemark>
 <Polygon> <outerBoundaryIs>  <LinearRing>
  <coordinates>
   135.2, 35.4, 0.
   135.4, 35.6, 0.
   135.2, 35.6, 0.
   135.2, 35.4, 0.
  </coordinates>
 </LinearRing> </outerBoundaryIs> </Polygon>

 <Style>
  <PolyStyle>
   <color>#a00000ff</color>
  <outline>0</outline>
  </PolyStyle>
 </Style>
</Placemark>

</Document> </kml>
```

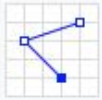
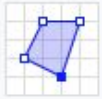
Geojson

- ✘ Es un formato estándar abierto diseñado para representar elementos geográficos sencillos, junto con sus atributos no espaciales
- ✘ Basado en json

https://tools.ietf.org/html/rfc7946

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Geojson

Point		{ "type": "Point", "coordinates": [30, 10] }
LineString		{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]] }
Polygon		{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]] }
		{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]] }

Representación de datos

GML (WFS)

```
<gml:Point srsDimension="2" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
    <gml:pos>49.40 -123.26</gml:pos>
</gml:Point>
```

Geojson

```
{ "type": "Point",
    "coordinates": [30, 10]
}
```

ESRI Json

```
{"x" : -118.15, "y" : 33.80, "spatialReference" : {"wkid" : 4326}}
```

TopoJson

TopoJSON es una extensión de geojson que codifica topología.

Su característica principal es la eliminación de arcos redundantes.

Por lo anterior logra una disminución del tamaño del archivo en comparación con un GeoJSON.

https://mapshaper.org/

TopoJson

Ejemplo:

Polígono geojson : 2 KB

Polígono topojson: 554 bytes

Tecnologías para el Desarrollo de Aplicaciones Web

Lenguajes / Plataformas de Desarrollo

- ✘ Servidor
  - ✗ Java
  - ✗ .net
  - ✗ Python
  - ✗ Go
  - ✗ Ruby
  - ✗ Node.js

Lenguajes / Frameworks para desarrollo web

- ✘ Servidor
  - ✗ Java: Spring Boot
  - ✗ .net: ASP.NET MVC
  - ✗ Python: Flask
  - ✗ Go: Revel
  - ✗ Ruby: Ruby on rails
  - ✗ Node.js : Sails
  - ✗ ...

Lenguajes / Scripting en Servidores de Mapas

- Servidor
  - Arcgis Server: Python
  - Geoserver: Python

- Ejemplos
  - Geoservicios complejos (Extracción, transformación, generación de salidas gráficas complejas, etc)

Lenguajes / Plataformas de Desarrollo

- ✘ Cliente Ligero
  - ✗ Flash / Flex

Lenguajes / Plataformas de Desarrollo

- ✘ Cliente Ligero
  - ✗ Flash / Flex

Cliente Ligero / Frameworks Javascript

- ✘ Interfaz de usuario
  - ✗ Angular
  - ✗ React
  - ✗ Vue
- ✘ Mapas
  - ✗ Leaflet
  - ✗ Openlayers
  - ✗ Mapbox gl / js
  - ✗ Arcgis Api for javascript

Lenguajes / Plataformas de Desarrollo

Cliente Pesado

- ✗ Arcgis Desktop: Python, .Net, Java
- ✗ Arcgis Pro: Python, .Net
- ✗ Qgis: Python

Ejemplos:

- ✘ Toolboxes de geoprocesamiento
- ✘ Botones personalizados

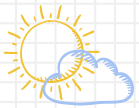
Lenguajes / Plataformas de Desarrollo

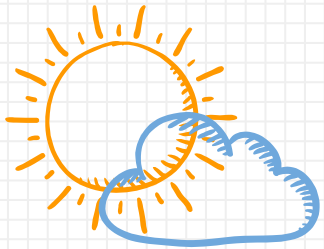
Móviles

- ✗ Nativas
  - Android: Java
  - iOS: Objective-C, Swift
- ✗ Híbridas
  - Ionic: Html, Css, Javascript
  - Xamarin: C#
  - React Native: Javascript
- ✗ Web
  - Responsive Html, Css, Javascript

Servidores de Aplicaciones

- Servidores de aplicaciones Web
  - Java: Ejm: Tomcat, Jboss
  - .Net: IIS
- Aplicaciones
  - Servidor de procesos
  - Portales Web
  - Servidor de Mapas
  - Servidor de datos abiertos
  - Servidor de metadatos
  - Servidor de ...





Gracias