





# HTTP

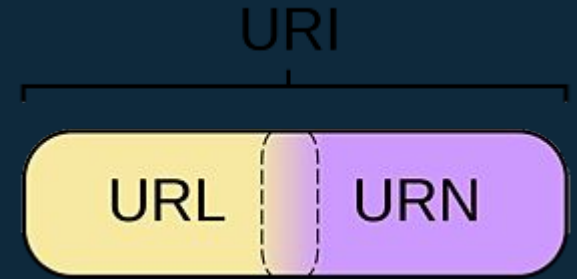
Hypertext Transfer Protocol es el protocolo de comunicación que permite las transferencias de información en la web.





# URI

Una URI es un identificador de recursos uniforme.



## URL

Indica dónde se encuentra el recurso pero puede variar.

## URN

Nombre de recurso uniforme. No indica exactamente dónde se encuentra el recurso.





# URI – Partes

- Schema
- Dominio o Dirección IP
- Número de puerto (opcional)
- Nombre del recurso
- Cadena de consulta
- Identificador de fragmento

Ejemplo:

`ftp://test.com/recurso.php?queryString#id`



# Sitio Web





# HTTP - Request

Al contactar al servidor el protocolo HTTP envía y recibe información a través de **headers**

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request  
Message  
Header

A blank line separates header & body

Request Message Body





# HTTP - Response

En la respuesta también aparecen **headers**. Puntualmente la primer línea debe indicar el estado del pedido.

```
HTTP/1.1 200 OK
```

```
Date: Sun, 08 Feb xxxx 01:11:12 GMT
```

```
Server: Apache/1.3.29 (Win32)
```

```
Last-Modified: Sat, 07 Feb xxxx
```

```
ETag: "0-23-4024c3a5"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 35
```

```
Connection: close
```

```
Content-Type: text/html
```

```
<h1>My Home page</h1>
```

Status Line

Response Headers

Response  
Message  
Header

A blank line separates header & body

Response Message Body





# HTTP – Status Codes

Algunos códigos son:

- ◇ 200 - OK
- ◇ 301 - Moved Permanently
- ◇ 307 - Temporary Redirect
- ◇ 403 - Forbidden
- ◇ 404 - Not Found
- ◇ 500 - Internal Server Error
- ◇ 503 - Service Unavailable







# ... nemotécnicamente

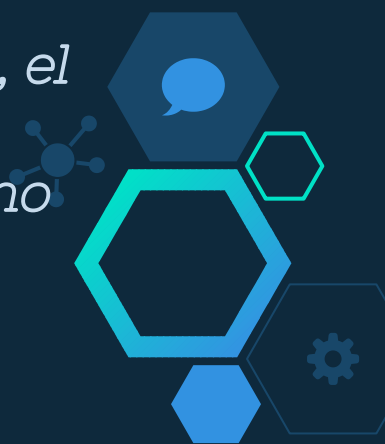
- ◆ 1xx - Hold on... (a.k.a.: aguanta un toque capo...)
- ◆ 2xx - Here we go (a.k.a.: tomá papá...)
- ◆ 3xx - Go away (a.k.a.: tomatelasss...)
- ◆ 4xx - You fucked up (a.k.a.: pifiaste fuerte...)
- ◆ 5xx - I fucked up (a.k.a.: mala mía...)





# HTTPS

*El protocolo de Transferencia de Hiper-Texto (HTTPS) es la versión segura de el http (Hyper Text Transfer Protocol) que todos conocemos y utilizamos habitualmente. Básicamente, lo que ocurre es que la página Web codifica la sesión con certificado digital. De este modo, el usuario tiene ciertas garantías de que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros.*





# HTTP – Método GET

El método GET se utiliza para **pedir información** de un recurso específico.

El query string se incluye en la URL

`/test/demo_form.php?nombre=Juan&apellido=Perez`





# HTTP – Método GET

- ◇ Los pedidos pueden ser cacheados
- ◇ Aparecen en el historial
- ◇ Pueden ser agregados a marcadores
- ◇ Nunca deberían usarse con información sensible
- ◇ Tienen restricción de longitud (2048 caracteres aunque configurable)
- ◇ Solamente deberían ser utilizados para **obtener información**





# HTTP – Método POST

El método POST **envía información** a ser procesada por un recurso específico

El query string viaja en el cuerpo del mensaje a través de HTTP y no está visible en la URL.

POST /test/demo\_form.php HTTP 1.1

Host: digitalhouse.com

**nombre=Juan&apellido=Perez**





# HTTP – Método POST


- ◇ Los pedidos no pueden ser cacheados
- ◇ No aparecen en el historial
- ◇ No pueden ser agregados a marcadores
- ◇ Sin restricción de longitud del query string.





# GET vs POST

	GET	POST
Back Button	Sin consecuencias	Se reenvía la información
Marcadores	Acepta	No acepta
Cache	Acepta	No acepta
Restricción longitud	Sí	No
Seguridad	No utilizar con datos sensibles	Más seguro que GET
Visibilidad	Se ve la información en la URL	No se ve la información en la URL





# GET vs POST en URL

Metodo get

`http://www.test.com/index.htm?name1=value1&name2=value2`

Metodo post

`http://www.test.com/index.htm`







# HTTP – Métodos

Existen otros métodos menos conocidos

- ◇ OPTIONS
- ◇ GET
- ◇ HEAD
- ◇ POST
- ◇ PUT
- ◇ DELETE
- ◇ TRACE
- ◇ CONNECT





PHP nos permite entender  
qué tipo de pedido llega y  
cómo acceder al **query string**  
a través de

Variables  
super  
globales





# PHP – Superglobals

Las variables superglobales son variables disponibles siempre en todos los ámbitos.





# PHP – \$\_GET

\$\_GET es una variable superglobal. Es un array asociativo que es generado a partir del **query string** en un pedido **GET**





# PHP – \$\_GET

Si hay un pedido a  
**localhost/test.php?nombre=Juan&apellido=Perez**

**\$\_GET** sería

**[“nombre” => “Juan”, “apellido” => “Perez”]**

Entonces, **\$\_GET[“nombre”]** tiene el valor “Juan”






# PHP – \$\_POST

\$\_POST es una variable superglobal. Es un array asociativo que es generado a partir del **query string** en un pedido **POST**.

Funciona igual a \$\_GET pero en pedidos hechos por POST.





# PHP – Superglobals

Otras superglobals:

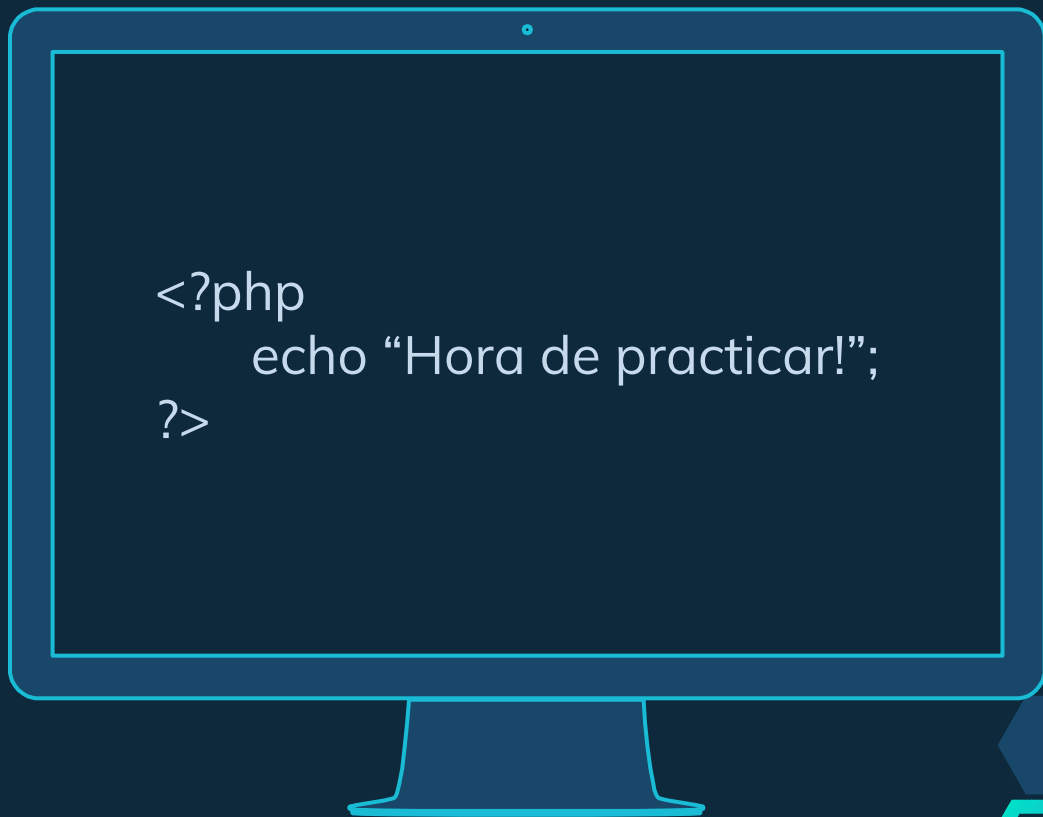
- ◇ \$GLOBALS
- ◇ \$\_SERVER
- ◇ \$\_GET
- ◇ \$\_POST
- ◇ \$\_FILES
- ◇ \$\_COOKIE
- ◇ \$\_SESSION
- ◇ \$\_REQUEST
- ◇ \$\_ENV





# ¡A practicar!

Ejercicios del 1 al 6







# Gracias!

¿Preguntas?

