

---

# SISTEMAS WEB

## CURSO 2022/2023

HTTP - HyperText Transfer Protocol  
Envío de datos del cliente al servidor:  
Formularios



Web Sistemak by [Oskar Casquero](#) & [María Luz Álvarez](#) is licensed under a [Creative Commons Reconocimiento 4.0 Internacional License](#).

# FUNCIONAMIENTO DE HTTP: FORMULARIOS

---

- **¿Cómo se envían información desde el cliente al servidor?**
  - Un cliente HTTP puede enviar diferentes tipos de datos a un servidor web:
    - datos en formato binario (ejemplo: imágenes)
    - datos en formato JSON o XML
    - **datos en formato formulario**
- En esta sesión vamos a estudiar el envío de datos en **formato formulario**, es decir, cadenas cortas de datos que se envían como pares de nombre-valor, por ejemplo:
  - Modelo genérico: name1=value1&name2=value2&name3=value3
  - Ejemplo del modelo: dni=12345678N&nombre=marta&apellido=perez

# FUNCIONAMIENTO DE HTTP:

## ENVÍO DE DATOS EN FORMATO FORMULARIO (POST)

---

- Supongamos que en un servidor web reside un recurso que calcula la letra del DNI. Si se envía una solicitud HTTP con el número DNI en un parámetro denominado **dni**, este recurso devuelve una respuesta HTTP que contiene la letra del DNI
- Para enviar el dato en formato formulario, el cliente define la siguiente petición HTTP:
  - Método: **POST**
  - Cabeceras para caracterizar el cuerpo del mensaje:
    - **Content-Type**: application/x-www-form-urlencoded
    - **Content-Length**: la longitud del cuerpo del mensaje
  - Los pares nombre-valor deben codificarse de una manera particular (**codificación UTF-8 + por ciento**) y formatearse (como un nombre = valor y un separador entre pares "&").

### Solicitud de ejemplo

```
POST /recurso HTTP/1.1
Host: sw2020.com
Accept: text/plain
Content-Type: application/x-www-form-urlencoded
Content-Length: 12
User-Agent: Cliente Python

dni=12345678
```

### Respuesta de ejemplo

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2015 08:07:43 GMT
Content-Length: 1
Content-Type: text/plain

Z
```

# FUNCIONAMIENTO DE HTTP:

## ENVÍO DE DATOS EN FORMATO FORMULARIO (GET)

---

- En lugar de enviar los pares de valores en el cuerpo del mensaje, se puede enviar al URI utilizando el carácter "?" → *query string*
  - Método: **GET**
  - **No** se deben usar los encabezados Content-Type y Content-Length
    - En *query string* solo puede enviar datos formateados.
    - La estructura de la solicitud HTTP limita la *query string*: los pares nombre- valor se encuentra entre el carácter "?" y la cadena "HTTP / 1.1".
  - La codificación y el formato de los datos se realizan como antes.
  - Desventaja: la cantidad de datos que se pueden enviar está limitada a 1024 octetos.

### Petición de ejemplo

```
GET /recurso?dni=12345678 HTTP/1.1
Host: sw2016.com
Accept: text/plain
User-Agent: Cliente Python
```

### Respuesta de ejemplo

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2015 08:07:43 GMT
Content-Length: 1
Content-Type: text/plain

Z
```

# FUNCIONAMIENTO DE HTTP:

## RESUMEN

---

GET /app/servlet/contactUs?name=XXX&email=YYY&message=ZZZ HTTP/1.1

Host: ws2017.com

Accept: text/html

Accept-Language: en-US,en

Connection: keep-alive

POST /app/servlet/contactUs HTTP/1.1

Host: ws2017.com

Accept: text/html

Accept-Language: en-US,en

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: 30

name=XXX&email=YYY&message=ZZZ

# FUNCIONAMIENTO DE HTTP:

## ENVÍO DE DATOS EN FORMATO FORMULARIO

---

- En el formato de formulario, se especifica para el contenido (las cadenas de nombre y valor de los pares nombre-valor) el siguiente **procedimiento de codificación**:
    - Si aparece un carácter especial (caracteres que son **delimitadores** en el URI o en la cadena de nombre-valor) o un carácter no recogido en el alfabeto [US-ASCII](#) (por ejemplo, un carácter con tilde), éste debe codificarse en dos pasos antes formar la URI:
      1. Codificar el carácter en [UTF-8](#). (valores en hexadecimal)
        - Ejemplo 1: & → 26
        - Ejemplo 2: á → C3 A1
      2. Codificar el resultado anterior en “porcentaje” (percent encoding)
        - Ejemplo 1 : 26 → %26
        - Ejemplo 2 : C3 A1 → %C3 %A1
- NOTA:** En los ejemplos anteriores, solo se codifica un carácter; pero debido al procedimiento de codificación, ¿cuántos caracteres se enviarán?
- Ejemplo 1: %26 → 3 caracteres: %, 2, 6
  - Ejemplo 2: %C3 %A1 → 6 caracteres: %, C, 3, %, A, 1

# FUNCIONAMIENTO DE HTTP:

## ENVÍO DE DATOS EN FORMATO FORMULARIO

---

- Ejemplo:
  - datos a enviar: nombre=Iñaki y apellido=Pérez
  - datos que se codificarán en el cuerpo del mensaje:  
nombre=I%C3%B1aki&apellido=P%C3%A9rez
- Si en la cadena del nombre o del valor de los pares nombre-valor aparece un espacio, éste se codifica con "+":
  - datos a enviar: nombre\_apellidos=Iñaki Pérez
  - datos que se codificarán en el cuerpo del mensaje:  
nombre\_apellidos=I%C3%B1aki+P%C3%A9rez
- Pero si se desea utilizar el "+" como dato en lugar de como separador, éste debe codificarse según lo indicado arriba (primero en UTF-8 y luego en porcentaje):
  - datos a enviar: num1=2, num2=2 y operador=+
  - datos que se codificarán en el cuerpo del mensaje:  
num1=2&num2=2&operato=%2B

# EJEMPLO:

## ENVÍO DE DATOS EN FORMATO FORMULARIO -- BURP

---

- Envío de datos a un servidor: obtener letra DNI
  - <https://gae-sw-2017.appspot.com/>



# EJEMPLO:

## ENVÍO DE DATOS EN FORMATO FORMULARIO -- GET

- Envío de datos a un servidor: obtener letra DNI
  - <http://gae-sw-2017.appspot.com/>

### Petición de Letra DNI GET

```
GET /processForm?dni=12345678 HTTP/1.1
Host:gae-sw-2017.appspot.com
```

### Respuesta del servidor

```
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2015 08:07:43 GMT
Content-Length: 9
Server: Google Frontend
Content-type: text/html; charset=utf-8

12345678Z
```

```
import requests
import urllib
import sys

metodo = 'GET'
uri_base = "http://gae-sw-2017.appspot.com/processForm"
cabeceras = {'Host': 'gae-sw-2017.appspot.com'}
query = {'dni': sys.argv[1]}
query_encoded = urllib.parse.urlencode(query)
uri= uri_base + '?' + query_encoded
respuesta = requests.request(metodo, uri,
                             headers=cabeceras, allow_redirects=False)
```

### Generamos la petición

```
codigo = respuesta.status_code
descripcion = respuesta.reason
print(str(codigo) + " " + descripcion)
for cabecera in respuesta.headers:
    print(cabecera + ": " + respuesta.headers[cabecera])
cuerpo = respuesta.content
print(cuerpo)
```

### Procesamos la respuesta

Datos en la  
URI

# EJEMPLO:

## ENVÍO DE DATOS EN FORMATO FORMULARIO -- POST

- Envío de datos a un servidor: obtener letra DNI
  - <https://gae-sw-2017.appspot.com/>

### Petición de Letra DNI POST

**POST** /processForm HTTP/1.1  
Host: gae-sw-2017.appspot.com  
Content-Type : application/x-www-form-urlencoded  
Content-Length:12  
  
dni=12345678

### Respuesta del servidor

HTTP/1.1 200 OK  
Date: Wed, 25 Nov 2015 08:07:43 GMT  
Content-Length: 9  
Server: Google Frontend  
Content-type: text/html; charset=utf-8  
  
12345678Z

Codificación  
de datos

```
import requests
import sys
import urllib

metodo = 'POST'
uri = "http://gae-sw-2017.appspot.com/processForm"
cabeceras = {'Host': 'gae-sw-2017.appspot.com',
             'Content-Type': 'application/x-www-form-urlencoded'}
cuerpo = {'dni': sys.argv[1]}
cuerpo_encoded = urllib.parse.urlencode(cuerpo)
cabeceras['Content-Length'] = str(len(cuerpo_encoded))
respuesta = requests.request(metodo, uri, data=cuerpo_encoded,
                             headers=cabeceras, allow_redirects=False)
```

Generamos la petición

```
codigo = respuesta.status_code
descripcion = respuesta.reason
print(str(codigo) + " " + descripcion)
for cabecera in respuesta.headers:
    print(cabecera + ": " + respuesta.headers[cabecera])
cuerpo = respuesta.content
print(cuerpo)
```

Procesamos la respuesta

# FUNCIONAMIENTO DE HTTP: FORMULARIOS (HTML)

---



(\*) image extracted from <http://www.freedback.com/>

```
<html>
  <head>
    <title>Formulario</title>
  </head>
  <body>
    <p>
      <form method="GET" action="/app/servlet/contactUs">
        <b>Name: </b>
        <input type="text" name="name">
        <br/>
        <b>Email: </b>
        <input type="text" name="email">
        <br/>
        <b>Message: </b>
        <input type="text" name="message">
        <br/><br/>
        <input type="submit">
      </form>
    </p>
  </body>
</html>
```

# EJERCICIO: CONSULTA EN EL *DIRECTORIO DE EHU* UTILIZANDO BURP Y EN PYTHON

---

- <https://www.ehu.eus/bilatu/buscar/bilatu.php?lang=es>

The screenshot shows a web form titled "BUSCAR" (Search) with the following sections:

- Datos personales** (Personal data):
  - Apellidos/ Nombre (Last name/ Name)
  - Nombre (Name)
  - Primer apellido (First surname)
  - Segundo apellido (Second surname)
  - Teléfono (Phone)
  - Correo electrónico (Email)
- Campus**:
  - ☐ Álava
  - ☐ Bizkaia
  - ☐ Gipuzkoa
- Areas** (Areas):
  - Centro (Center): -- Seleccione una opción --
  - Biblioteca (Library): -- Seleccione una opción --
  - Instituto (Institute): -- Seleccione una opción --
  - Departamento (Department): -- Seleccione una opción --
  - Servicios Centrales (Central Services): -- Seleccione una opción --
  - Rectorado (Rectorate): -- Seleccione una opción --
  - Cargo académico (Academic position): -- Seleccione una opción --