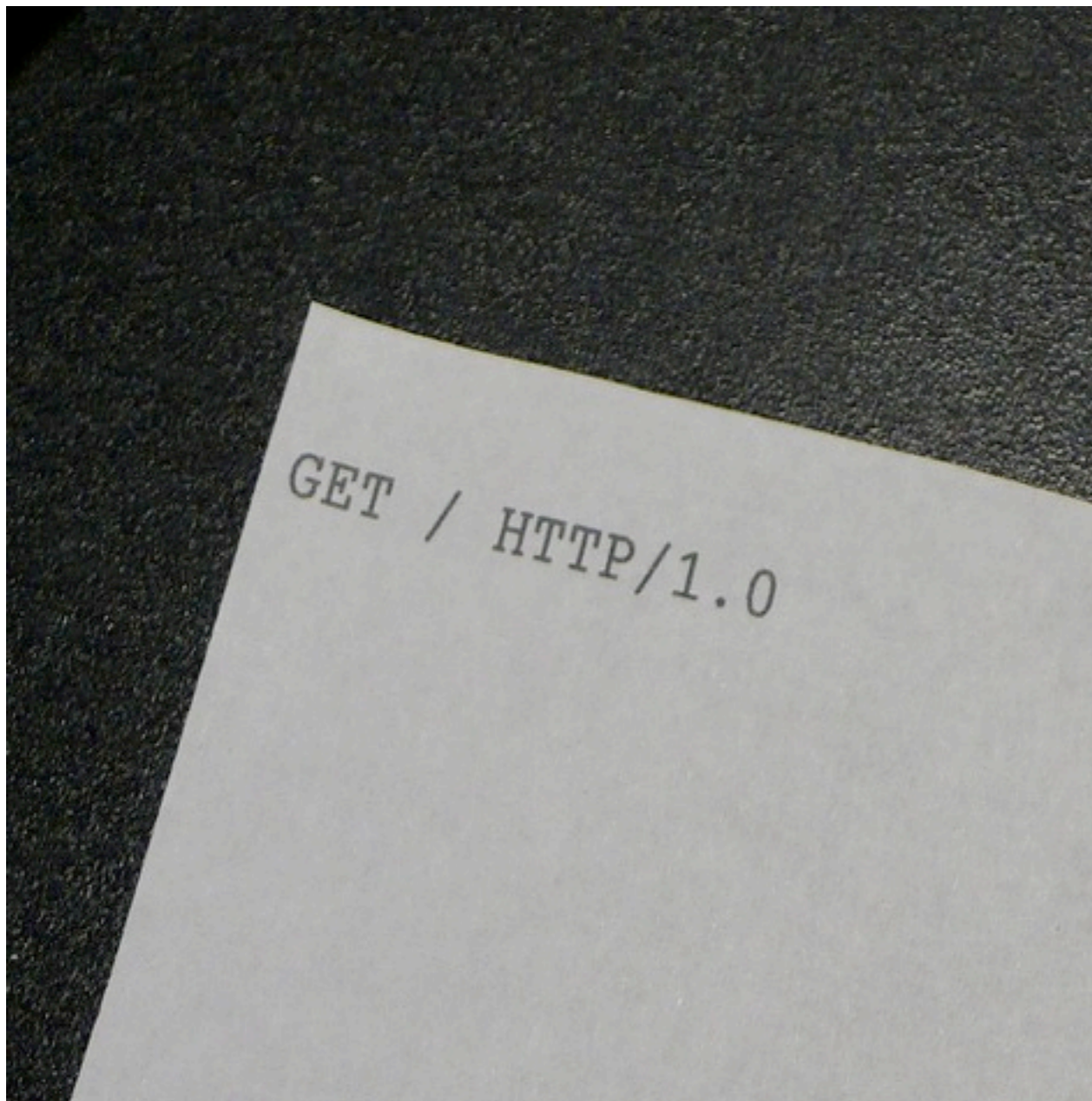


A close-up photograph of a blue mesh screen, likely a security screen or a decorative element. The screen is composed of a grid of small, dark blue squares. In the center, there is a logo consisting of a red and blue square. To the right of the logo, there is some faint, illegible text. The background behind the screen is dark and out of focus.

Tema 1

HTTP y aplicaciones web



1.1

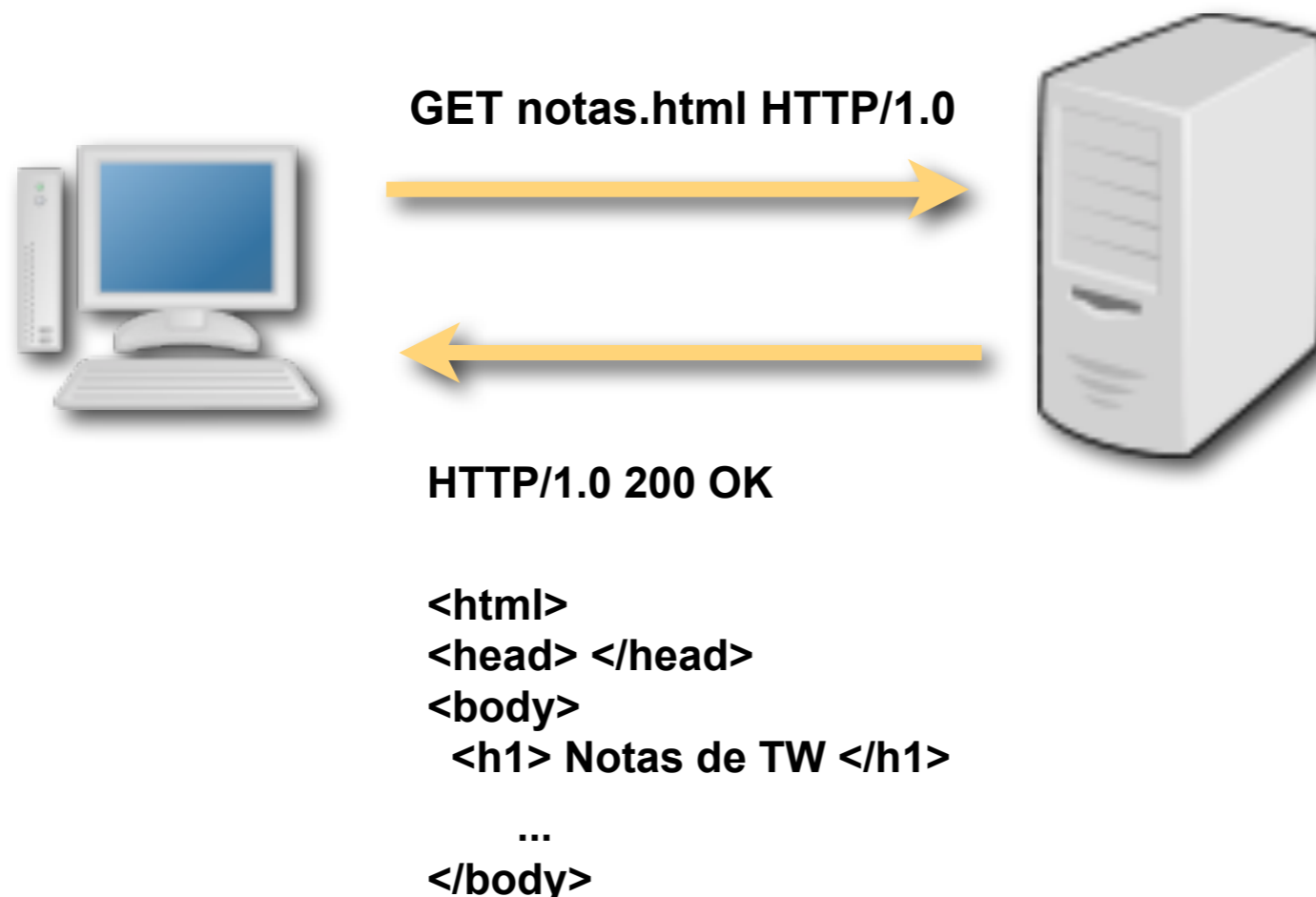
Introducción a HTTP

HTTP y aplicaciones web

Http es el protocolo cliente/servidor en el que se basa la web. Veremos aquí una explicación sencilla de cómo funciona

Petición/respuesta HTTP

- Un **servidor web** está a la escucha por un puerto, aceptando **peticiones** y haciendo **respuestas** según el protocolo http
- El protocolo especifica la sintaxis de peticiones y respuestas
- El intercambio de información se hace en modo texto



Estructura de una petición

Línea de
petición

```
GET / HTTP/1.1
```

Cabeceras

```
Host: www.ua.es
```

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6;  
rv:6.0.2) Gecko/20100101 Firefox/6.0.2
```

```
Accept: text/html,application/xhtml+xml,application/  
xml;q=0.9,*/*;q=0.8
```

```
Accept-Language: es-es,es;q=0.8,en-us;q=0.5,en;q=0.3
```

```
Accept-Encoding: gzip, deflate
```

```
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
```

```
Connection: keep-alive
```

```
If-Modified-Since: Tue, 20 Sep 2011 08:58:31 GMT
```

```
If-None-Match: "f90d3d-63b6-4e7855b7;4dd6141f"
```

Estructura de una respuesta

Línea de estado

```
HTTP/1.1 200 OK
```

Cabeceras

```
Date: Tue, 20 Sep 2011 09:08:58 GMT
```

```
Server: Apache/1.3.41
```

```
Content-Location: index.html.es-es
```

```
Vary: negotiate,accept-language
```

```
TCN: choice
```

```
Last-Modified: Tue, 20 Sep 2011 08:58:31 GMT
```

```
Etag: "f90d3d-63b6-4e7855b7;4dd6141f"
```

```
Accept-Ranges: bytes
```

```
Content-Length: 25526
```

```
Keep-Alive: timeout=15, max=100
```

```
Connection: Keep-Alive
```

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

```
Content-Language: es-es
```

(línea en blanco...)

Cuerpo entidad

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="es "
```

```
xml:lang="es ">
```

```
<head>
```

```
<title>Universidad de Alicante</title>
```

```
...
```



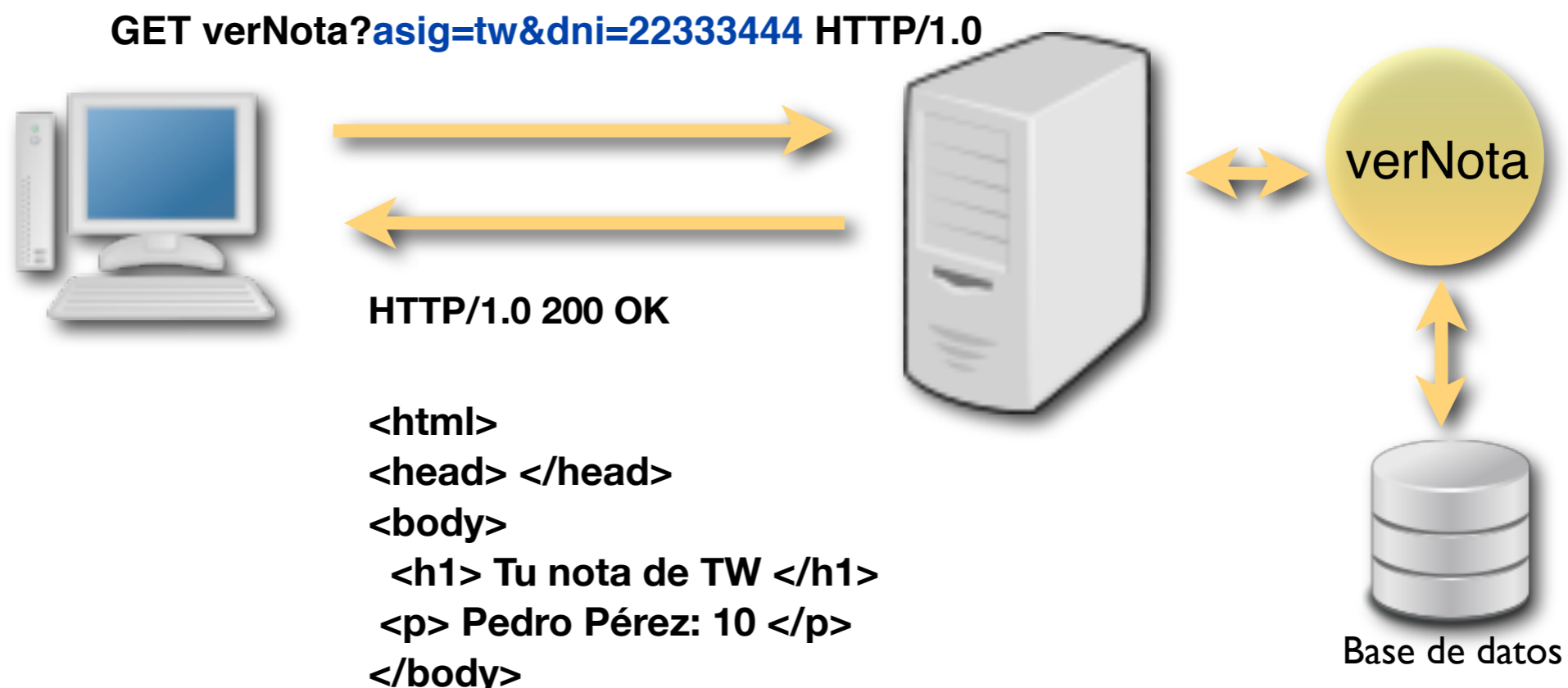

1.2 Aplicaciones web

Las aplicaciones web son programas remotos que se comunican mediante el protocolo HTTP

HTTP y aplicaciones web

Aplicaciones web

- Una aplicación web es una colección de “rutinas” o “subprogramas”. A cada uno se accede a través de una URL
- La comunicación con la aplicación se hace siguiendo la “sintaxis” de HTTP
- Al igual que en línea de comandos podemos pasar parámetros. En HTTP se pasan en la primera línea de la petición o bien al final



¿En qué se programan las aplicaciones web?

- Hay que distinguir entre programación en el lado del **servidor** y programación en el lado del **cliente**
- En el **lado del servidor**:
 - Necesitamos un lenguaje en el que sea sencillo
 - ◆ Parsear la petición y los parámetros HTTP
 - ◆ Trabajar con bases de datos
 - ◆ Conectar con otros servidores
 - ◆ Generar respuestas en formato HTTP
 - En principio, valdría cualquier lenguaje (¡hasta C!), siempre que tengamos las librerías adecuadas
 - ◆ Ejemplo: **ruby** es un lenguaje orientado a objetos que nació sin tener nada que ver con la web. Hoy es un lenguaje “de moda” en aplicaciones web gracias a “**ruby on rails**”, un conjunto de librerías para trabajar con HTTP y desarrollar aplicaciones web
 - ◆ PHP, .NET, Java, Python, ...

- Hay que distinguir entre programación en el lado del servidor y programación **en el lado del cliente**
- En el lado del cliente:
 - Necesitamos un lenguaje en el que sea sencillo
 - ◆ Programar una interfaz “rica” y “dinámica”
 - ◆ Hacer cálculos o procesamiento local
 - ◆ Pedirle más datos al servidor si es necesario
 - Como vamos a programar en el navegador, necesitamos que éste tenga un compilador o un intérprete del lenguaje
 - ◆ Todos los navegadores tienen un intérprete de Javascript
 - ◆ Se pueden añadir otros mediante plugins: Flash