

Aplicaciones

Juan F. Pérez

Departamento MACC
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario

juanferna.perez@urosario.edu.co

Primer Semestre de 2019

Contenidos

1 Aplicaciones y Bases de datos

2 Aplicaciones web

- Servidor web

Aplicaciones y Bases de datos

Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones

Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones
- Aplicación ofrece una interfaz (gráfica) de usuario

Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones
- Aplicación ofrece una interfaz (gráfica) de usuario
- Front-end: interfaz de usuario

Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones
- Aplicación ofrece una interfaz (gráfica) de usuario
- Front-end: interfaz de usuario
- Back-end: comunicación con base de datos

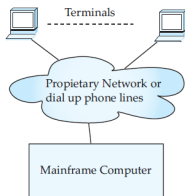
Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones
- Aplicación ofrece una interfaz (gráfica) de usuario
- Front-end: interfaz de usuario
- Back-end: comunicación con base de datos
- Middle layer: lógica de negocio que define qué acciones se realizan y cómo (front-end y back-end)

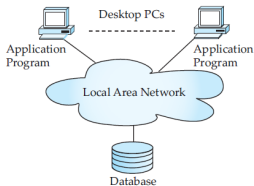
Aplicaciones y Bases de datos

- Usuarios interactúan con BDs a través de aplicaciones
- Aplicación ofrece una interfaz (gráfica) de usuario
- Front-end: interfaz de usuario
- Back-end: comunicación con base de datos
- Middle layer: lógica de negocio que define qué acciones se realizan y cómo (front-end y back-end)
- Ejemplos

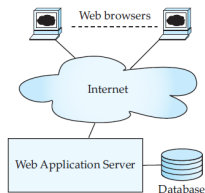
Aplicaciones y Bases de datos



(a) Mainframe Era



(b) Personal Computer Era



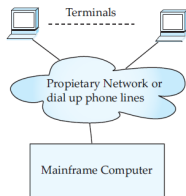
(c) Web era

■ Mainframes y terminales

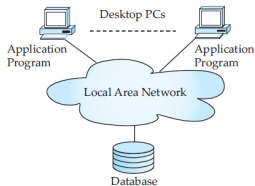
1

¹Tomado de Database System Concepts

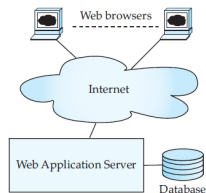
Aplicaciones y Bases de datos



(a) Mainframe Era



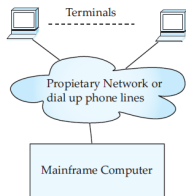
(b) Personal Computer Era



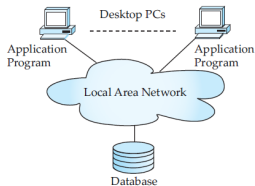
(c) Web era

- Mainframes y terminales
- Servidor - cliente

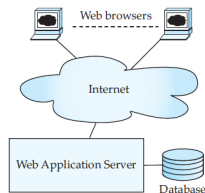
Aplicaciones y Bases de datos



(a) Mainframe Era



(b) Personal Computer Era



(c) Web era

- Mainframes y terminales
- Servidor - cliente
- Servidor - cliente web

1

¹Tomado de Database System Concepts

Aplicaciones web

Aplicaciones web

- Cliente: navegador (browser)

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)
- Texto con formato

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)
- Texto con formato
- Formularios

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)
- Texto con formato
- Formularios
- No requiere software adicional en el cliente

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)
- Texto con formato
- Formularios
- No requiere software adicional en el cliente
- JavaScript: mayor interacción con el usuario

Aplicaciones web

- Cliente: navegador (browser)
- Sintaxis: HyperText markup Language (HTML)
- Texto con formato
- Formularios
- No requiere software adicional en el cliente
- JavaScript: mayor interacción con el usuario
- Seguro y sin requerimientos adicionales para el usuario final

Aplicaciones web

- Servidor: implementa la lógica de negocio

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets
- Java Server Pages (JSP)

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets
- Java Server Pages (JSP)
- PHP

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets
- Java Server Pages (JSP)
- PHP
- Perl

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets
- Java Server Pages (JSP)
- PHP
- Perl
- Python

Aplicaciones web

- Servidor: implementa la lógica de negocio
- Comunica con la base de datos
- Java servlets
- Java Server Pages (JSP)
- PHP
- Perl
- Python
- JavaScript

Aplicaciones web

- URL: uniform resource locator

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`
- Nombre único para cada documento en la web

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`
- Nombre único para cada documento en la web
- HTTP: HyperText Transfer Protocol

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`
- Nombre único para cada documento en la web
- HTTP: HyperText Transfer Protocol
- Protocolo para transmitir archivos

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`
- Nombre único para cada documento en la web
- HTTP: HyperText Transfer Protocol
- Protocolo para transmitir archivos
- `www.urosario.edu.co/`: nombre de una máquina que ejecuta un servidor Web

Aplicaciones web

- URL: uniform resource locator
- `http://www.urosario.edu.co/`
- Nombre único para cada documento en la web
- HTTP: HyperText Transfer Protocol
- Protocolo para transmitir archivos
- `www.urosario.edu.co/`: nombre de una máquina que ejecuta un servidor Web
- `https://www.google.com/search?q=urosario` argumentos adicionales

HTML

```
<html>
<body>
<table border>
<tr> <th>ID</th> <th>Nombre </th> <th>Unidad</th> </tr>
<tr> <td>00128</td> <td>Perez</td> <td>MACC</td> </tr>
<tr> <td>12345</td> <td>Ramirez</td> <td>ECON</td> </tr>
<tr> <td>19991</td> <td>Gomez</td> <td>EMCS</td> </tr>
</table>
</body>
</html>
```

HTML formularios

```
<html>
<body>
<form action="consultaPersona" method=get>
Buscar:
<select name="tipoPersona">
<option value="estudiante" selected>Estudiante </option>
<option value="instructor"> Instructor </option>
</select> <br>
Nombre: <input type=text size=20 name="nombre">
<input type=submit value="Enviar">
</form>
</body>
</html>
```

HTML formularios

- `<form action="consultaPersona" method=get>`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual
- `<select name="tipoPersona">`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual
- `<select name="tipoPersona">`
- `<input type=text size=20 name="nombre">`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual
- `<select name="tipoPersona">`
- `<input type=text size=20 name="nombre">`
- Argumentos *tipoPersona* y *nombre* que usa el programa que se ejecuta al llamar el URL `consultaPersona`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual
- `<select name="tipoPersona">`
- `<input type=text size=20 name="nombre">`
- Argumentos *tipoPersona* y *nombre* que usa el programa que se ejecuta al llamar el URL `consultaPersona`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Se envía la solicitud al URL `consultaPersona` relativo a la página actual
- `<select name="tipoPersona">`
- `<input type=text size=20 name="nombre">`
- Argumentos *tipoPersona* y *nombre* que usa el programa que se ejecuta al llamar el URL `consultaPersona`
- El cliente espera un HTML como respuesta

HTML formularios

- `<form action="consultaPersona" method=get>`

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET
- Especificado en el protocolo HTTP

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET
- Especificado en el protocolo HTTP
- Argumentos en el URL

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET
- Especificado en el protocolo HTTP
- Argumentos en el URL
- Quedan en la historia, pueden almacenarse en caché

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET
- Especificado en el protocolo HTTP
- Argumentos en el URL
- Quedan en la historia, pueden almacenarse en caché
- No debe usarse para información delicada

HTML formularios

- `<form action="consultaPersona" method=get>`
- Método GET
- Especificado en el protocolo HTTP
- Argumentos en el URL
- Quedan en la historia, pueden almacenarse en caché
- No debe usarse para información delicada
- `.../consultaPersona?tipoPersona=estudiante&nombre=juan`

HTML formularios

- `<form action="consultaPersona" method=post>`

HTML formularios

- `<form action="consultaPersona" method=post>`
- Método POST

HTML formularios

- `<form action="consultaPersona" method=post>`
- Método POST
- Especificado en el protocolo HTTP

HTML formularios

- `<form action="consultaPersona" method=post>`
- Método POST
- Especificado en el protocolo HTTP
- Argumentos en la consulta

HTML formularios

- `<form action="consultaPersona" method=post>`
- Método POST
- Especificado en el protocolo HTTP
- Argumentos en la consulta
- No quedan en la historia, no se almacenan en caché

HTML formularios

- `<form action="consultaPersona" method=post>`
- Método POST
- Especificado en el protocolo HTTP
- Argumentos en la consulta
- No quedan en la historia, no se almacenan en caché
-

```
POST /consultaPersona HTTP/1.1
Host: www.juanfperez.com
tipoPersona=estudiante&nombre=juan
```

Servidor web

Servidor web

- Servidor web

Servidor web

- Servidor web
- Transferencia de documentos HTML

Servidor web

- Servidor web
- Transferencia de documentos HTML
- Comunicación con navegadores usando el protocolo HTTP

Servidor web

- Servidor web
- Transferencia de documentos HTML
- Comunicación con navegadores usando el protocolo HTTP
- Ejecuta programas/aplicaciones en el servidor y retorna los resultados al cliente en forma de HTML

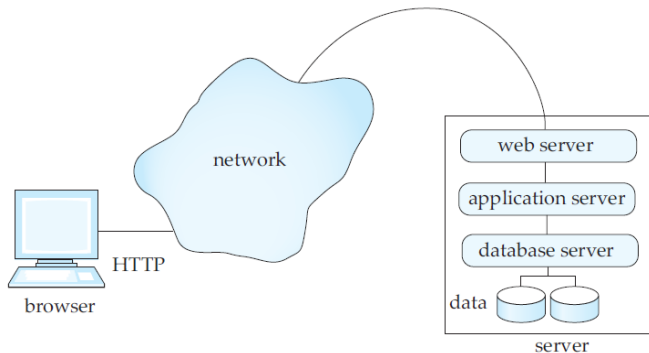
Servidor web

- Servidor web
- Transferencia de documentos HTML
- Comunicación con navegadores usando el protocolo HTTP
- Ejecuta programas/aplicaciones en el servidor y retorna los resultados al cliente en forma de HTML
- Comunicación entre servidor web y aplicaciones: common gateway interface (CGI)

Servidor web

- Servidor web
- Transferencia de documentos HTML
- Comunicación con navegadores usando el protocolo HTTP
- Ejecuta programas/aplicaciones en el servidor y retorna los resultados al cliente en forma de HTML
- Comunicación entre servidor web y aplicaciones: common gateway interface (CGI)
- La aplicación se comunica con la base de datos: ODBC, JDBC

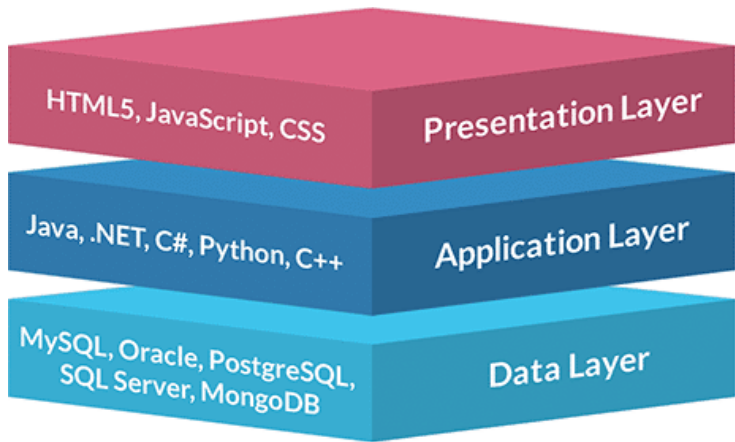
Aplicaciones de 3 capas



2

²Tomado de Database System Concepts

Aplicaciones de 3 capas

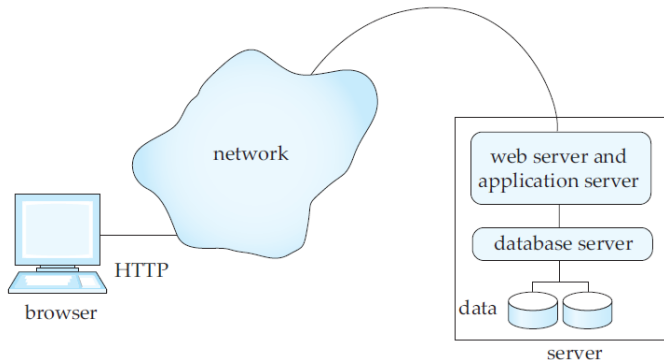


3

³Tomado de

<https://www.jinfony.com/resources/bi-defined/3-tier-architecture-complete-overview/>

Aplicaciones de 2 capas



Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)
- HTTP: sin conexión (connectionless)

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)
- HTTP: sin conexión (connectionless)
- Al llegar una consulta se crea una conexión temporal, y al enviar la respuesta se destruye

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)
- HTTP: sin conexión (connectionless)
- Al llegar una consulta se crea una conexión temporal, y al enviar la respuesta se destruye
- Nueva consulta del mismo cliente: nueva conexión

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)
- HTTP: sin conexión (connectionless)
- Al llegar una consulta se crea una conexión temporal, y al enviar la respuesta se destruye
- Nueva consulta del mismo cliente: nueva conexión
- Permite atender mayor número de clientes

Conecciones

- Al loguearse en el OS o al conectarse a una DB, se crea una sesión
- Sesión permite reconocer al usuario hasta que se desconecte (conexión-desconexión)
- HTTP: sin conexión (connectionless)
- Al llegar una consulta se crea una conexión temporal, y al enviar la respuesta se destruye
- Nueva consulta del mismo cliente: nueva conexión
- Permite atender mayor número de clientes
- Algunas aplicaciones requieren mantener información del cliente

Conecciones

- Información adicional para identificar a los usuarios

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web
- Preferencias de configuración

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web
- Preferencias de configuración
- Definido por el servidor

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web
- Preferencias de configuración
- Definido por el servidor
- ID de sesión (en cookie y en servidor)

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web
- Preferencias de configuración
- Definido por el servidor
- ID de sesión (en cookie y en servidor)
- Autenticación

Conecciones

- Información adicional para identificar a los usuarios
- Cookies: información almacenada por el cliente para identificarse con cada servidor web
- Preferencias de configuración
- Definido por el servidor
- ID de sesión (en cookie y en servidor)
- Autenticación
- Timeout: cookie inválida después de cierto tiempo (lista de sesiones válidas)