

embarcadero®

Delphi Academy

Consejos prácticos, trucos y técnicas



Desarrollo WEB

Parte #1

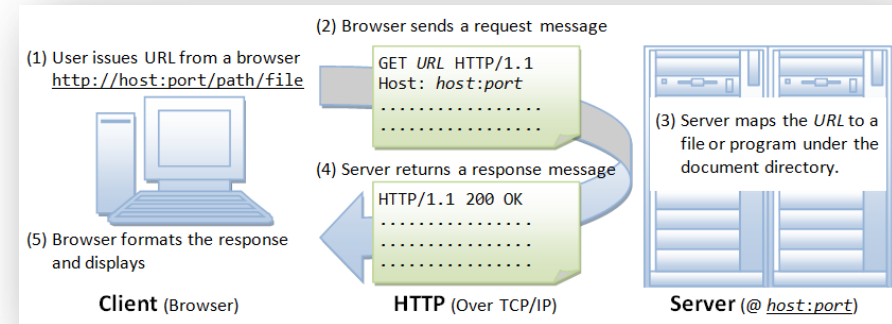
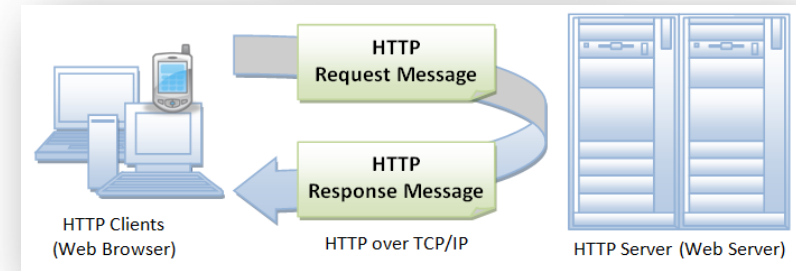
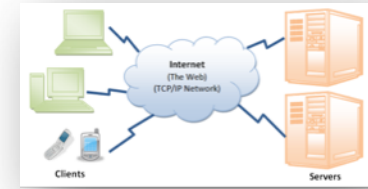
Fernando Rizzato
Lead Software Consultant, *Latin America*

AGENDA

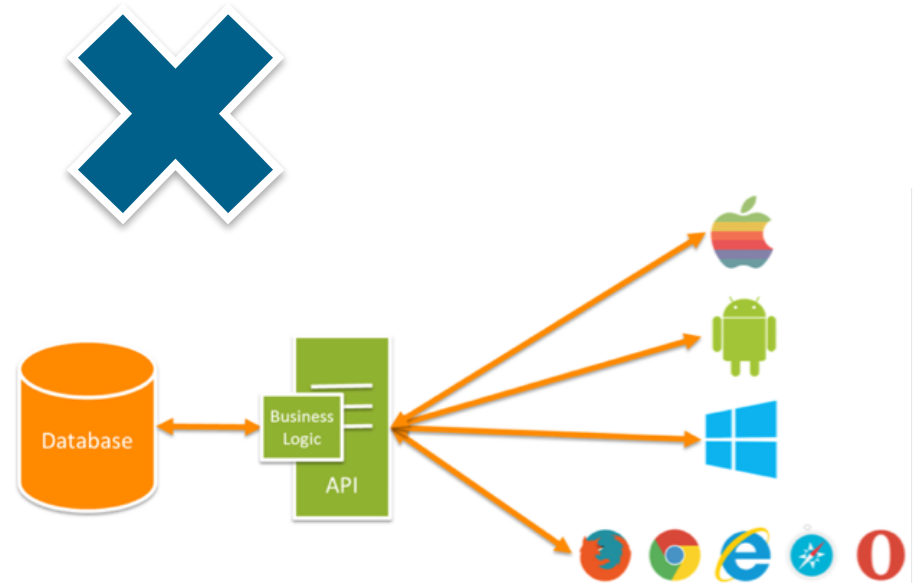
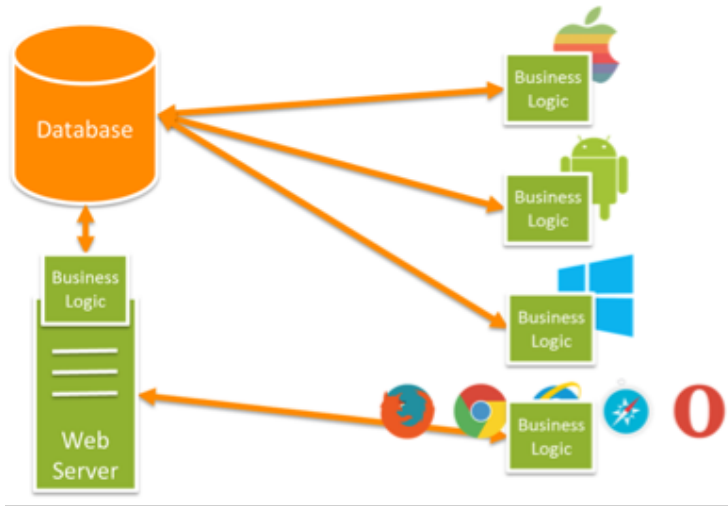
- Conceptos
 - Comunicación HTTP
 - Aplicaciones Desktop frente a Aplicaciones WEB
 - Browser: HTML, CSS, JavaScript
- Arquitecturas y Posibilidades
 - *Stateful* versus *Stateless*
 - Web Server y App Server
- Demos
 - #1: WebBroker
 - #2: IW + BootStrap
 - #3: RAD Server + Ext JS

COMUNICACIÓN HTTP

- *Hypertext Transfer Protocol* (HTTP) es un protocolo de aplicación para sistemas de información distribuidos
- HTTP es la base de la comunicación de datos para los sistemas disponibles en Internet



APLICACIONES DESKTOP FRENTE A APLICACIONES WEB



BROWSER: HTML, CSS, JAVASCRIPT

- **HTML** significa *Hyper Text Markup Language*
 - HTML describe la estructura de las páginas Web mediante marcadores
 - Los elementos HTML son los bloques de construcción de páginas HTML
 - Los elementos HTML se representan por tags
 - Las tags HTML etiquetan fragmentos de contenido como "encabezado", "párrafo", "tabla", etc.
 - Los navegadores no muestran las tags HTML, pero las usan para representar el contenido de la página

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

BROWSER: HTML, CSS, JAVASCRIPT

- **CSS** significa *Cascading Style Sheets*
 - El CSS describe cómo deben aparecer los elementos HTML en la pantalla, papel o en otros medios
 - El CSS ahorra mucho trabajo. Puede controlar el diseño de varias páginas Web a la vez
 - Los conjuntos de estilo externo se almacenan en archivos CSS

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

BROWSER: HTML, CSS, JAVASCRIPT

- **JavaScript** (o simplemente JS) es un lenguaje de programación de alto nivel, dinámico, débilmente tipado, multi-paradigma e interpretado.
 - Al lado de HTML y CSS, JavaScript es una de las tres principales tecnologías para la producción de contenido de la *World Wide Web*.
 - Se utiliza para hacer las páginas interactivas. La mayoría de los sitios web lo emplea, y todos los navegadores web modernos lo soportan sin la necesidad de *plug-ins*, a través de un motor de JavaScript incorporado.

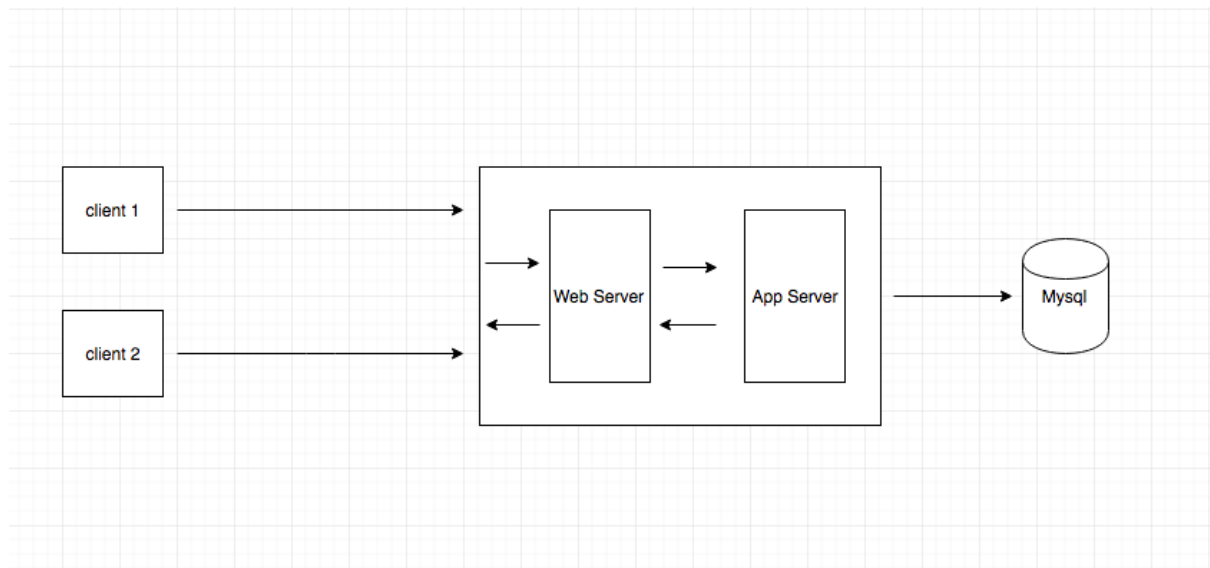
```
<!DOCTYPE html>
<html><head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML =
    "Paragraph changed.";
}
</script>
</head>
<body>
<h1>A Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try
it</button>
</body>
</html>
```

STATEFUL VERSUS STATELESS

- **Stateful** - mantiene el control del estado de las interacciones, generalmente almacenando valores en una ubicación designada
 - A favor: más fácil de desarrollar
 - Contra: mayor carga para el servidor
- **Stateless** – cuando no hay registro de las interacciones anteriores, y cada solicitud debe ser tratado enteramente basado en información que viene con el mismo
 - A favor: más fácil de escalar
 - Contra: más difícil de desarrollar

WEB SERVER Y APP SERVER

- **Web Server** está diseñado para servir contenido HTTP
- **App Server** también puede servir contenido HTTP, pero no está limitado a este, soportando otros protocolos y estándares



ARQUITECTURAS Y POSIBILIDADES

- WebBroker² + Bootstrap e/ou JS¹ (+ DataSnap²)
- IntraWeb³ + Bootstrap e/ou JS¹ (+ DataSnap²)
- uniGui/Ext JS³ (+ DataSnap²)
- JS¹ + DataSnap² (REST)
- JS¹ + RAD Server² (REST)
- JS¹ + Delphi MVC Framework² (REST)

(1) JQuery, AngularJS, Ext JS, ...

(2) Stand-alone, Windows service, IIS o Apache (Windows y Linux)

(3) Stand-alone, Windows service, IIS (Windows, Linux en los planes)

DEMOS

RECURSOS ADICIONALES

■ Documentación

- http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Creating_Internet_server_applications_Index
- http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Creating_WebBroker_Applications
- http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Using_Web_Broker_Index
- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/css/default.asp>
- <https://www.w3schools.com/js/default.asp>

■ Blogs

- <https://spring.io/guides/gs/consuming-rest-jquery/>
- <https://www.nginx.com/resources/glossary/application-server-vs-web-server/>
- <https://www.oyewiki.com/tech/web-server-vs-application-server-difference-explained>
- <http://etutorials.org/Programming/mastering+delphi+7/Part+IV+Delphi+the+Internet+and+a+.NET+Preview/Chapter+20+Web+Programming+with+WebBroker+and+WebSnap/Delphi+s+WebBroker+Technology/>
- <https://community.embarcadero.com/blogs/entry/join-marco-cantu-for-web-razor-scripting-with-webbroker>
- <https://community.embarcadero.com/blogs/entry/how-to-make-restful-webbroker-using-c-builder>

GRACIAS!

Preguntas?

Me puedes encontrar en:

@FernandoRizzato

fernando.rizzato@embarcadero.com

Síguenos en

fb.com/EMBTLatAm