
Introducció a la programació web avançada

PID_00172684

Jordi Sánchez Cano



Llicència Creative Commons, versió 3.0, modalitat BY-SA (attribution - share alike), que permet modificar l'obra, reproduir-la, distribuir-la o comunicar-la públicament sempre que se'n reconegui l'autoria i sempre que l'obra derivada quedi subjecta a la mateixa llicència que el material original.

Índex

Introducció	5
Objectius	6
1. Evolució de la programació web	7
1.1. Introducció a l'arquitectura client-servidor	7
1.2. Pàgines web	8
1.2.1. Pàgines estàtiques	8
1.2.2. Pàgines dinàmiques	8
1.3. Aplicacions web	9
1.4. Evolució del Web	10
2. Tecnologies i llenguatges de client	12
2.1. JavaScript	12
2.2. Miniaplicacions de Java	12
2.3. Adobe Flash	14
2.4. ActiveX	14
3. Tecnologies i llenguatges de servidor	16
3.1. CGI	16
3.2. ISAPI	18
3.3. ASP	18
3.4. PHP	19
3.5. Java EE	20
3.6. ASP.NET	21
3.7. Ruby on Rails	22
4. RIA	24
4.1. AJAX	24
4.2. Adobe Flex	25
4.3. JavaFX	25
4.4. Silverlight	25

Introducció

En aquest mòdul es fa una introducció a les tecnologies i als llenguatges relacionats amb el Web des dels seus inicis fins a l'actualitat.

Inicialment es descriuen les diferències entre pàgines tradicionals i pàgines dinàmiques. Fet això, s'explica què són les aplicacions web i els seus avantatges sobre les aplicacions d'escriptori.

Posteriorment s'expliquen les principals tecnologies i els llenguatges de client i servidor d'una manera cronològica. Finalment, s'introdueix la definició d'*aplicacions d'Internet enriquides*, RIA (*rich Internet applications*), i de les tecnologies relacionades més populars.

Objectius

Amb l'estudi d'aquest mòdul, assolireu els objectius següents:

- 1.** Entendre la diferència entre pàgines estàtiques i pàgines dinàmiques.
- 2.** Conèixer què són les aplicacions web i RIA.
- 3.** Veure l'evolució de les tecnologies i dels llenguatges de servidor i client en entorn web.

1. Evolució de la programació web

Aquest mòdul se centra en les aplicacions web que són per naturalesa client-servidor. A continuació s'expliquen els conceptes principals sobre arquitectura client-servidor, pàgines i aplicacions web.

1.1. Introducció a l'arquitectura client-servidor

Una arquitectura client-servidor és composta per diferents equips que formen part d'una mateixa xarxa en què uns proporcionen serveis a d'altres.

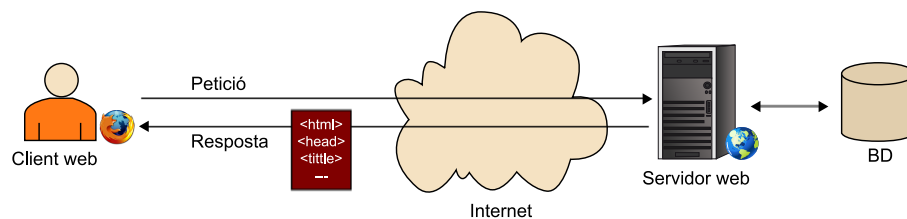
L'ús de serveis es fa mitjançant l'intercanvi de peticions, respostes i de dades. Els equips que sol·liciten els serveis són anomenats **equips client** i els que proporcionen el servei es denominen **equips servidor**.

Els equips proporcionen serveis mitjançant l'execució d'aplicacions que s'encarreguen de rebre i processar les diferents sol·licituds i respondre-hi. Aquestes aplicacions es denominen també *servidor*, com, per exemple, els servidors de correu, servidors web i servidors d'arxius. Un mateix dispositiu pot oferir diversos serveis segons les aplicacions que estigui executant.

Les aplicacions encarregades de fer ús dels serveis es denominen **clients**, com, per exemple, els clients de correu o els clients FTP¹. Els navegadors web són un altre exemple de clients web.

⁽¹⁾ FTP és la sigla de *file transfer protocol*, 'protocol de transferència de fitxers'.

Figura 1. Peticions de pàgines per navegadors web



En la figura 1 es mostra un exemple de client-servidor web. Els clients envien peticions al servidor i aquest les processa i envia pàgines web de tornada.

L'arquitectura d'aplicacions client-servidor presenta avantatges com els següents:

- Els recursos estan centralitzats en un o diversos servidors, cosa que permet la unificació i la integritat de les dades, el control dels diferents accessos de clients, etc.

- Aquesta arquitectura permet l'escalabilitat del servei, i també l'execució en paral·lel. Els dispositius físics poden ser ampliat o reemplaçats per altres que ofereixin un rendiment millor. També és possible ampliar el nombre de servidors d'un mateix servei que permeti el procés d'una mateixa petició en paral·lel, en què cada servidor duria a terme una part de la tasca.
- L'administració és més senzilla, ja que se centra en la part servidor.

1.2. Pàgines web

Una pàgina web és un document escrit en un llenguatge de marques que permet representar informació principalment textual. La característica principal d'una pàgina són els enllaços o els hiperenllaços, que permeten la navegació d'un document a l'altre.

Les pàgines es poden classificar en estàtiques o dinàmiques, segons el comportament que presenten des que se les sol·licita fins que estan disponibles al navegador.

1.2.1. Pàgines estàtiques

Les pàgines web estàtiques són les que no ofereixen cap interacció ni modificació del seu contingut durant tot el procés de sol·licitud i presentació al navegador: el mateix contingut que es troba en el servidor s'envia sense cap modificació.

Una vegada que el navegador mostra la pàgina, aquesta no fa cap acció ni cap resposta a esdeveniments.

La tecnologia necessària per a mostrar pàgines estàtiques és un servidor HTTP i un navegador web. El navegador sol·licita les pàgines mitjançant peticions HTTP i el servidor torna els arxius corresponents.

1.2.2. Pàgines dinàmiques

Les pàgines dinàmiques, a diferència de les estàtiques, ofereixen dinamisme, ja sigui al servidor en sol·licitar la pàgina o bé al navegador del client.

El dinamisme en el costat del servidor permet generar parts HTML d'un document d'una manera dinàmica segons els paràmetres de sol·licitud, el context de l'usuari, el navegador que s'estigui utilitzant, etc.

Llenguatge de marques

El llenguatge de marques més estès és l'HTML (*hypertext markup language*).

Imatges en pàgines web

Les pàgines HTML també poden mostrar imatges i objectes incrustats que dependran de la implementació del navegador o complements instal·lats.

Primeres pàgines web

Les primeres pàgines web eren estàtiques i només presentaven documents amb enllaços o *links* cap a altres pàgines.

HTTP

HTTP és la sigla d'*hypertext transfer protocol*. Es tracta d'un protocol orientat a transaccions per a sol·licitar, entre altres operacions, recursos allotjats en un servidor. Els servidors HTTP també es denominen **servidors web**.

El dinamisme generat durant la representació de la pàgina al navegador es fa mitjançant llenguatges interpretats com JavaScript, i mitjançant el DOM². Aquestes pàgines poden respondre a diferents esdeveniments: càrrega inicial de la pàgina, clic sobre un botó, moure el ratolí sobre una àrea, etc. També poden efectuar modificacions sobre el mateix document sense actualitzar la pàgina amb una petició nova: ocultar una taula, mostrar una imatge, afegir elements a una llista, etc.

Amb l'evolució de les tecnologies de servidor i de client sorgeix un nou concepte d'aplicació centralitzada a partir del Web que s'explica a continuació.

1.3. Aplicacions web

Una aplicació web és una aplicació resident en un servidor remot accessible a partir d'una aplicació client, generalment un navegador.

Amb l'evolució i l'aparició de diferents tecnologies, protocols i estàndards tant de client com de servidor, els desenvolupadors poden optar per un model d'aplicació centralitzada en què la lògica de negoci i les dades són en servidors remots. Les interfícies capten les accions i les dades introduïdes pels usuaris i les envien al servidor a l'espera d'una resposta. Així mateix, els clients reben interfícies en forma de pàgines web amb dades processades des del servidor.

En els últims anys les aplicacions web s'han anat estenent per Internet oferint diferents serveis i comerços: correu electrònic, botigues en línia, blogs, etc.

Les aplicacions web ofereixen alguns avantatges respecte a les aplicacions d'escriptori, com, per exemple:

- **Compatibilitat i multiplataforma:** les aplicacions web envien interfícies que utilitzen estàndards compatibles amb els navegadors més populars: HTML, JavaScript, CSS³, etc.
- **Fàcils de distribuir i modificar:** tota l'aplicació està allotjada en un servidor i no en els terminals client, per la qual cosa les modificacions fetes es reflectiran en els accessos posteriors.

⁽²⁾DOM és la sigla de *document object model*.

DOM

DOM és una API (*application programming interface*, interfície de programació d'aplicacions) que permet als programes i *scripts* accedir al contingut d'un document d'una manera dinàmica i manipular-lo.

Navegadors

Els navegadors web es consideren clients lleugers, ja que la major part del procés es duu a terme al servidor.

⁽³⁾CSS és la sigla de *cascading style sheets*, 'fulls d'estil en cascada'.

- **Portabilitat:** les dades dels usuaris són generalment en el servidor, per la qual cosa es podrà utilitzar l'aplicació des de qualsevol terminal amb un navegador compatible.

Les aplicacions web també presenten alguns inconvenients:

- No ofereixen interfícies ni funcionalitats tan riques com les aplicacions d'escriptori a causa de les limitacions de les tecnologies del costat de client. Tot i així, l'evolució dels estàndards i l'aparició d'altres fa que en l'actualitat es disposi d'aplicacions difícils de distingir entre web i escriptori.
- Com que l'aplicació es distribueix per mitjà de la Xarxa, la disponibilitat, resposta i experiència de l'usuari dependran de la seva velocitat.

Complements

Algunes tecnologies de servidor utilitzen mecanismes propis que requereixen la instal·lació de connectors (*plugins*) al navegador. Un exemple molt estès podria ser Flash Player o miniaplicacions (*applets*) de Java.

Evolució d'estàndards

La nova especificació d'HTML versió 5 permet la reproducció d'àudio i vídeo sense necessitat d'incorporar complements addicionals com Flash Player.

1.4. Evolució del Web

L'evolució web ha estat impulsada tant per canvis tecnològics com socials. En els seus inicis, el denominat *Web 1.0* es basava en la lectura de continguts pels clients i en la publicació d'aquests continguts pels administradors dels llocs web. Per tant, el Web 1.0, des d'un punt de vista dels usuaris, era un mitjà només informatiu.

Amb l'aparició de noves tecnologies i tendències d'ús sorgeix el terme *Web 2.0*, associat al Web orientat a la interacció i xarxes socials en què la participació dels usuaris és fonamental per a la incorporació de continguts nous.

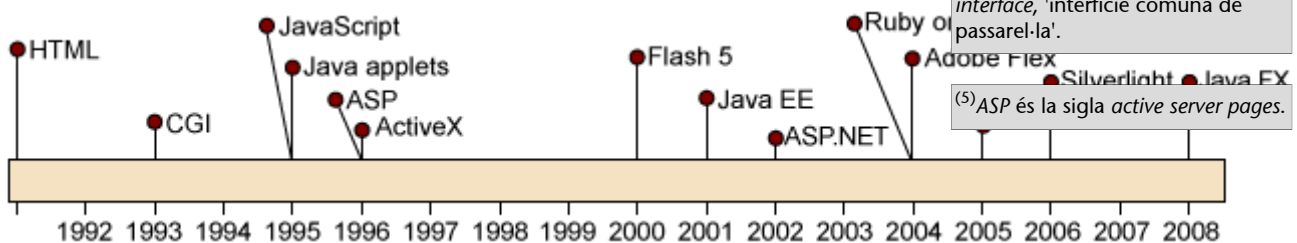
Per tant, en el Web 2.0 els usuaris també són participants dels continguts com en fòrums de discussió, wikis, blogs, etc.

La figura següent mostra un cronograma amb l'aparició de les tecnologies web més importants.

Wikis

Les wikis són pàgines web on la informació pot ser editada pels internautes mitjançant el seu navegador.

Figura 2. Principals tecnologies web



⁽⁴⁾ CGI és la sigla *common gateway interface*, 'interfície comuna de passarel·la'.

⁽⁵⁾ ASP és la sigla *active server pages*.

Les pàgines estàtiques apareixen amb el llenguatge HTML i, encara que posteriorment apareixen les pàgines dinàmiques amb CGI⁴, JavaScript, ASP⁵, etc., aquestes continuen coexistint d'una manera paral·lela.

Algunes d'aquestes tecnologies s'executen al navegador i d'altres tant en client com en servidor. En els apartats següents s'expliquen cadascuna d'elles.

2. Tecnologies i llenguatges de client

A continuació, s'expliquen algunes tecnologies que s'executen en el client.

2.1. JavaScript

JavaScript és un llenguatge de *script* creat per Netscape Communications que es va incloure en el seu navegador Netscape Navigator 2.0 el 1995. El 1997 JavaScript va ser acceptat com a estàndard ECMA amb el nom ECMAScript. Posteriorment, va ser declarat estàndard per l'ISO.

En l'actualitat, tots els navegadors incorporen una implementació de JavaScript i permeten executar accions en pàgines HTML per a dotar-les de dinamisme i interactivitat. Alguns usos poden ser:

- Respondre a accions de l'usuari sobre una pàgina, com ara clics, moure el punter del ratolí, prémer tecles, etc.
- Permet comprovar els camps de formularis abans que siguin enviats al servidor. Això ajuda a detectar entrades de dades invàlides en camps amb un format concret. Per exemple, un camp que sol·liciti l'edat de l'usuari no ha de permetre caràcters alfabètics.
- Amb l'ús conjunt d'altres tecnologies, permet dur a terme peticions asíncrones i actualitzar porcions de pàgines sense necessitat de carregar la pàgina sencera. Això es coneix amb el nom AJAX i s'estudia en profunditat en altres mòduls.
- Modificar el contingut d'una pàgina d'una manera dinàmica. Per exemple, una funció JavaScript pot reestructurar una taula, modificar-ne el color de fons, canviar l'estil d'un paràgraf, etc. Per a modificar el contingut d'un document s'utilitza el DOM.

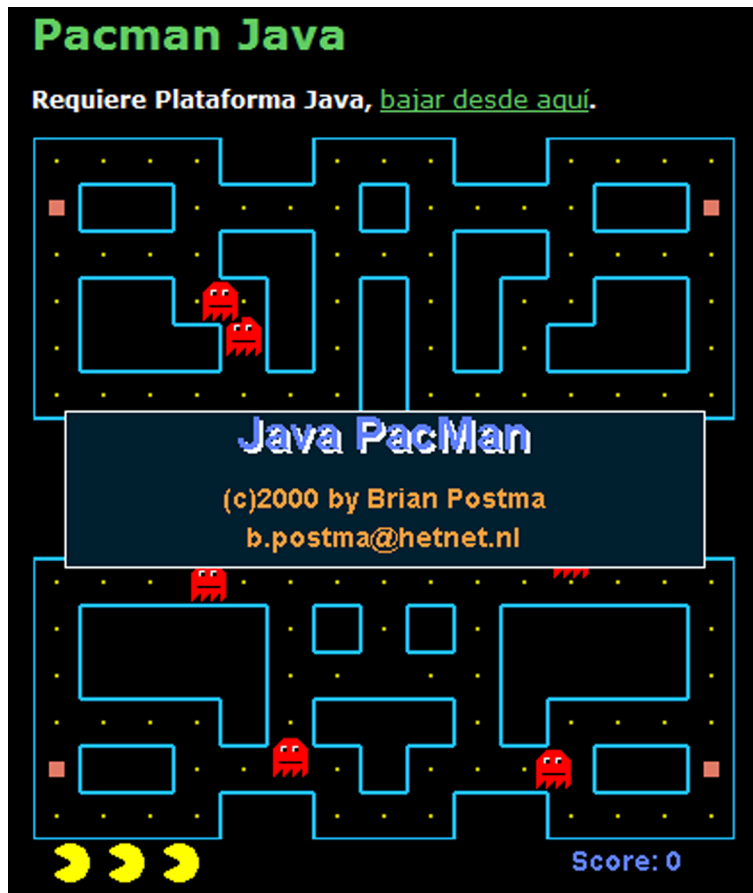
Origen de JavaScript

JavaScript es va denominar inicialment LiveScript en què *live* feia referència a 'donar vida' a les pàgines estàtiques. El desembre de 1995 va ser rebatejat com a JavaScript.

2.2. Miniaplicacions de Java

Les miniaplicacions de Java són programes desenvolupats en llenguatge Java que poden ser incorporats en una pàgina HTML com a objectes incrustats. Es descarreguen i s'executen al navegador del client mitjançant una màquina virtual.

Figura 3. Joc Pacman desenvolupat en una miniaplicació de Java

**Execució de les miniaplicacions**

Les miniaplicacions de Java s'executen mitjançant la màquina virtual de Java, la JVM (*Java virtual machine*). El navegador web requerirà tenir instal·lat un complement de la JVM per a executar-los.

Les miniaplicacions presenten gairebé tants inconvenients com avantatges i s'han de tenir en compte diverses consideracions abans d'elegir aquesta tecnologia.

1) Avantatges:

- Navegadors multiplataforma i múltiples. Hi ha entorns d'execució de Java per als sistemes operatius més utilitzats. Així mateix, els navegadors més estesos també disposen de complements per a l'execució de miniaplicacions en pàgines HTML.
- Permet fer aplicacions més sofisticades que JavaScript.
- L'execució d'una miniaplicació precompilada és més ràpida que JavaScript.

- Si s'atorguen els permisos necessaris al navegador, una miniaplicació pot tenir accés als recursos del sistema. Encara que en alguns contextos de seguretat, això podria ser un inconvenient.

2) Inconvenients:

- La inicialització d'una miniaplicació pot trigar en excés si la JVM no està en funcionament.
- Una miniaplicació depèn d'una versió de JRE⁶ específica. Si el navegador no disposa d'aquesta versió, caldrà actualitzar-la.
- La miniaplicació desenvolupa la seva execució dins del contenidor que l'encapsula sense tenir accés a la resta de la pàgina.

Entorns d'execució d'una miniaplicació

Una mateixa miniaplicació allotjada en una pàgina es pot executar en diferents navegadors com Firefox, Chrome, Safari, etc., i en diferents sistemes operatius com la família de Windows, Linux o Mac OS.

⁽⁶⁾JRE és la sigla de *Java runtime environment*, 'entorn d'execució de Java'.

2.3. Adobe Flash

Adobe Flash és una plataforma multimèdia per a afegir animacions i interacció en pàgines web. Va ser introduïda el 1996 i actualment és mantinguda i distribuïda per Adobe Systems.

Està basada en fotogrames que inclouen gràfics vectorials, imatges i fluxos de dades (*streams*) bidireccionals d'àudio i vídeo. La lògica es fa mitjançant un llenguatge de *script* i orientat a objectes anomenat *ActionScript*.

Les animacions es compilen en fitxers amb extensió SWF⁷ i s'inclouen a les pàgines com a objectes incrustats. L'execució es duu a terme al navegador client, que ha d'incorporar un complement Flash Player.

Ús principal de Flash

Flash s'utilitza generalment per a crear animacions, anuncis publicitaris, reproducció d'àudio i vídeo, etc.

⁽⁷⁾SWF és la sigla de *small web format*, 'petit format web'.

2.4. ActiveX

ActiveX és una tecnologia propietària de Microsoft per al desenvolupament de components reutilitzables entre diferents aplicacions. Va ser presentada el 1995 com el resultat de l'ús d'altres tecnologies de Microsoft ja existents.

Flash Player

Flash Player és un entorn basat en una màquina virtual que interpreta ActionScript i executa les animacions Flash. Hi ha complements per als navegadors web i en gran varietat de dispositius mòbils.

ActiveX ofereix una sèrie de mecanismes per a encapsular components de programari amb una interfície que permet incloure'ls dins d'altres aplicacions heterogènies. D'aquesta manera, una aplicació pot incloure components ActiveX sense importar el llenguatge en el qual estigui escrit.

En l'actualitat, hi ha un gran nombre d'aplicacions, moltes de les quals de Microsoft, que exposen funcionalitats com controls ActiveX, entre les quals hi ha Internet Explorer, Visual Studio i Windows Media Player.

Els components ActiveX poden ser distribuïts com a aplicacions per Internet per a mostrar elements complexos com ara animacions, calendaris, graelles de dades riques, documents Word o Excel, etc.

ActiveX enfront de les miniaplicacions de Java

Els controls ActiveX es poden comparar amb les miniaplicacions de Java. Les miniaplicacions es poden executar en una àmplia gamma de plataformes i dispositius, a diferència dels controls ActiveX, que depenen del navegador i de les plataformes de Microsoft.

3. Tecnologies i llenguatges de servidor

A continuació, veurem alguns exemples de tecnologies que s'executen en servidor.

3.1. CGI

CGI apareix l'any 1993 com una manera estàndard de sol·licitar i transferir dades a un programa allotjat en un servidor mitjançant HTTP. Aquests programes són comunament coneguts per CGI.

CGI⁸ va ser una de les primeres tecnologies que va permetre distribuir pàgines web dinàmiques des del costat del servidor.

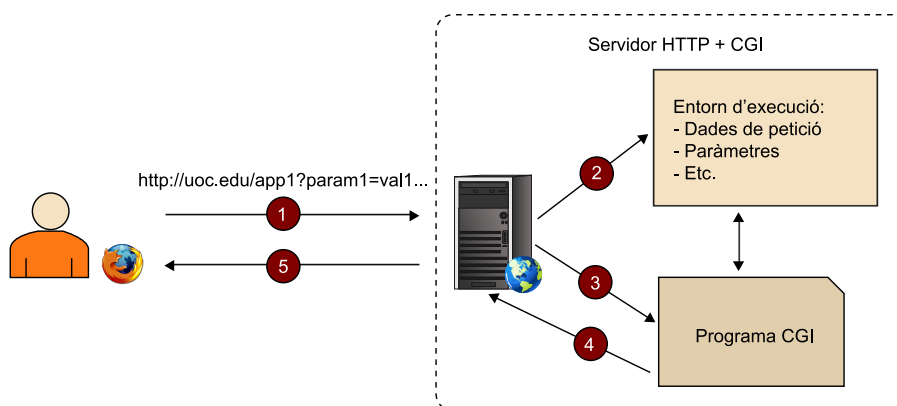
La primera tecnologia per a webs dinàmics

Les aplicacions CGI van ser les primeres solucions que permetien generar pàgines web amb continguts dinàmics.

⁽⁸⁾ CGI és la sigla de *common gateway interface*, 'interfície comuna de passarel·la'.

Els CGI poden estar escrits en qualsevol llenguatge com C, C++, Java o Perl. Permeten generar continguts dinàmics per la sortida del programa a partir de les dades d'entrada. Normalment generen pàgines HTML, per la qual cosa se solen utilitzar llenguatges amb un gran suport per a la manipulació de cadenes.

Figura 4. Funcionament bàsic d'una CGI



El funcionament bàsic d'una CGI és el següent:

- 1) Un client fa una petició HTTP a una aplicació CGI. Aquesta petició es pot fer a partir d'un Submit d'un formulari o un enllaç.
- 2) El servidor prepara l'entorn d'execució amb els paràmetres d'entrada obtinguts de la petició, el context d'execució de la CGI, etc.
- 3) El servidor executa el programa CGI proporcionant les dades d'entrada.

4) El programa s'executa escrivint codi HTML per la sortida del procés. Durant l'execució pot obtenir dades addicionals a partir de fitxers locals o una base de dades.

5) Finalment el servidor envia de tornada la resposta HTTP amb les dades de sortida de la CGI.

Un exemple de programa CGI escrit en C++ seria el següent:

```
#include "stdafx.h"
#include <iostream>
#include <cstdlib>

int _tmain(int argc, _TCHAR* argv[])
{
    /* La sortida següent indica al servidor el tipus
    d'información de sortida. En aquest cas, enviem
    text HTML.*/
    std::cout<<"Content-type: text/html\n\n";
    //S'envia la pàgina per la sortida del programa.
    std::cout<<"<html><head><title>Exemple de CGI</title></head><body>";
    //Dins de body, es mostra "Hola món!" set vegades
    entre etiquetes de capçalera mitjançant un bucle"
    for(int i = 7 ; i > 0; i--)
    {
        std::cout<<"<h"<<i<<">Hola món!</h"<<i<<">";
    }
    std::cout<<"</body></html>";
    return 0;
}
```

La sortida generada pel programa tindria aquest aspecte:

```
Content-type: text/html

<html>
  <head><title>Exemple de CGI</title></head>
  <body>
    <h7>Hola món!</h7>
    <h6>Hola món!</h6>
    <h5>Hola món!</h5>
    <h4>Hola món!</h4>
    <h3>Hola món!</h3>
    <h2>Hola món!</h2>
    <h1>Hola món!</h1>
  </body>
```

```
</html>
```

Un programa CGI pot fer qualsevol operació que permeti el llenguatge i l'entorn utilitzats dins del servidor, i es pot utilitzar per exemple per a consultar una base de dades, treballar amb arxius del servidor, etc.

D'altra banda, CGI presenta un inconvenient, ja que cada petició requereix l'execució d'un programa al servidor que finalitza en enviar la resposta. Això requereix una sobrecàrrega en efectuar la creació del procés i el seu context, la preparació de l'entorn d'execució i la seva destrucció. Aquestes tasques poden sobrecarregar el servidor o fins i tot saturar-lo en el pitjor dels casos.

3.2. ISAPI

Microsoft va crear la interfície ISAPI⁹ per resoldre el problema de sobrecàrrega respecte a la tecnologia CGI.

⁽⁹⁾ISAPI és la sigla d'*Internet server application programming interface*.

Bàsicament, ISAPI permet allotjar la funcionalitat dels programes en DLL¹⁰ en lloc d'executables.

⁽¹⁰⁾DLL és la sigla de *dynamic-link library*, 'biblioteca d'enllaços dinàmics'. Terme que es refereix a arxius amb codi font executable.

El codi d'una extensió ISAPI es carrega en memòria una sola vegada i permet l'execució del mateix codi per diferents peticions. Els paràmetres i el context d'execució, els gestiona el servidor d'aplicacions. Les aplicacions basades en ISAPI s'executen sota IIS¹¹.

⁽¹¹⁾IIS és la sigla d'*Internet information services*.

ISAPI també presenta alguns inconvenients, com ara la complexitat de desenvolupament i de depuració, que són resolts per futures tecnologies de servidor que s'expliquen en els subapartats següents.

3.3. ASP

ASP¹² és una tecnologia de Microsoft per a generar pàgines dinàmiques sota un servidor d'aplicacions IIS.

⁽¹²⁾ASP és la sigla d'*active server pages*, 'pàgines de servidor actives'.

IIS

Inicialment les pàgines ASP s'executaven dins del servidor d'aplicacions IIS. Actualment hi ha diversos servidors web que permeten servir pàgines ASP.

ASP ofereix l'avantatge de poder barrejar codi JavaScript o VBScript amb HTML al mateix document mitjançant marques que diferencien el codi incrustat. En sol·licitar-se una pàgina, el servidor interpreta les parts de codi VBScript i les substitueix per la sortida que generen. El resultat enviat als clients és una pàgina HTML.

VBScript

VBScript (*Visual Basic script*) és un llenguatge interpretat basat en Visual Basic.

Un exemple senzill de pàgina ASP podria ser la següent:

```
<html>
<body>
<%
response.write("Hola món!")
%>
</body>
</html>
```

El resultat de l'exemple és una simple pàgina que mostra el text *Hola món!*

ASP ofereix avantatges sobre CGI, ja que crear i mantenir aplicacions ASP resulta molt més senzill. D'altra banda, ASP també ofereix els avantatges de reutilització de codi i processos igual que ISAPI.

3.4. PHP

PHP¹³ és un llenguatge interpretat i utilitzat generalment per al desenvolupament de pàgines web dinàmiques.

⁽¹³⁾ PHP és la sigla d'*hypertext pre-processor*.

Va ser creat per Rasmus Lerdorf el 1994 i va ser continuat per The PHP Group. Aquest grup defineix l'estàndard del llenguatge en no disposar-ne d'un oficial. PHP és un llenguatge basat en eines amb llicència de programari lliure.

Igual que les pàgines ASP, PHP es pot incrustar dins de pàgines HTML per a generar pàgines dinàmiques. El servidor que allotja les pàgines ha de disposar d'un mòdul per a la interpretació d'aquest llenguatge.

Servidor de PHP

Generalment s'utilitza el servidor d'aplicacions Apache amb el mòdul PHP.

PHP ha tingut un èxit generalitzat en el desenvolupament web gràcies a la seva senzillesa, al fet que és gratuït i que disposa d'un gran conjunt d'extensions que permeten l'accés a diferents proveïdors de bases de dades i altres tasques habituals com expressions regulars, suport XML, *sockets*, etc.

La següent pàgina PHP mostra cinc vegades el text "Hola món" de manera dinàmica:

```
<html>
  <head>
    <title>Exemple PHP</title>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
  </head>

  <body>
    <?php
```

```
$makeHTML = '<h%d>Hola mundo</h%d>';  
for($i = 6; $i > 1; $i--){  
    printf($makeHTML,$i,$i);  
}  
?  
</body>  
</html>
```

El codi en l'interior de l'etiqueta `<?php...?>` és interpretat pel servidor en sol·licitar la pàgina. La sortida generada s'inserirà a l'interior del cos de la pàgina, i mostrarà el resultat següent:

```
<html>  
  <head>  
    <title>Exemple PHP</title>  
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  
  </head>  
  
  <body>  
    <h6>Hola mundo</h6>  
    <h5>Hola mundo</h5>  
    <h4>Hola mundo</h4>  
    <h3>Hola mundo</h3>  
    <h2>Hola mundo</h2>  
  </body>  
</html>
```

3.5. Java EE

Java Platform Enterprise Edition és una plataforma independent i centrada en el llenguatge Java per al desenvolupament i l'execució d'aplicacions distribuïdes. Va ser creada i desenvolupada per Sun Microsystems i posteriorment va ser adquirida per Oracle.

Java EE permet desenvolupar aplicacions portables, robustes i escalables amb una arquitectura distribuïda en capes ben diferenciades i la interacció entre aquestes. Disposa d'un conjunt de serveis, API i protocols i comprèn un ampli conjunt de tecnologies i especificacions per a desenvolupar aplicacions web, accés a bases de dades, serveis web, suport per a l'XML, correu electrònic, etc.

Java EE està orientat a aplicacions de gran envergadura i les seves especificacions permeten la comunicació, distribució i reutilització dels components. En fer una separació del codi en nivells ben diferenciats, també se simplifica el manteniment de les aplicacions.

Aplicació distribuïda

Una aplicació distribuïda consisteix en un conjunt de components que s'executen en entorns separats, generalment connectats per mitjà d'una xarxa.

Accés a bases de dades

Java disposa de la seva API d'accés a bases de dades, anomenada JDBC (*Java database connectivity*).

3.6. ASP.NET

ASP.NET és una tecnologia per a crear aplicacions web interactives i dinàmiques. Forma part de la plataforma .NET de Microsoft i va ser introduït l'any 2000 com a resposta a la plataforma Java EE.

Plataforma .NET

La plataforma .NET és un entorn de treball (*framework*) de Microsoft per al desenvolupament d'aplicacions web, escriptori, serveis, etc. Inclou un conjunt de llenguatges i biblioteques de classes.

L'execució de les aplicacions .NET es duu a terme sota una màquina virtual comparable a la JVM.

ASP.NET ofereix un model de programació molt similar al de les aplicacions d'escriptori mitjançant les pàgines ASP.NET. Aquestes són compostes per una part visual que s'executa en el costat del client i una part lògica que s'executa en el costat del servidor per a respondre a les accions i als esdeveniments des del navegador client.

Les pàgines ASP.NET tenen un aspecte molt semblant a una pàgina HTML i poden incorporar controls i codi *script* de servidor. Aquests elements s'executaran al servidor per a generar codi HTML abans que se serveixi la pàgina.

El codi següent correspon a una pàgina ASP.NET que conté un control de servidor i un *script*:

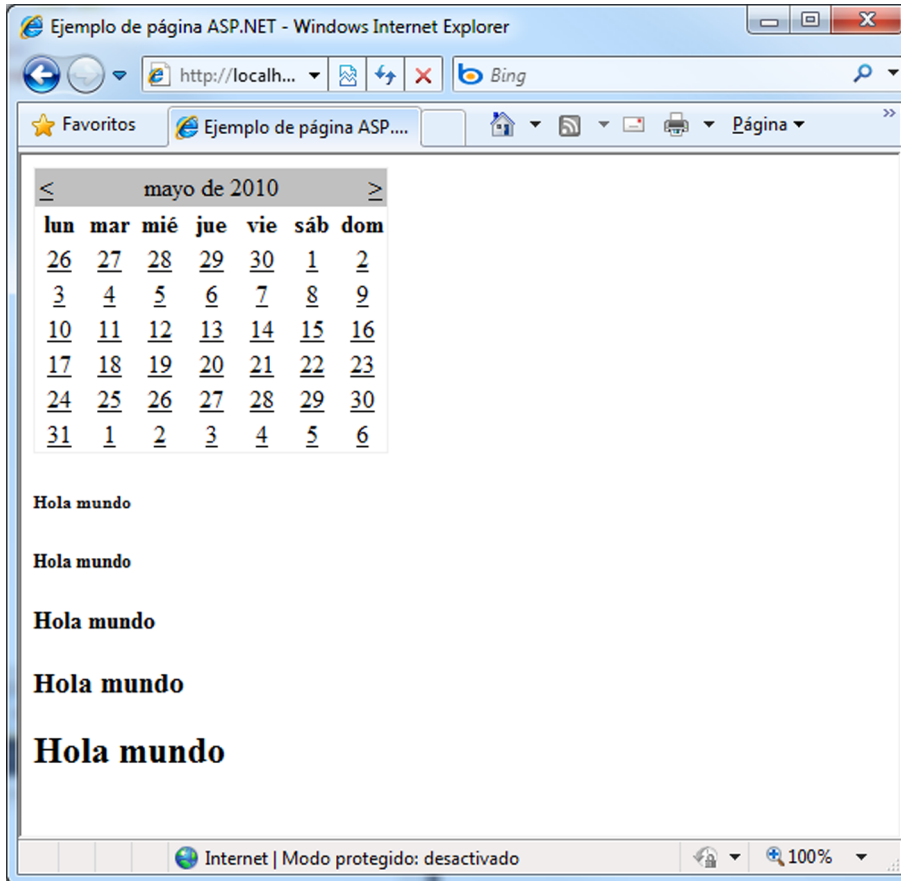
```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile=" Default2.aspx.cs" Inherits="Default2" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Ejemplo de página ASP.NET</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar1" runat="server"></asp:Calendar>
            <% for (int i = 6; i > 1; i-->
                {%>
                <h<%=i%>>Hola mundo</h<%=i%>>
                <%} %>
            </div>
        </form>
    </body>
</html>
```

La pàgina té un aspecte molt semblant al d'un document HTML corrent excepte per la directiva `<%@ Page %>` en la part superior. Aquesta indica algunes característiques de la pàgina com el llenguatge de programació utilitzat, en aquest cas C#, o el fitxer en què hi ha el codi. Al cos es troba el bloc `<form>`, en què hi ha un control de servidor i un *script*. El control de servidor correspon a

l'etiqueta `<asp:Calendar>` que permet mostrar un calendari. Posteriorment, entre etiquetes `<% %>` hi ha el codi *script* de la pàgina. Aquest simplement fa un bucle per a escriure “Hola mundo” cinc vegades entre capçaleres de `<h6>` fins i tot `<h2>`. El resultat de la pàgina es mostra en la figura 5.

Figura 5. Resultat de l'exemple ASP.NET



Si es comprova el codi font, es pot veure que el control de servidor i l'*script* s'han transformat en codi HTML perquè pugui ser visualitzat pel navegador del client.

3.7. Ruby on Rails

Ruby és un llenguatge de programació interpretat i orientat a objectes inspirat en altres llenguatges com Python, Perl i Smalltalk. El resultat és un llenguatge de programació concís i llegible però potent alhora. D'altra banda, Rails és un entorn de treball¹⁴ basat en el llenguatge Ruby que facilita el desenvolupament, desplegament i manteniment d'aplicacions web.

⁽¹⁴⁾En anglès, *framework*.

El llenguatge Ruby és un llenguatge de codi obert i gratuït creat per Yukihiro Matsumoto. Matsumoto va iniciar el seu desenvolupament el 1993 i el va presentar finalment el 1995. Posteriorment va sorgir l'entorn de treball Rails el juliol de 2004 desenvolupat inicialment per David Heinemeier i actualment mantingut per una comunitat de programadors.

Rails està construït sobre dos principis molt sòlids:

1) **No et repeteixis (DRY¹⁵)**: qualsevol concepte en l'aplicació s'ha de definir només una vegada. Això evita la duplicació de la informació.

⁽¹⁵⁾DRY és la sigla de *don't repeat yourself*.

2) **Convenció sobre configuració**: només caldrà definir les configuracions que no segueixen les convencions de Rails.

Aquests dos principis fan que Ruby on Rails sigui un entorn de treball àgil vàlid per a gairebé totes les necessitats del mercat.

Algunes de les característiques més importants de Rails són les següents:

a) El desenvolupament de qualsevol aplicació segueix el patró model-vista-controlador (MVC), que divideix l'aplicació en tres capes lògiques ben diferenciades:

- **Model**: capa encarregada de la gestió de les dades.
- **Vista**: s'encarrega de la interacció amb l'usuari mitjançant la interfície gràfica.
- **Controlador**: capa que rep les accions de l'usuari i fa les accions d'obtenció i representació de dades.

b) Disposa d'una capa ORM per a l'accés a les bases de dades.

c) Les vistes es basen en HTML i codi Rails incrustat per a oferir dinamisme del costat del servidor.

d) Disposa d'un conjunt de controls i funcions AJAX¹⁶ basats en l'entorn de treball Prototype.

ORM

ORM (*object-relational mapping*) es podria traduir per *mapatge objecte-relacional*. Es tracta d'un patró de disseny que relaciona d'una manera automàtica classes d'entitat amb taules en una base de dades relacional.

Prototype

Prototype és una biblioteca escrita en JavaScript per a facilitar el desenvolupament de pàgines web dinàmiques amb AJAX. Ofereix funcions que abstrueixen les diferències entre les implementacions del DOM pels navegadors.

⁽¹⁶⁾AJAX és la sigla d'*asynchronous JavaScript and XML*.

4. RIA

Les RIA¹⁷ són aplicacions web que ofereixen característiques addicionals sobre les aplicacions tradicionals basades en pàgines dinàmiques. Aquestes característiques addicionals es basen en millores gràfiques, d'interacció i usabilitat.

⁽¹⁷⁾RIA és la sigla de *rich Internet applications*.

Les aplicacions RIA ofereixen resultats comparables a les aplicacions d'escriptori. Segons la tecnologia utilitzada, es requereix la instal·lació d'un complement al navegador per a poder interpretar les interfícies rebudes del servidor, com és el cas de Silverlight i JavaFX.

Generalment, les aplicacions web requereixen recarregar la pàgina completa per a mostrar informació nova, per insignificant que sigui. Durant l'ús de l'aplicació es produeix un trànsit innecessari de pàgines entre client i servidor que degrada la interactivitat i l'experiència de l'usuari. D'altra banda, les aplicacions RIA permeten mantenir una comunicació asíncrona amb el servidor per a l'intercanvi d'informació sense interferir amb la interacció de l'usuari.

A continuació, s'enumeren algunes de les tecnologies que permeten desenvolupar aplicacions RIA.

4.1. AJAX

El model tradicional d'una aplicació web es basa en la recàrrega total d'una pàgina per a mostrar informació nova. AJAX¹⁸ és l'ús conjunt de diferents tècniques i tecnologies que permet mantenir una comunicació asíncrona amb el servidor i així actualitzar parts del document d'una manera dinàmica.

⁽¹⁸⁾AJAX és la sigla d'*asynchronous JavaScript and XML*.

Encara que ja es feia ús d'aquestes tècniques i tecnologies, no va ser fins a l'any 2005 que va rebre el nom d'AJAX. L'ús d'AJAX es va popularitzar en part gràcies a Google amb les seves aplicacions GMail, Google Docs i d'altres.

Les principals tecnologies utilitzades per AJAX són XHTML, JavaScript, XML i CSS.

4.2. Adobe Flex

Adobe Flex és un entorn per al desenvolupament i desplegament d'aplicacions RIA basades en el seu reproductor Flash Player. Sorgeix el 2004 com un conjunt de tecnologies propietàries de Macromedia entre les quals hi ha ActionScript i MXML¹⁹.

ActionScript és un llenguatge de *script* orientat a objectes per a implementar la lògica de l'aplicació. D'altra banda, MXML o *multimedia extensible markup language*, és el llenguatge de marques basat en XML emprat per a descriure les interfícies d'usuari. MXML es pot semblar a XHTML i ActionScript a JavaScript.

Flex permet crear aplicacions sofisticades amb interfícies atractives i una gran interactivitat: arrossegar i deixar anar objectes, llistes ordenables, animacions, etc. Les aplicacions s'executen a partir del navegador, que haurà de disposar del complement de Flash Player.

4.3. JavaFX

JavaFX engloba un conjunt de tecnologies per a desenvolupar aplicacions RIA executables en una gran varietat de dispositius. Actualment és possible desenvolupar aplicacions JavaFX per a escriptori, navegadors i dispositius mòbils.

Les aplicacions JavaFX utilitzen el llenguatge JavaFX Script. Aquest va ser desenvolupat per Sun Microsystems per a implementar d'una manera ràpida interfícies d'usuari per a la plataforma Java. Les aplicacions JavaFX poden fer ús de qualsevol classe Java d'una manera directa.

4.4. Silverlight

Silverlight és un entorn de treball per a construir aplicacions RIA allotjades en servidors. Igual que Adobe Flex i JavaFX, Silverlight requereix tenir instal·lat un complement en el navegador. Quan el navegador carrega una pàgina que inclou codi Silverlight, el complement corresponent s'executa per a renderitzar i interpretar el contingut en una regió específica de la pàgina.

Amb Silverlight es poden crear pàgines interactives amb gràfics, elements multimèdia i gràfics vectorials.

Silverlight està basat en un subconjunt de la plataforma .NET. Per a programar les interfícies, s'utilitza un llenguatge declaratiu basat en XML anomenat XAML²⁰. La lògica es fa mitjançant els llenguatges C# o Visual Basic.

⁽¹⁹⁾ MXML és la sigla de *multimedia extensible markup language*.

MXML

MXML és un llenguatge per a descriure interfícies d'usuari. En ser textual, els desenvolupadors poden llegir i modificar les interfícies amb un simple editor de text.

Aplicacions JavaFX

Sun planeja implementacions de JavaFX per a altres dispositius com ara reproductors DVD o BlueRay, televisions, reproductors lleugers com els que podem trobar als cotxes, etc.

⁽²⁰⁾ XAML és la sigla d'*extensible application markup language*. XAML és un llenguatge de marques basat en XML per a descriure interfícies.

Microsoft disposa de l'eina Blend que permet dissenyar interfícies d'una manera gràfica i ràpida. La lògica, la compilació i el desplegament es poden dur a terme mitjançant les eines de Visual Studio.