

## **Agraïments**

Durant els últims mesos he estat treballant, gairebé en tot moment, d'una forma o altre en la realització d'aquest projecte. En nombroses ocasions he hagut d'acudir a amics i companys, els quals han estat al meu costat, ajudant-me i donant-me suport per poder arribar a aconseguir la fita: acabar satisfactòriament el projecte.

A l'empresa eMascaró Crossmedia i més concretament a Toni Mascaró, Laura Renzetti i Jordi F. per haver-me donat l'oportunitat de treballar i realitzar el projecte dins de l'empresa on treballa.

A l'Anna G., per la paciència i el continu suport fent-me veure que era capaç d'acabar exitosament el projecte.

Als meus companys i amics Javier M., Arturo M., David A. i Sergi P., per tota l'ajuda i consultes resoltes sobre ActionScript i PHP.



## **Resum:**

“Google Analytics per a tothom” són un conjunt d’aplicacions web per als clients de l’empresa eMascaró Crossmedia. Aquests podran analitzar i portar un estricte control de les estadístiques més significatives de les seves pàgines web.

El projecte consta de dues grans aplicacions principalment:

- Aplicació en Flash per a usuaris principiants.
- Aplicació per a telèfons mòbils o PDAs per a empresaris o gent més experimentada.

## **Resumen:**

“Google Analytics per a tothom” son un conjunto de aplicaciones web para los clientes de la empresa eMascaró Crossmedia. Estos podrán analizar i llevar un estricto control de las estadísticas más significativas de sus páginas web.

El proyecto consta de dos grandes aplicaciones principalmente:

- Aplicación en Flash para usuarios principiantes.
- Aplicación para teléfonos móviles o PDAs para empresarios o gente más experimentada.

## **Abstract:**

"Google Analytics per a tothom" is a collection of web applications for the company eMascaró Crossmedia customers, who will be able to analyze and to have control of the most representative statistics of their web sites.

The project contains two main applications:

- Flash applications for beginner level customers
- Mobile phone or PDA applications for business people or for people with higher knowledge.



## Index

<b>1. INTRODUCCIÓ</b> .....	<b>1</b>
<b>2. OBJECTIUS</b> .....	<b>3</b>
<b>3. PLANIFICACIÓ</b> .....	<b>5</b>
<b>4. API DE GOOGLE ANALYTICS</b> .....	<b>9</b>
4.1 Documentació de la Api .....	9
<b>5. LENGUATGES DE PROGRAMACIÓ I EINES</b> .....	<b>11</b>
5.1 Llenguatges de programació.....	11
5.1.1 Què es PHP? .....	12
5.1.2 Característiques del PHP .....	13
5.1.3 Què és ActionScript? .....	14
5.2 Programes .....	17
5.2.1 Macromedia Dreamweaver CS3.....	17
5.2.2 Macromedia Flash CS3 .....	20
5.2.3 Sepy .....	22
5.2.4 Photoshop CS3 .....	23
5.2.5 PhpMyAdmin .....	25
<b>6. GOOGLE ANALYTICS AMB FLASH</b> .....	<b>27</b>
6.1 Entrada al Gestor .....	27
6.1.1 Introducció.....	27
6.1.2 Argumentació del codi font .....	28
6.2 Entrada al Panell d'accés.....	31
6.2.1 Introducció.....	31
6.2.2 Argumentació del codi font .....	33
6.3 Panell Google Analytics en Flash.....	39
6.3.1 Introducció.....	39
6.3.2 Navegant per l'aplicació.....	41
6.4 Codi font .....	52
<b>7. GOOGLE ANALYTICS MOBILE</b> .....	<b>67</b>
7.1 Navegant per l'aplicació.....	67
7.1.1 Introducció.....	67
7.1.2 Panell d'accés .....	69
7.1.3 Llistat de llocs web .....	70
7.1.4 Panell d'estadístiques generals .....	71

7.1.5 Panell de Menú.....	73
7.1.6 Panell de pàgines més vistes .....	74
7.1.7 Panell de paraules clau .....	75
7.1.8 Panell de països .....	76
7.1.9 Panell de referidors.....	77
7.1.10 Panell de navegadors .....	78
7.1.11 Logout .....	78
7.2 Codi Font.....	79
7.2.1 Introducció .....	79
7.2.2 Panell d'accés.....	81
7.2.3 Llistat de llocs web.....	84
7.2.4 Panell d'estadístiques generals.....	89
7.2.5 Resta de panells .....	92
<b>8. CONCLUSIONS.....</b>	<b>97</b>
<b>9. BIBLIOGRAFIA .....</b>	<b>99</b>
<b>10. ANNEX: CONTINGUT DEL CD.....</b>	<b>101</b>

## Índex de Figures destacades

<b>Figura 1:</b> Exemple de “dimensions” .....	10
<b>Figura 2:</b> Exemple de “metrics”.....	10
<b>Figura 3:</b> Esquema funcionament PHP .....	13
<b>Figura 4:</b> Esquema càrrega del ActionScript 2 .....	15
<b>Figura 5:</b> Esquema càrrega del ActionScript 3 .....	16
<b>Figura 6:</b> Dreamweaver CS3 .....	17
<b>Figura 7:</b> Macromedia Flash CS3 .....	20
<b>Figura 8:</b> Sepy .....	22
<b>Figura 9:</b> Photoshop CS3 .....	23
<b>Figura 10:</b> PhpMyAdmin .....	25
<b>Figura 11:</b> Entrada al gestor de contingut .....	27
<b>Figura 12:</b> Error en l'accés al gestor de continguts .....	28
<b>Figura 13:</b> Entrada al panell d'accés .....	31
<b>Figura 14:</b> Configuració de la compte .....	32
<b>Figura 15:</b> Camps d'accés al panell d'analytics .....	33
<b>Figura 16:</b> Google Analytics en Flash .....	39
<b>Figura 17:</b> Zones del panell Google Analytics en Flash .....	39
<b>Figura 18:</b> Calendari del panell Google Analytics en Flash .....	41
<b>Figura 19:</b> Panell principal Google Analytics en Flash .....	42
<b>Figura 20:</b> Gràfic ampliat del panell principal de Google Analytics en Flash .....	43
<b>Figura 21:</b> Gràfic del panell visió general dels usuaris .....	45
<b>Figura 22:</b> Detall del gràfic del panell visió general dels usuaris .....	45
<b>Figura 23:</b> Mapa del panell de visites per ubicació .....	46
<b>Figura 24:</b> Gràfic del panell visió general de les fonts de trànsit .....	49
<b>Figura 25:</b> Taula del panell de paraules clau .....	51
<b>Figura 26:</b> Google Analytics Mobile .....	67
<b>Figura 27:</b> Accés a Google Analytics Mobile .....	69
<b>Figura 28:</b> Llistat de llocs web Google Analytics Mobile .....	70
<b>Figura 29:</b> Panell d'estadístiques generals Google Analytics Mobile .....	71
<b>Figura 30:</b> Panell Menú de Google Analytics Mobile .....	73

<b>Figura 31:</b> Panell de pàgines més vistes Google Analytics Mobile .....	74
<b>Figura 32:</b> Panell de paraules clau Google Analytics Mobile.....	75
<b>Figura 33:</b> Panell de països de Google Analytics Mobile .....	76
<b>Figura 34:</b> Panell de referidors de Google Analytics Mobile .....	77
<b>Figura 35:</b> Panell de navegadors de Google Analytics Mobile .....	78



## 1. Introducció

Avui dia, es pot fer un seguiment de les estadístiques de les webs mitjançant l'accés públic de Google ([http://www.google.com/intl/es\\_ALL/analytics/index.html](http://www.google.com/intl/es_ALL/analytics/index.html)). Mitjançant aquest accés, qualsevol usuari pot consultar un ampli nombre de dades d'analítica web.

Traducció de la definició de Google Analytics de la web oficial:

**“La solució d'anàlisi web gratuïta per empreses, i més intel·ligent i fàcil d'utilitzar.”**

*Google Analytics és una solució d'anàlisi web per empreses que proporciona informació molt vàlida sobre el trànsit del lloc web i la eficàcia del pla de màrqueting. Ara, gràcies a unes funcions potents, flexibles i fàcils de fer servir, podrà veure i analitzar el trànsit des d'una perspectiva diferent. Google Analytics l'hi ajudarà a dissenyar anuncis orientats, a millorar les seves iniciatives de màrqueting i a crear llocs web que generen més conversions.”*

Amb Google Analytics es pot saber la procedència de totes les visites de la web, el nombre de pàgines que un usuari visita, el temps que l'usuari està visitant la pàgina web i moltes altres dades d'interès. Disposar de les dades més importants, permetrà a l'usuari poder-les analitzar correctament i conèixer l'estat real de la pàgina web. Així com decidir si es necessari realitzar millores a la pàgina web.

Totes aquestes dades, per a un usuari expert en anàlisi de pàgines web, són de gran utilitat. En canvi, si l'usuari que accedeix té uns nivells baixos-mitjos de coneixements en analítica web, totes aquestes dades es converteixen en informació poc útil. L'usuari, no sabrà per on començar a interpretar les dades, i com a conseqüència, no serà capaç extreure conclusions de si el lloc web té un funcionament correcte.

Amb les eines que formaran part d'aquest projecte, es facilitarà a tot tipus d'usuari a tenir accés a les dades que més informació pot aportar en cada cas segons les necessitats.

Aquest és un projecte real realitzat per a que l'empresa **eMascaró Crossmedia** vengui als seus clients. La majoria de requeriments, formen part d'un estudi del client o de les necessitats que aquests han fet saber a la nostre empresa.



## 2. Objectius

L'objectiu principal del projecte és aconseguir que els diferents usuaris puguin portar un control adequat de les estadístiques de les seves webs independentment dels seus coneixements en **SEO** (*Search Engine Optimization*).

Una web, per estar ben posicionada en un cercador (**Google, Yahoo** o **Bing** per exemple), ha de seguir tot un conjunt de normes, consells i/o directrius per tal de poder-se situar en les posicions més altes dels cercadors i, d'aquesta forma, tenir una major probabilitat de ser trobada i visitada per l'usuari.

Actualment, una de les feines més valorades en el món de les pàgines webs, és la de, mitjançant tècniques legals, posicionar una pàgina web en les primeres posicions dels cercadors. Aquesta tasca comença amb un exhaustiu estudi de totes les dades que proporciona Google Analytics.

Amb la realització d'aquest projecte, tant usuaris experts com usuaris principiants, podran portar aquest control tant important dels accessos a la pàgina web. Cal tenir molt present que aquesta és una aplicació de gestió de dades, en cap moment es podran realitzar modificacions a les pàgines web. Per a realitzar aquestes millores, que podran comportar millores en les estadístiques, l'usuari haurà de posar-se en contacte amb el seu desenvolupador web.

Es buscarà la forma més adequada, creant diferents aplicacions si fos necessari, per que els diferents tipus d'usuaris puguin desenvolupar tasques d'anàlisi web de forma senzilla.



### 3. Planificació

El projecte es desglossa en diverses fases per tal de poder assolir de forma satisfactòria el desenvolupament de l'aplicació. Es necessari trobar la forma més eficaç i eficient de desenvolupar per tal de satisfer al client.

- En reunions amb clients, s'observa que aquests, tenen **problemes alhora d'interpretar la informació** de Google Analytics. El client no és capaç d'entendre correctament la informació que Google els ofereix, o expressa limitacions alhora de poder visualitzar la informació des de dispositius mòbils fora del seu despatx.
- Es determina els **diferents tipus d'usuari**. Ens trobem amb un perfil de nivell baix-mitg (tipus A) de coneixements en posicionament web i un altre perfil amb un nivell més experimentat en la matèria (tipus B).
  - o Els usuaris tipus A correspondran a persones que desenvolupen altes tasques dins de la seva feina, però que s'encarregaran de tot el tema de la web i per tant, els seus coneixements seran força limitats. Per altre banda, els usuaris de tipus B correspondran a informàtics o els propis caps de l'empresa, que en tot moment voldran conèixer i portar un control sobre l'evolució del seu lloc web.
- Un cop determinats els usuaris es realitza un **estudi de les característiques de cada tipus d'usuari**. L'usuari de baix nivell, tipus A, requerirà una interfície gràfica senzilla i pràctica amb la informació bàsica de Google Analytics. L'usuari de nivell mitg-alt, tipus B, requerirà una interfície més elaborada i amb una informació més detallada i útil alhora d'analitzar el correcte funcionament del lloc web.

- Amb els perfils d'usuari definits, és el moment de **pensar possibles solucions**. Al tractar-se de dos tipus d'usuaris ben diferents, la millor solució serà realitzar **dues aplicacions**.
  - Per els usuaris de tipus A, es pot realitzar una aplicació de escriptori amb tota la informació de Google Analytics que es desitgi, o una aplicació web en Flash, per aconseguir una interfície gràfica senzilla però atractiva. L'aplicació d'escriptori té un contra molt important per aquest tipus d'usuari, i és que requereix una instal·lació prèvia; i per experiències d'anteriors clients que han trucat a l'empresa per que se'ls expliquin accions informàtiques molt bàsiques, trobem que aquesta opció podria portar molts problemes.
  - Per els usuaris de tipus B, es pot realitzar una aplicació completa d'escriptori, amb tota la informació ben detallada sobre el seu lloc web, o preparar una aplicació web per a dispositiu mòbil amb la informació detallada, però alhora útil per analitzar el lloc web amb les dades més importants. Qualsevol de les dues opcions, pot ser una gran alternativa per aquests usuaris.
  
- Un cop valorades les diferents opcions que hem plantejat, s'ha de **triar una de les solucions per a cada tipus d'usuari**.
  - Per els usuaris de **tipus A**, optarem per l'opció de **l'aplicació web en Flash**, ja que considerem que és molt més senzill de fer servir i, alhora, aconseguirem mostrar les dades de forma dinàmica per que l'usuari es trobi agust utilitzant-la.
  - Per els usuaris de **tipus B**, finalment **l'aplicació web per a dispositius mòbils** és la més adequada, ja que hem de tenir en compte que al tractar-se de gent de negocis, sovint no estaran a l'oficina, o amb un ordinador a mà, i per tant, una aplicació compatible per a tots els telèfons mòbils serà de gran utilitat.

- A continuació és molt important **valorar i decidir quin llenguatge de programació** s'utilitzarà.
  - o Versió Google Analytics en Flash: Per a la comunicació amb Google es farà servir: PHP. Per a mostrar les dades a l'usuari es mostrarà mitjançant Flash. Per a poder comunicar el Flash i el PHP es farà servir ActionScript. En aquest cas s'ha triat la versió AS3.
  - o Versió Google Analytics Mobile: En aquest cas, tota l'aplicació es realitzarà en PHP, HTML i CSS. En cap moment es farà servir Flash o funcions JavaScript, ja que hi ha alguns dispositius mòbils que no ho accepten. Estem intentant crear una eina compatible amb tots els navegadors i dispositius per tal de poder garantir al client una total accessibilitat des del seu dispositiu mòbil.
- Un cop tot decidit, és molt important **realitzar un estudi de la Api de Google Analytics**; observant que només es disposa d'una versió de la Api per a PHP, per tant, serà necessari realitzar una conversió de PHP a ActionScript per a l'aplicació en Flash dels usuaris tipus A.
- Finalment s'ha de **desenvolupar l'aplicació**. En aquesta etapa bàsicament ens dedicarem a escriure tot el codi de programació necessari perquè l'aplicació funcioni correctament.





## 4. Api de Google Analytics

### 4.1 Documentació de la Api

Aquesta és la URL de la informació que Google proporciona de la seva Api:

<http://code.google.com/intl/es-ES/apis/analytics/docs/gdata/gdataDeveloperGuide.html>

En l'estudi de la Api de Google Analytics, hem pogut observar que únicament es pot disposar d'una versió de la Api en el llenguatge PHP, per tant, a mesura que anem extraient les dades que necessitem, haurem de realitzar una conversió d'aquestes, i comunicar-les de la forma més adequada amb el ActionScript, per a poder mostrar-les en Flash satisfactòriament en el cas de l'aplicació en Flash. Per una altra banda, en la versió mòbil, es faran servir arxius XML per a emmagatzemar les dades temporalment.

Una de les parts importants de la Api de Google Analytics, és que per extreure les dades que desitgem, s'han de fer combinacions de dos o més paràmetres. Aquests són les “*dimensions*” i “*metrics*”.

En la documentació de la Api de Google Analytics s'explica detalladament quin és el valor que retorna cada consulta d'una “*dimension*” o “*metric*”.

## Exemple d'algunes “dimensions”:

Dimensions		
The following table lists the available dimension fields.		
Dimension Category	Dimension Name	Description
D1. Visitor		
	<code>ga:browser</code>	The names of browsers used by visitors to your website. For example, "Internet Explorer" or "Firefox."
	<code>ga:browserVersion</code>	The browser versions used by visitors to your website. For example, 2.0.0.14
	<code>ga:city</code>	The cities of website visitors, derived from IP addresses. The <code>city</code> field falls in a hierarchy of geographical groupings used in Analytics, which proceeds in the following order: continent, sub-continent, country, region, sub-region, and city.
	<code>ga:connectionSpeed</code>	The qualitative network connection speeds of website visitors. For example, T1, DSL, Cable, Dialup.
	<code>ga:continent</code>	The continents of website visitors, derived from IP addresses.
	<code>ga:countOfVisits</code> (deprecated)	See <a href="#">ga:visitCount</a> . Number of visits to your website. This is calculated by determining the number of visitor sessions.
	<code>ga:country</code>	The countries of website visitors, derived from IP addresses.
	<code>ga:date</code>	The date of the visit. An integer in the form YYYYMMDD.
	<code>ga:day</code>	The day of the month from 01 to 31.
	<code>ga:daysSinceLastVisit</code>	The number of days elapsed since visitors last visited your website. Used to calculate visitor loyalty.

Figura 1: Exemple de “dimensions”

## Exemple d'alguns “metrics”:

Metrics		
Metric Category	Metric Name	Description
M1. Visitor		
	<code>ga:bounces</code>	The total number of single-page visits to your website.
	<code>ga:entrances</code>	The number of entrances to your website. The value will always be equal to the number of visits when aggregated over your entire website. Thus, this metric is most useful when combined with dimensions such as <code>ga:landingPagePath</code> , at which point entrances as a metric indicates the number of times a particular page served as an entrance to your website.
	<code>ga:exits</code>	The number of exits from your website. As with entrances, it will always be equal to the number of visits when aggregated over your entire website.
	<code>ga:newVisits</code>	The number of visitors whose visit to your website was marked as a first-time visit.
	<code>ga:pageviews</code>	The total number of pageviews for your website when aggregated over the selected dimension. For example, if you select this metric together with <code>ga:pagePath</code> , it returns the number of page views for each page.
	<code>ga:timeOnPage</code>	How long a visitor spent on a particular page or set of pages. Calculated by subtracting the initial view time for a particular page from the initial view time for a subsequent page. Thus, this metric does not apply to exit pages for your website. The value from this metric is returned in the XML as a string, with the value represented in total seconds. Different client libraries have various ways of representing this value, such as a double, float, long, or string.
	<code>ga:timeOnSite</code>	The total duration of visitor sessions over the selected dimension. For example, suppose you combine this field with a particular ad campaign. In this case, the metric will display the total duration of all visitor sessions for those visitors who came to your website via a particular ad campaign. The value from this metric is returned in the XML as a string, with the value represented in total seconds. Different client libraries have various ways of representing this value, such as a double, float, long, or string.

Figura 2: Exemple de “metrics”

Per tant, serà necessari realitzar totes les consultes i comprovacions necessàries fins a obtenir el valor de la dada desitjada.

## **5. Llenguatges de programació i eines**

### **5.1 Llenguatges de programació**

Per a la realització del projecte és molt important fer una bona tria dels diferents llenguatges de programació que es faran servir. És indispensable tenir en compte en tot moment les avantatges i limitacions que cada llenguatge ofereix. Per a poder garantir una bona optimització de l'aplicació, també ens trobem amb la dificultat de saber triar els llenguatges adequats per tal que la comunicació entre el diferents llenguatges es pugui realitzar amb el menor nombre de dificultats i problemes.

Aquest projecte està implementat en PHP i ActionScript 3.

De tots els llenguatges de programació web, el PHP s'ha considerat el llenguatge més adequat per a portar a terme aquest primer procés de desenvolupament del projecte que consisteix en recollir la informació de la Api de Google Analytics.

Primer, s'ha d'analitzar la Api de Google Analytics i realitzar les crides corresponents per a extreure la informació necessària. Tot això es farà amb PHP.

En el cas de l'aplicació de Google Analytics en Flash, el PHP ens facilitarà en tot moment l'acció de crides des del Flash (en ActionScript) per rebre la informació en un format estructurat de la forma adequada per poder-la interpretar. Un cop interpretada, s'haurà de tractar per mostrar-la per pantalla de forma que l'usuari pugui entendre-la.

Per una altre banda, en l'aplicació per a telèfon mòbil, la informació recollida amb el PHP de la Api de Google Analytics, la guardarem en arxius XML. D'aquesta forma, només caldrà realitzar la descarrega de les dades de Google, un cop al dia.

El PHP és un llenguatge de programació amb el qual ja estic familiaritzat. Per tant, només caldrà fer ús de l'ajuda de la documentació de PHP, en determinats moments per a fer servir funcions molt determinades que normalment no s'utilitzen per desenvolupar i programar pàgines web. D'aquesta manera, també ens podrem centrar més en l'aspecte de processament i tractament de dades mitjançant ActionScript 3 i en crear una interfície gràfica més adequada en cada cas.

### 5.1.1 Què es PHP?

El PHP és un acrònim que significa *PHP Hypertext Pre-processor*. Inicialment el seu nom va ser PHP Tools. Es tracta d'un llenguatge interpretat d'alt nivell inclòs en pàgines web HTML i executat en un servidor.

Traducció de la definició de PHP segons PHP.net:

*“El PHP és un llenguatge de script incrustat dins del HTML. La major part de la seva sintaxis ha estat agafada de C, Java i Perl amb algunes característiques específiques de si mateix. La fita del llenguatge és permetre ràpidament als programadors web poder generar pàgines dinàmiques.”*

La versió més actual del PHP és la 5.3.0 (per a Windows) que està en ús des del 30 de Juny de 2009. Però per a trobar les primeres referències al PHP, ens hem de traslladar a finals del 1994, quan Rasmus Lerdorf va escriure un codi com a resultat d'una modificació del Perl. El seu primer us va ser per mantenir el control sobre qui visitava el seu currículum en el seu propi lloc web. Durant els tres anys següents, el codi va anar evolucionant fins el que es coneix com a PHP/FI 2.0 que va permetre començar arribar a un gran nombre d'usuaris. No va ser fins el 1997 quan Zeev Surasky i Andi Gutmans van incloure noves característiques i d'aquesta forma el llenguatge va rebre el nom de PHP 3.0, arribant a una gran quantitat d'usuaris i agafant força pes en el món de la programació. La versió 4 va aconseguir que més d'un milió de servidors tinguessin aquesta capacitat implementada i el nombre d'usuaris van seguir augmentant fins arribar a la versió actual en la que el PHP s'ha convertit en el principal mètode de programació web.

A continuació tenim un esquema del funcionament general:

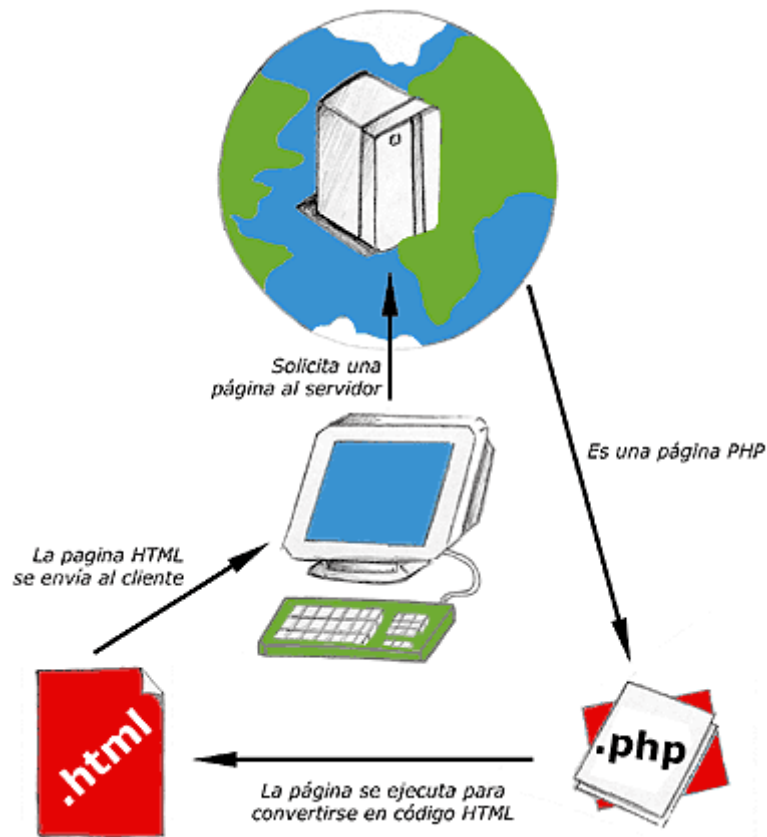


Figura 3: Esquema funcionament PHP

### 5.1.2 Característiques del PHP

El PHP té la capacitat de ser executat en la majoria de sistemes operatius, així com UNIX (i d'altres com Linux o Mac OSX) i Windows, i pot interactuar amb els servidors web més populars ja que existeix en versió CGI, mòdul per Apache, i ISAPI.

A continuació citaré algunes de les característiques més importants del PHP:

- És un llenguatge multiplataforma (un llenguatge multiplataforma és aquell que fa referència als programes o llenguatges de programació que poden funcionar en diverses plataformes com per exemple, Windows, Linux o Mac OSX).
- Capacitat de connexió amb la majoria dels motors de base de dades que s'utilitzen en l'actualitat, destacant la connectivitat amb MySQL.

- Permet aplicar tècniques de programació orientada a objectes.
- El PHP no obliga al programador a seguir una determinada metodologia a l'hora de programar. El programador pot aplicar qualsevol tècnica de programació i desenvolupament que li permeti escriure un codi lògic, ordenat i estructurat.
- PHP té una documentació amplia i completa en la web oficial de PHP.

### 5.1.3 Què és ActionScript?

Per començar a explicar que és l'ActionScript, abans és necessari tenir coneixement de que és el Flash.

Flash és un software originalment dissenyat per crear animacions que es puguin fer servir en pàgines web. Poc a poc, ha anat evolucionant de tal forma, que actualment es fan webs senceres únicament en Flash. El Flash proporciona nombroses avantatges respecte a webs planes fetes amb HTML, PHP, ASP o qualsevol altre llenguatge de programació web. El Flash fa servir vectors gràfics que són ideals per a la web, ja que són lleugers i no consumeixen gaires recursos. El Flash està compost per objectes, cadascú amb la seva ruta respectiva dins del SWF.

Un cop ja tenim una mica de coneixement de que és el Flash, podem passar a explicar detalladament de que es tracta l'ActionScript.

L'ActionScript és un llenguatge de programació utilitzat en aplicacions web animades. Va ser incorporat des de la versió 4 de Flash, on actualment s'ha convertit en un llenguatge robust i complet.

L'ActionScript és un llenguatge de programació que permet ampliar de forma considerable les diferents funcionalitats que Flash ofereix al programador des del panel de disseny. Gracies a ell, podem crear animacions o seqüències amb un alt grau d'interacció entre l'usuari final i l'aplicació.

Actualment, es fa servir des de fa molt poc de temps, la versió 3.0. Aquesta nova versió aporta moltes més possibilitats de la seva antecessora. Un dels principals canvis que es pot apreciar respecte la versió 2.0, és un canvi en la sintaxis del llenguatge. Particularment

aquest aspecte és un punt en contra per a la realització d'aquest projecte, ja que, prèviament a la realització del projecte m'he de familiaritzar amb aquest nou sistema de programació i així, aconseguir fer un bon ús de totes les oportunitats que ofereix ActionScript 3.0.

Per començar a explicar algunes de les característiques del nou llenguatge de programació ActionScript 3, cal remarcar el fet de que aquesta nova versió, és un llenguatge de programació molt més estricte que la seva anterior versió. Per exemple, en aquest cas la definició de variables és molt més estricte i laboriosa que en l'anterior.

És important també citar el fet que en ActionScript 2 (AS2), cada objecte que carrega dades o variables, té el seu propi sistema d'events. En aquest, si en algun moment de la programació, era necessari realitzar la càrrega d'objectes que no corresponien als formats als quals anteriorment s'havien assignat per a ser creats, era molt complicat realitzar-ho.

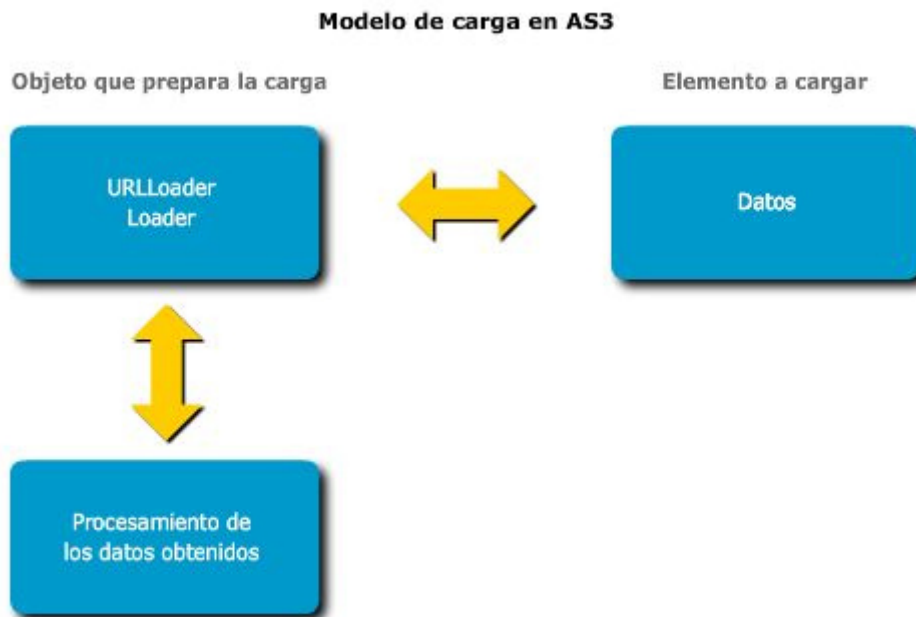
A continuació tenim un esquema del model de càrrega en AS2.



*Figura 4: Esquema càrrega del ActionScript 2*

En ActionScript3 (AS3), tot i que el codi és més complicat i laboriós d'escriure, ofereix un sistema decent i molt eficient des del qual es pot carregar qualsevol tipus d'arxiu. És molt important remarcar el fet que amb aquesta nova versió s'han solucionat problemes com el de realitzar precàrreges per arxius de text o inclús XML.

A continuació tenim un esquema del model de càrrega en AS3.



*Figura 5: Esquema càrrega del ActionScript 3*

Tot seguit trobem un llistat de característiques del nou llenguatge de programació AS3:

- El AS3 ha estat llençat per a fer-se servir amb Flex 2.0
- AS3, té llibreries millorades i una forma de declaració més estricta orientada a objectes. Aquesta permet un millor rendiment i disminució de *memory leaks* (error de software que succeeix quan un bloc de memòria reservada no és alliberada en un programa).
- Les llibreries de connexió a webservice són molt més fàcils d'utilitzar en aquesta versió AS3.



## 5.2 Programes

### 5.2.1 Macromedia Dreamweaver CS3

#### 5.2.1.1 Característiques de Dreamweaver

El Macromedia Dreamweaver és un editor dinàmic HTML de pàgines web. Apart de permetre l'escriptura del codi de programació té l'opció de visualitzar al moment tot allò que s'està programant. Per tant, podem dir que aquesta és una eina útil tant per a dissenyadors com a programadors.

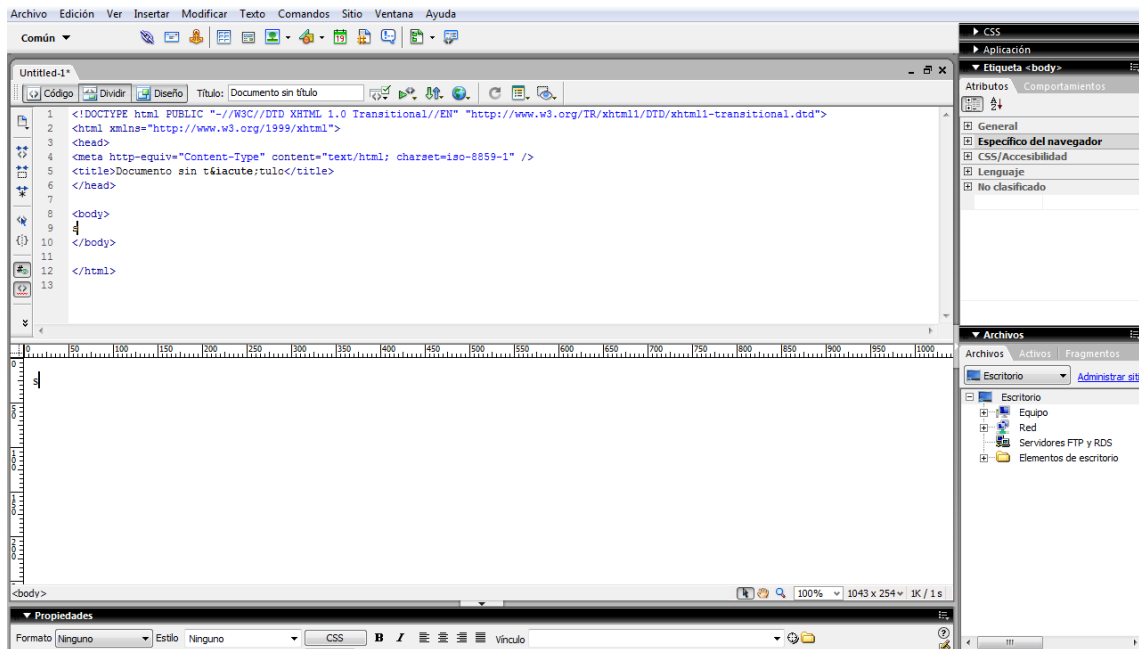


Figura 6: Dreamweaver CS3

El Dreamweaver permet treballar simultàniament amb codi HTML per a crear la base de la web, amb JavaScript per a crear els elements necessaris i amb CSS per a donar un estil gràfic i visual òptim a la web.

Una altre dada important és que el programa permet desenvolupar aplicacions que s'estiguin executant en un servidor i d'aquesta manera poder accedir a les dades i continguts de forma local.

Pot ser un programa adaptable a usuaris amb un nivell baix de coneixements de programació web, ja que disposa d'una gran varietat d'elements preprogramats que únicament cal seleccionar-los i automàticament el programa escriu la part de codi necessari per al bon funcionament d'aquest. Per altre banda és molt útil per a programadors experts, ja que té la capacitat de treballar amb diversos llenguatges de programació web avançada com poden ser PHP, JSP, ASP, ASP.NET i molts altres.

El Dreamweaver disposa de diverses possibilitats que faciliten la connexió a una Base de Dades (BBDD) sense ser un expert en aquesta matèria. Per tant dona accés a usuaris no experts a fer ús d'aquesta metodologia. Tot i així és indispensable tenir certs coneixements de ODBC i tenir per altre banda en el servidor web configurat la BBDD.

Cal remarcar el fet que actualment el Dreamweaver és el programa d'edició de webs dinàmiques més utilitzat per a tot tipus d'usuaris.

### ***5.2.1.2 Macromedia Dreamweaver i el Projecte***

Hi ha una gran varietat de raons per les que Dreamweaver és el editor de pàgines web ideal per al desenvolupament del projecte.

Per començar, cal remarcar que la part del projecte de Google Analytics en Flash, tot i estar realitzat majoritàriament en Flash i ActionScript, està implementat sobre una interfície web en HTML i PHP. Per tant, és indispensable l'ús d'un bon editor. En el cas de Google Analytics Mobile, al estar realitzat tot amb PHP, aquest serà l'únic editor web que farem servir durant el seu desenvolupament.

Dreamweaver permet integrar de forma satisfactòria aplicacions fetes amb Flash, facilitant en tot moment diversos paràmetres a tenir en compte i aconseguir l'efecte desitjat sense un nombre elevat de problemes.

Dreamweaver permet l'escriptura del codi de programació d'una forma clara i neta. L'editor de codi és capaç d'interpretar la presència de diferents codi de programació web en una mateixa plantilla i diferencia mitjançant colors i estils el codi escrit. Això facilita molt l'escriptura i la comprensió del codi escrit.

Exemple:

```
<body>
  <div class="estil1">Això és un exemple de codi HTML</div>
  <?php echo "Això és un exemple de codi PHP"; ?>
</body>
```

Aquest, Permet l'edició estructurada d'altres formats d'arxiu com XML, que en el cas del nostre projecte és important.

Un altre punt molt important i útil de Dreamweaver, és el fet que disposa d'un editor de llocs web prèviament inserits per l'usuari. Permetent l'edició del lloc web, ja sigui en local o mitjançant la modificació dels arxius i la posterior actualització per FTP dels arxius en el servidor. Aquesta opció és molt útil alhora de fer llocs web amb un alt contingut i nombre d'arxius, ja que accedir a ells és molt senzill.

Apart de totes les utilitats que ofereix, el programa ofereix una ajuda molt completa per a qualsevol dubte.

## 5.2.2 Macromedia Flash CS3

### 5.2.2.1 Característiques de Macromedia Flash

El Macromedia Flash és un programa per crear una gran varietat d'aplicacions per a web. Aquest programa disposa de totes les eines necessàries per a realitzar aplicacions animades per diferents plataformes i dispositius.

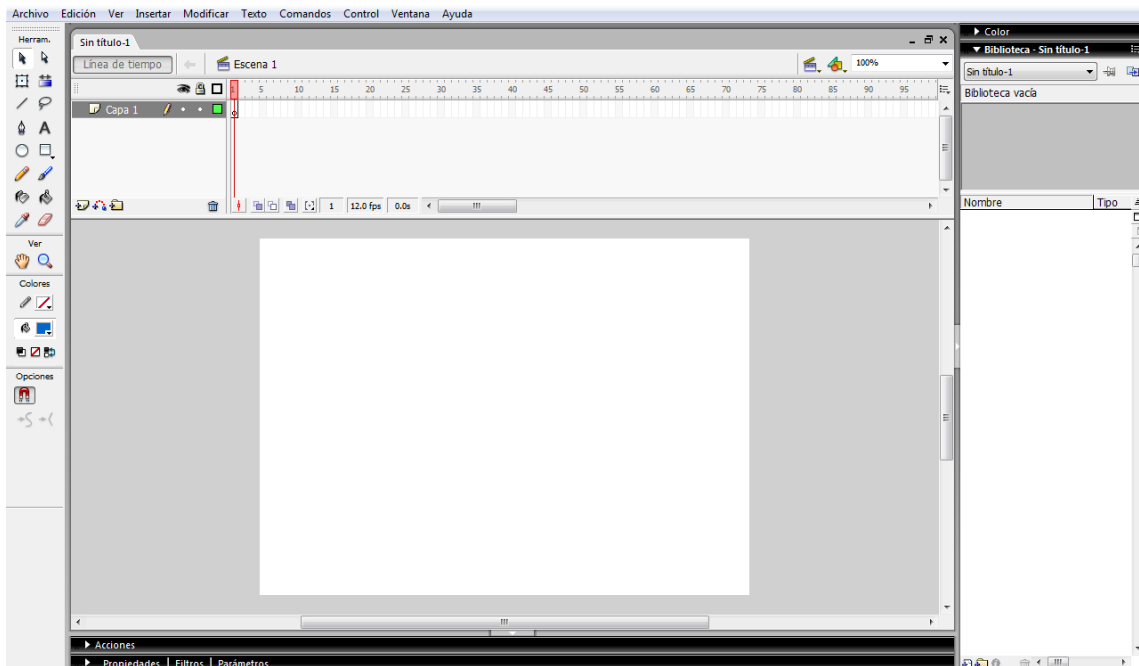


Figura 7: Macromedia Flash CS3

Macromedia Flash permet realitzar aplicacions gestionades per dades, animacions com anuncis publicitaris, targetes de felicitació, tot tipus de jocs, animacions que continguin vídeo, interfícies d'usuari, com pot ser el menú d'una pàgina web o aplicacions dinàmiques d'Internet. Aquestes últimes ofereixen un ampli de possibilitats, on es crea una aplicació que proporciona al usuari una interfície gràfica per a mostrar i manipular dades emmagatzemades anteriorment en un servidor.

Es poden crear aplicacions de Flash amb una ampla varietat i diversitat de contingut multimèdia que incloguin imatges, sons, vídeo i efectes especials de tot tipus.

Flash és la forma ideal de crear una aplicació amb una interfície gràfica prou atractiva per a que l'usuari interactui de forma senzilla. Flash ofereix una llibertat total a l'hora de col·locar i ubicar tots els elements en el espai de l'escenari de l'aplicació.

Tots els arxius que es creen en Flash, es creen en el format SWF que posteriorment s'han d'incloure en una pàgina web HTML per a mostrar-ho a la resta d'usuaris.

Per a que un usuari pugui visualitzar correctament l'aplicació, és necessari que el seu ordinador tingui instal·lat prèviament el Flash Player corresponent. Cal dir que això no suposa cap problema a la hora de programar, ja que la majoria d'usuaris ja disposen d'aquest element instal·lat en els seus ordinadors.

### ***5.2.2.2 Macromedia Flash i el Projecte***

Gràcies a la combinació entre animacions Flash i la programació en ActionScript, podem aconseguir el resultat desitjat en el projecte.

Per a crear una aplicació en Flash, es pot fer de dues formes. La primera és crear els elements i gràfics de l'aplicació amb les eines de dibuix que ofereix el programa sobre la mateixa línia del temps, importar els elements multimèdia desitjats i finalment, determinar en la línia del temps el moviment o ubicació d'aquests per a aconseguir l'aplicació desitjada. O per una altra banda, tal i com s'ha fet en la realització d'aquest projecte, es poden crear la major part dels elements o gràfics mitjançant programació en ActionScript. Aquest mètode és molt més complex que l'anterior, però alhora permet accedir a un ventall de possibilitats molt més ampli, ja que quan es realitza una aplicació 'gran' és fonamental portar un cert ordre i control de tot el que conté l'aplicació.

Un altre avantatge alhora d'utilitzar Flash per el projecte, és que s'utilitzen gràfics vectorials que requereixen molta menys memòria i espai d'emmagatzematge, ja que estan representades mitjançant formules matemàtiques, enlloc de grans conjunts de dades. Aquest punt és molt important en el nostre projecte, ja que al estar realitzant consultes contínuament a la Api de Google, és totalment necessari aconseguir reduir el temps de càrrega.

D'aquesta forma, s'aconsegueix una interfície gràfica complexa i molt més atractiva per l'usuari, amb una mida d'arxius de càrrega ràpida, ja que hem de tenir molt en compte el fet que l'aplicació estarà online en un servidor i s'ha de tenir present que els usuaris disposen a vegades de connexions lentes a Internet.

### 5.2.3 *Sepy*

#### 5.2.3.1 *Característiques del Sepy*

El Sepy és un editor de codi ActionScript. És un programa molt senzill però útil per escriure codi en ActionScript, ja que, com a molts altres editors específics de cada llenguatge, disposa de l'opció d'autocompletar codi que ajuda i facilita l'escriptura de forma més àgil.

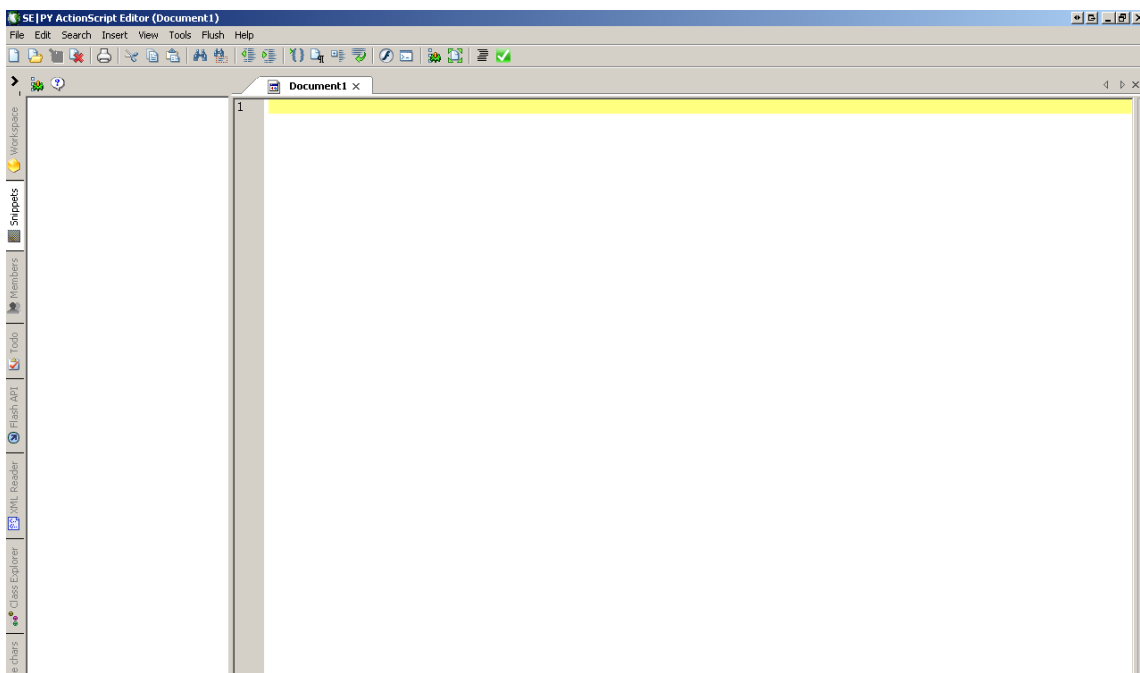


Figura 8: *Sepy*

## 5.2.4 Photoshop CS3

### 5.2.4.1 Característiques del Photoshop

El Photoshop és el programa més utilitzat alhora de dissenyar i realitzar edició de fotografies.

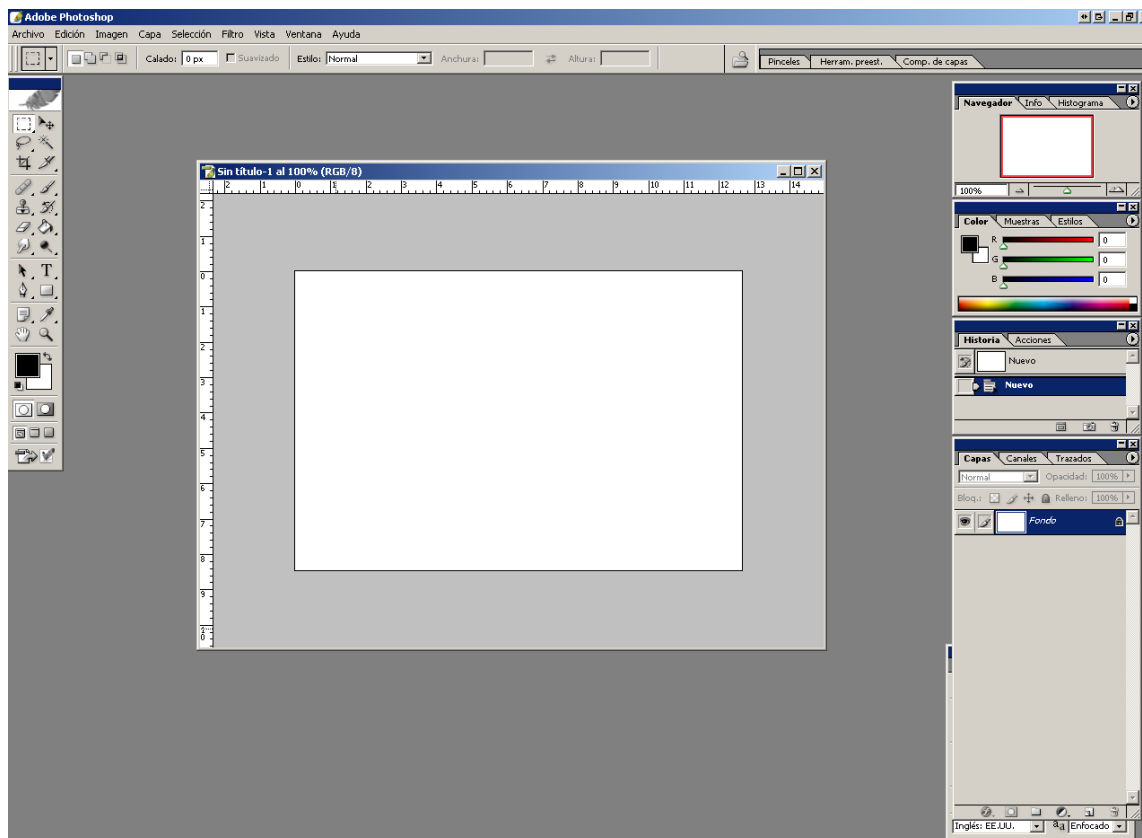


Figura 9: Photoshop CS3

És un programa per a la creació, edició i retoc de imatges. Disposa d'una gran varietat d'eines i possibilitats per a realitzar qualsevol efecte sobre fotografies o, simplement, és prou flexible per permetre realitzar el disseny de qualsevol tipus d'element, ja sigui una pàgina web o un cartell publicitari.

Photoshop ja no és una eina únicament feta servir per dissenyadors o maquetadors, Photoshop s'ha convertit en una eina molt utilitzada per a fotògrafs. Aquests el fan servir per a fer el procés de retoc i/o manipulació de la imatge, abans de fer-les arribar al laboratori per d'impressió d'aquestes.

#### 5.2.4.2 *Photoshop i el Projecte*

Per a realitzar el disseny de la interfície gràfica per a l'aplicació, era necessari fer un disseny previ, per a després reproduir aquest disseny perfectament detallat, a l'aplicació programada en ActionScript.

El Photoshop proporciona les eines necessàries per a fer un disseny simple, treballat i alhora amigable per l'usuari.

Primerament, el que es fa, és decidir de quina mida serà l'aplicació i es crea un nou document en blanc amb les mides exactes. Mitjançant les regles, marquem els límits de cada part de l'aplicació; amb les eines de disseny anem perfeccionant i detallant cada element que formarà part de l'aplicació.

És important que cada element estigui ben detallat en una capa diferent i amb una etiqueta identificativa per a portar un control de cada element. També es pot aplicar un ampli nombre d'efectes a cadascuna de les capes i elements del disseny.

Un cop tenim el disseny perfectament acurat, ordenarem tots els elements en la biblioteca per tal que després, sigui més fàcil seleccionar elements que siguin necessaris per a retallar i guardar com a JPG, PNG o GIF, i importar-los a la biblioteca de Macromedia Flash, per tal de poder-los fer servir en les crides en ActionScript. A vegades, segons l'efecte d'ombrejat o degradat de color, surt més a compte importar l'element, que no realitzar la programació de l'efecte desitjat en Flash, ja que a part d'estalviar-se recursos, també aconseguim reduir pes final de l'aplicació. És molt important anar amb compte amb el pes de cada element que importem al Flash, per això és convenient fer servir formats d'arxiu adaptats a web, com pot ser el PNG o el GIF.



## 5.2.5 PhpMyAdmin

### 5.2.5.1 Característiques del PhpMyAdmin

PhpMyAdmin és un programa que permet accedir a totes les funcions d'una BBDD MySQL a través d'una interfície web, és a dir, un gestor de Bases de dades. L'aplicació ens permet crear taules, inserir dades en taules existents, navegar per els registres de les taules, editar-los i esborrar-los, eliminar taules, etc. També ens permet executar sentències SQL per a realitzar totes les accions.

Totes aquestes accions, es realitzen des d'una interfície web que prèviament s'han instal·lat en el nostre servidor web.

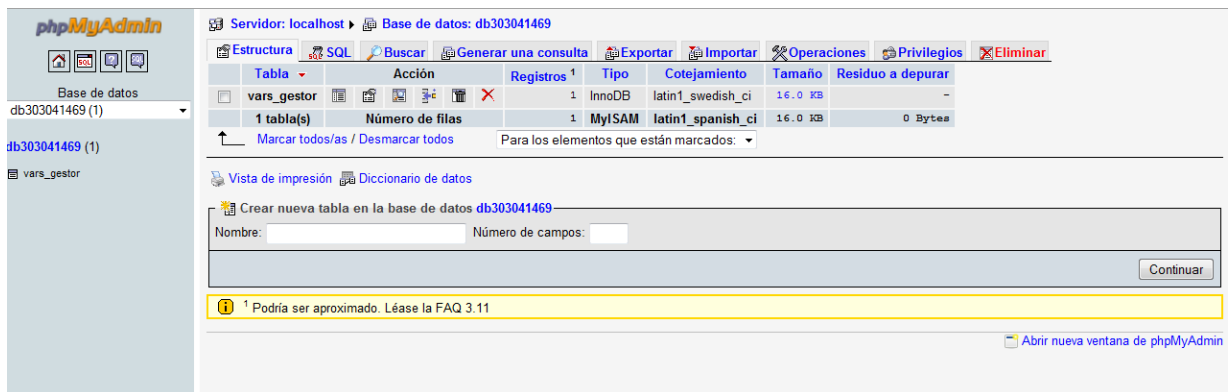


Figura 10: PhpMyAdmin

### 5.2.5.2 PhpMyAdmin i el Projecte

Per a realitzar el projecte, en segons quin moment de l'execució de l'aplicació, es pot necessitar l'accés a una Base de dades, per a guardar de forma encriptada certa informació que no pot ser accessible públicament. El fet que es decideixi encriptar aquesta informació, és per que es tracta d'informació personal i és convenient evitar que possibles 'hackers' accedeixin a ella.

Cal remarcar també, el fet que PhpMyAdmin és un programa de lliure distribució en PHP i es pot fer servir des de qualsevol ordinador. Únicament és necessari haver instal·lat prèviament uns arxius en el servidor web.

Per accedir al PhpMyAdmin, només cal escriure en un navegador web el domini seguit de /phpmyadmin. En el cas del projecte de final de carrera, aquest accés podria ser: [www.tfc2009.e/phpmyadmin](http://www.tfc2009.e/phpmyadmin)

## 6. Google Analytics amb Flash

### 6.1 Entrada al Gestor

#### 6.1.1 Introducció

Per accedir a l'aplicació del projecte, l'usuari ha de disposar d'un navegador web (*Firefox*, *Internet Explorer*, *Safari*, *Chrome* o qualsevol altre) i connexió a Internet.

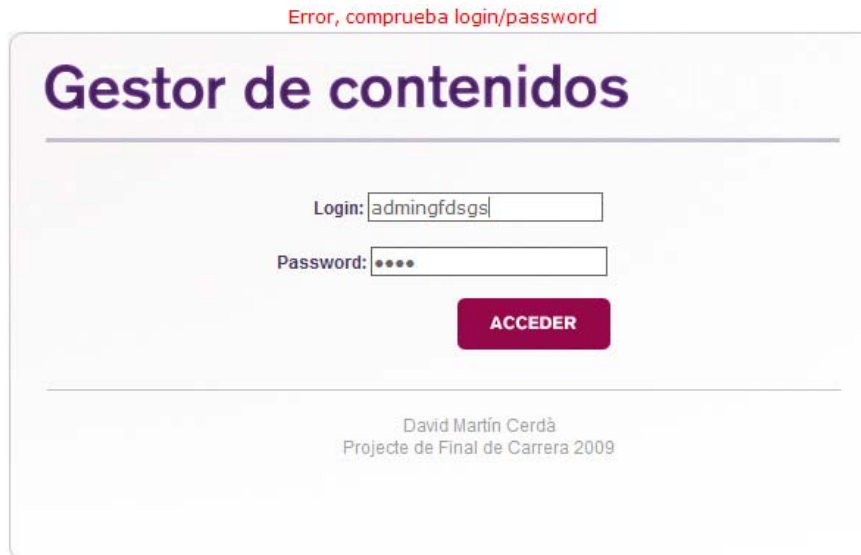
Cada usuari, per accedir a l'aplicació, disposarà d'un *login* i *password* assignat. L'usuari haurà d'escriure el seu *login* i *password* en l'entrada del gestor per identificar-se. Normalment, aquest accés es realitzarà mitjançant: [www.domini-lloc-web.com/gestor](http://www.domini-lloc-web.com/gestor). És necessari aquest accés a un gestor ja que inicialment, l'usuari només disposarà de l'opció de l'aplicació de Google Analytics en Flash, però no es descarta que en un futur es puguin oferir més opcions. Per tant, és necessària una pantalla d'accés a un gestor on, un cop l'usuari hagi escrit correctament el seu *login* i *password*, accediran a les diferents opcions.

El fet que cada usuari disposi d'un *login* i *password* és principalment per seguretat; al tractar-se d'una aplicació per a un client, és important que únicament el client tingui coneixement d'aquest *password*. En un principi, només hi haurà un únic usuari i contrasenya, però si el client ho sol·licités, es podrien crear tants accessos com sigui necessari.



Figura 11: Entrada al gestor de continguts

En cas que l'usuari no introdueixi correctament el seu *login* i *password*, es mostrarà per pantalla una alerta indicant aquest fet.



The image shows a web form for a content manager. At the top, there is a red error message: "Error, comprueba login/password". Below this, the title "Gestor de contenidos" is displayed in a large, bold, purple font. The form contains two input fields: "Login:" with the text "adminfdsgs" and "Password:" with four dots. A red button labeled "ACCEDER" is positioned below the password field. At the bottom of the form, the text "David Martín Cerdà" and "Proyecto de Final de Carrera 2009" is visible.

---

Si se presenta alguna incidencia con el backoffice, contacte con nosotros en [info@emascarotourism.com](mailto:info@emascarotourism.com) o llame al 902932774.  
by eMascaró Tourism - [www.emascarotourism.com](http://www.emascarotourism.com)

*Figura 12: Error en l'accés al gestor de continguts*

En cas que aquestes dades siguin correctes, accedirem a la següent pantalla, on l'usuari tindrà l'accés a l'aplicació de Google Analytics en Flash.

### **6.1.2 Argumentació del codi font**

Per a realitzar l'accés al gestor, l'usuari haurà d'introduir un *login* i un *password*. En aquest cas, aquest accés el fan mitjançant un usuari i contrasenya prèviament emmagatzemats en el codi font PHP de l'accés al gestor. Es podria haver optat per emmagatzemar aquesta informació en una Base de Dades, però finalment, es va considerar que era prou mesura de seguretat emmagatzemar-ho en el mateix codi font de forma encriptada. Aquesta només és una primera mesura de seguretat, ja que, tot i accedint amb

el *login* i *password* correctes, en el següent punt d'accés, l'usuari haurà d'introduir un altre *login* i *password* personal que correspondrà a un compte d'email

Un cop l'usuari ha introduït el seu *login* i *password* i prem el botó d'Accedir, el formulari crida a la funció: **login.php**.

```
@session_start();

$login=md5($_POST["login"]);
$password=md5($_POST["password"]);

if ($login=="21232f297a57a5a743894a0e4a801fc3" && $password=="21232f297a57a5a743894a0e4a801fc3")
{
    $_SESSION["idioma_edicion"]="es";
    $_SESSION["idioma"]="es";
    @$_SESSION["bk"]="1";
    ?>
    <html>
    <head>
    <script language="javascript">
    document.location.href="web/negocio/bk/mant_buscadores.php";
    </script>
    </head>
    </html>
    <?php
} else {

    header("location: index.php?error=".urlencode("Error, comprueba login/password"));
```

Podem observar com el que primer fem, és iniciar la sessió de servidor. `@session_start()`;

A continuació, mitjançant el mètode POST de recollida de dades d'un formulari (pot ser: POST o GET, cadascuna de les dues té unes característiques pròpies), es recull el *login* i el *password*, i tot seguit s'encrypta en MD5 (un tipus de encriptació).

```
$login=md5($_POST["login"]);
```

```
$password=md5($_POST["password"]);
```

Tot seguit, és necessari comprovar si el *login* i *password* introduïts corresponen l'usuari i contrasenya encryptats en el moment de la programació de l'accés web.

```
If ($login=="21232f297a57a5a743894a0e4a801fc3" &&
```

```
password=="21232f297a57a5a743894a0e4a801fc3")
```

En el cas que siguin correctes les dades d'identificació, es procedirà a emmagatzemar diverses dades en la sessió de servidor i a continuació, a redirigir a l'usuari a la pàgina d'accés següent.

```
$_SESSION["idioma_edicion"]="es";
$_SESSION["idioma"]="es";
 @$_SESSION["bk"]="1";

<html>
<head>
<script language="javascript">
document.location.href="web/negocio/bk/mant_buscadores.php";
</script>
</head>
</html>
```

En cas contrari, si les dades introduïdes no són correctes, es llençarà un missatge d'error.

```
header("location: index.php?error=".urlencode("Error, comprueba login/password"));
```

## 6.2 Entrada al Panell d'accés

### 6.2.1 Introducció

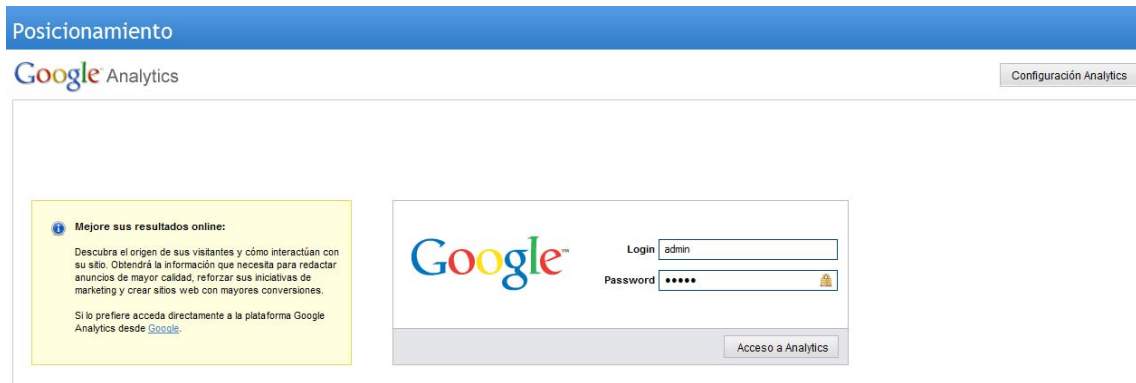


Figura 13: Entrada al panell d'accés

Un cop l'usuari ha accedit correctament al gestor, s'accedeix a un altre control de seguretat. En aquest cas, ens trobem en la pàgina on s'han d'introduir les dades del compte de Google Analytics, per a realitzar la connexió i accedir al panell amb les dades de l'aplicació en Flash de Google Analytics.

En aquest punt, s'ha de realitzar una acció prèvia; abans d'introduir el *login* i *password*, s'ha d'introduir el codi identificador del compte de Google Analytics del lloc web del qual es vol obtenir les dades. Per a portar a terme aquesta acció, accedirem al botó situat a la part superior dreta de la pantalla. Fem clic sobre el botó: “*Configuración Analytics*”.

Configuración Analytics

Observem com es mostra per pantalla un nou camp a omplir per tal de poder configurar correctament l'aplicació.

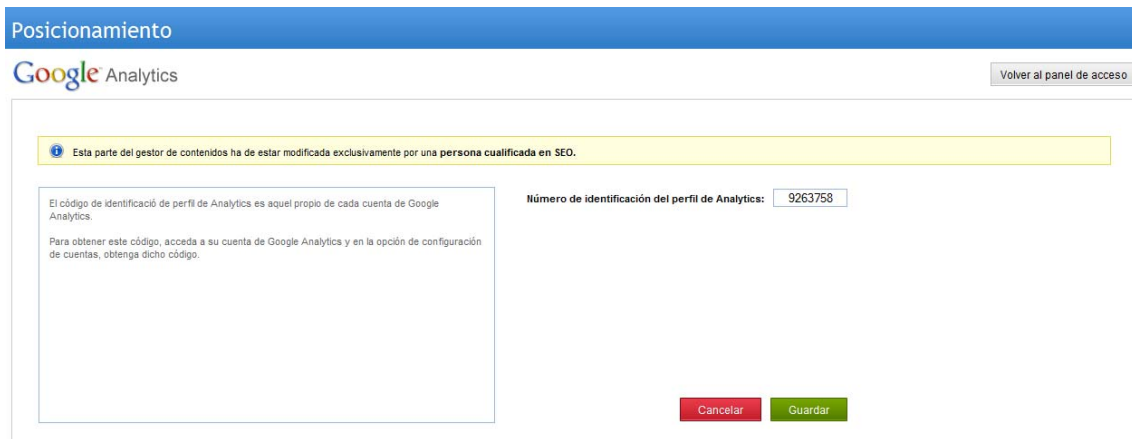


Figura 14: Configuració de la compte

En la capsa de text de l'esquerra, apareixerà escrita una sèrie d'instruccions i/o informació complementaria per informar al usuari sobre que és el que ha de realitzar. S'informarà de com obtenir aquest codi en el seu compte de Google Analytics.

En la capsa de text de la dreta, és on introduïrem el codi identificatiu de cada compte.

**Número de identificación del perfil de Analytics:**

Un cop introduït aquest codi, mitjançant el botó “*Guardar*”, s'emmagatzemarà la informació en la base de dades, per tal que no es tingui que tornar a introduir en posteriors accessos a l'aplicació.

En aquest cas, al tractar-se d'un codi personal de cada usuari i compte relacionat, aquesta dada l'emmagatzemarem en base de dades i encriptada per a major seguretat.

Quan ja s'ha introduït i guardat el codi d'identificació de cada compte, tornarem a la pantalla anterior on es podia introduir el *login* i *password* del compte de Google Analytics, mitjançant el botó de “*Volver al panel de acceso*”.

En la pantalla principal, introduïrem l'usuari i contrasenya del compte de Google Analytics, i a continuació, accedirem al panell en Flash de Google Analytics.







```
###codi HTML###
```

```
}else{
```

```
###codi HTML###
```

### Codi de la pàgina del panell d'accés

En aquest punt, el codi es tracta d'un formulari que realitzarà l'enviament de les dades d'accés introduïdes per l'usuari. Observem com es fa servir el mètode POST un altre cop. També estem guardant una variable amb el valor *connect* per a poder realitzar comprovacions posteriors.

```
<form name="form" action="" method="post">
<input type="hidden" name="pagina" value="connect" />
```

Observem com el camp *hidden* (camp ocult) amb el nom: *página*, té com a valor: **connect**, per tant, quan l'usuari ompli els camps del formulari i realitzi l'acció d'enviar, la variable **\$p\_anal** passarà a tenir el valor de **connect** i per tant, al recarregar-se la pàgina enlloc d'entrar en el primer **if** mostrat anteriorment, aquest accedirà al primer **elseif**. Aquest correspon a l'acció de mostrar la pantalla de l'aplicació Flash, ja que es considera que ja s'ha introduït el *login* i *password*.

### Codi de la pàgina de aplicació en Flash

El codi de l'aplicació en Flash s'explicarà en un altre punt, ja que, com s'ha comentat anteriorment, aquesta aplicació està feta amb Flash i ActionScript 3.

Tot i així, en el PHP que estem analitzant (**mant\_buscadors.php**), s'escriu el codi HTML i PHP necessari per incloure l'arxiu Flash (SWF) a la pàgina web.

```
var mymovie = new FlashObject("../plugins/analytics/analytics.swf?php=../plugins/analytics/flash_analytics.php&id=<?php
echo $ga_id; ?>&login=<?php echo $ga_login; ?>&password=<?php echo $ga_password; ?>&key=<?php echo $key; ?>", "pelicula_analytics",
"100%", "100%", "7", "");
mymovie.addParam("quality", "high");
mymovie.addParam("wmode", "transparent");
mymovie.addParam("bgcolor", "#ffffff");
mymovie.addParam("align", "middle");
mymovie.addParam("allowScriptAccess", "sameDomain");
mymovie.write("analytics");
```

Aquest tros de codi, correspon al codi necessari per incloure el Flash a la pàgina. Observem que es fa servir la funció: **FlashObject**. Aquesta és la que s'encarrega d'incloure el Flash.

```
var mymovie = new  
FlashObject("../plugins/analytics/analytics.swf?php=../plugins/analytics/flash_analytics.php&id=<?php echo $ga_id; ?>&login=<?php echo $ga_login; ?>&password=<?php echo $ga_password; ?>&key=<?php echo $key; ?>", "pelicula_analytics", "100%", "100%", "7", "");
```

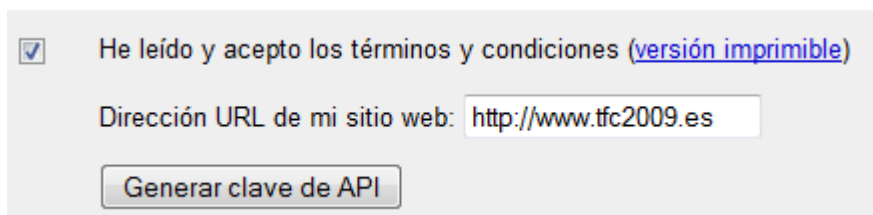
En negreta, observem un conjunt de variables que se li passen al Flash per tal que l'ActionScript sigui capaç de, posteriorment, realitzar les crides necessàries per a realitzar la captura de dades. Aquestes variables són:

- **\$ga\_id**: es tracta del identificador del compte de Google Analytics anteriorment introduït per l'usuari.
- **\$ga\_login** i **\$ga\_password**: aquestes dues variables són l'usuari i contrasenya amb els quals l'usuari està intentant accedir a l'aplicació de Google Analytics en Flash.
- **\$key**: aquesta variable és una sèrie de números i lletres que Google necessita per a tenir un control dels llocs web que accedeixen a dades seves. En el nostre cas, ho necessitarem per incloure el Mapa d'ubicació de Google Maps.

Per aconseguir la Api Key, únicament és necessari accedir a:

<http://code.google.com/intl/es-ES/apis/maps/signup.html>.

En aquesta URL web, Google ens permetrà aconseguir una Api Key, únicament serà necessari acceptar els termes i condicions legals, i introduir el domini web en el qual tindrem inclòs el mapa de Google Maps.



He leído y acepto los términos y condiciones ([versión imprimible](#))

Dirección URL de mi sitio web:

Un cop s'ha generat la Api Key, Google et retorna el teu codi:

## Registro en el API de Google Maps

Thank You for Signing Up for a Google Maps API Key!

Your key is:

```
ABQIAAAAE7mJR4J90-hpA_jx1VfKDNhQ67Bw8PBcTQu3RIDMrPSUqidnapBSYRIIQGfHHxIXkCY0eXQ3e66yL5w
```

La resta de paràmetres que es poden observar, són únicament necessaris per a configurar correctament el fet d'incloure un arxiu Flash (SWF) en una pàgina web.

```
mymovie.addParam("quality", "high");
mymovie.addParam("wmode", "transparent");
mymovie.addParam("bgcolor", "#ffffff");
mymovie.addParam("align", "middle");
mymovie.addParam("allowScriptAccess", "sameDomain");
mymovie.write("analytics");
```

Finalment, en aquest tros de codi també trobem una funció, la qual el Flash cridarà en el cas que el *login* i *password* que l'usuari a introduït no siguin correctes.

```
function login_error(){
    alert("Connexión abortada: login y/o password incorrecto.\n Intentenlo de nuevo.");
    document.location.href="mant_buscadores.php?acc=analytics";
}
```

## **Codi de la pàgina de petició de l'identificador del compte de Google Analytics**

En aquest punt, el codi es tracta bàsicament d'un camp de text amb una explicació breu sobre el que l'usuari haurà de realitzar, un altre camp de text per introduir el identificador i un botó guardar que realitzarà l'acció d'emmagatzemar l'identificador.

En el següent tros de codi, observarem com es realitza la declaració del formulari en HTML; podem veure com la acció del formulari, un cop l'usuari premi el botó “*Guardar*” o “*Enviar*” segons la ocasió, el formulari automàticament cridarà al PHP:

**editor\_analytics2.php** que s'encarregarà de realitzar l'acció de guardar l'identificador introduït per l'usuari.

```
<form name="form" action="editor_analytics2.php" method="post">
```

Aquest PHP: **editor\_analytics2.php**, és l'arxiu encarregat de modificar el camp corresponent al identificador de la base de dades.

```
set_data("UPDATE vars_gestor set
valor=AES_ENCRYPT('".$_POST["g_code"]."',".KEY."') WHERE
clave='code_analytics';");
```

En aquest cas, és imprescindible emmagatzemar aquesta dada en una BBDD i guardar-la encriptada per evitar possibles atacs de *hackers*.

Prèviament, s'ha d'haver configurat la BBDD. Únicament necessitarem una taula amb un camp: **code\_analytics**.

```
CREATE TABLE IF NOT EXISTS `vars_gestor` (
  `id` tinyint(3) unsigned NOT NULL AUTO_INCREMENT,
  `clave` varchar(50) DEFAULT NULL,
  `valor` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

Podem observar com el identificador un cop en la BBDD, queda emmagatzemat encriptat.

+ Opciones			
	id	clave	valor
<input type="checkbox"/>	1	code_analytics	¬fóðñĚĚ/–ŪÆĪP=Ū

## 6.3 Panell Google Analytics en Flash

### 6.3.1 Introducció

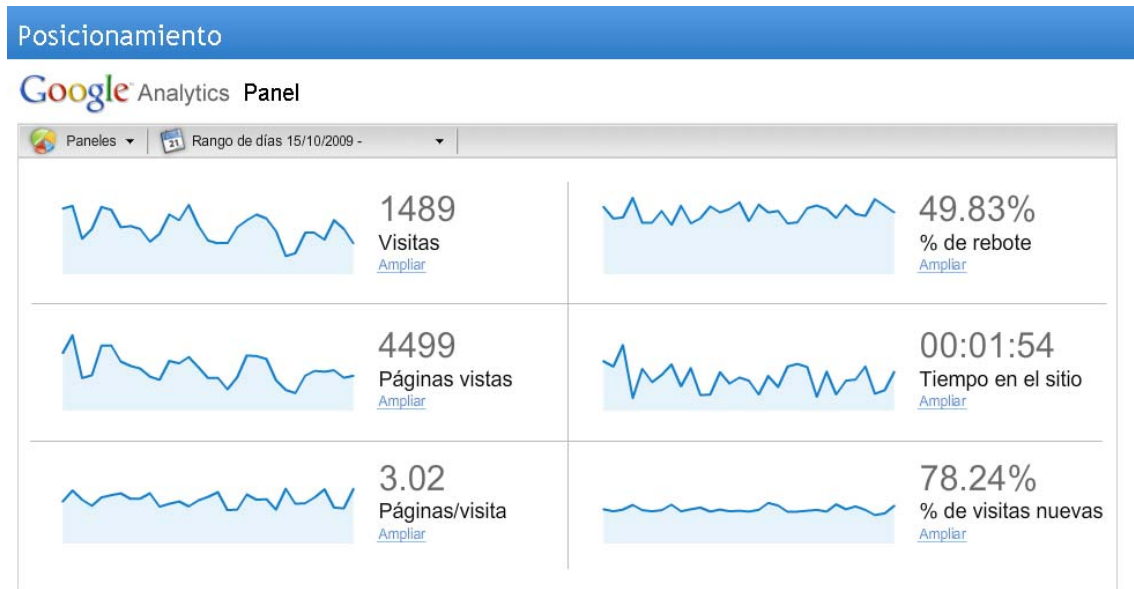


Figura 16: Panell Google Analytics en Flash

Quan l'usuari ja ha introduït correctament el *login* i el *password* del seu compte de Google Analytics, accedim al panell de Google Analytics en Flash. Observem com d'entrada, a primer cop d'ull, es poden veure les principals dades del lloc web en el període de temps predeterminat.

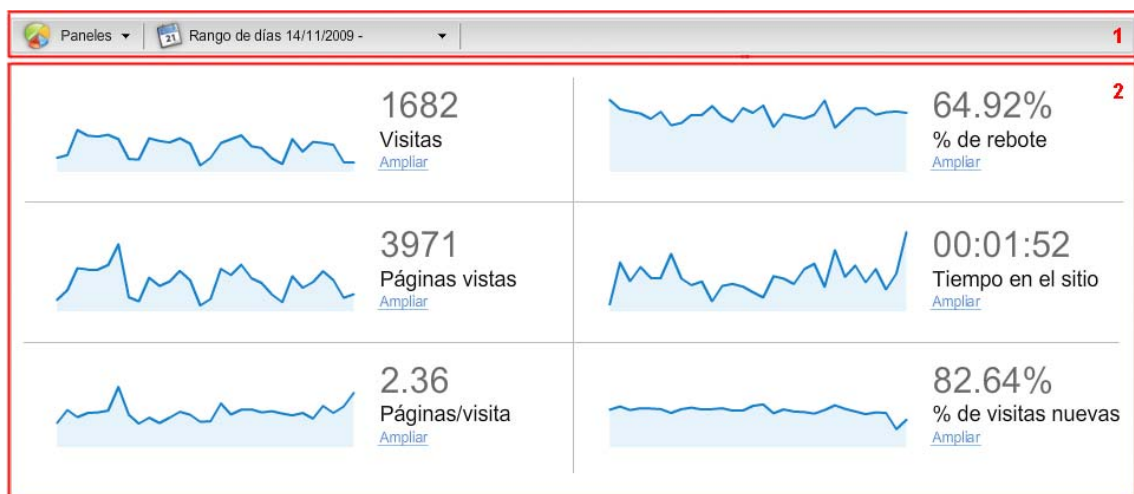


Figura 17: Zones del panell Google Analytics en Flash

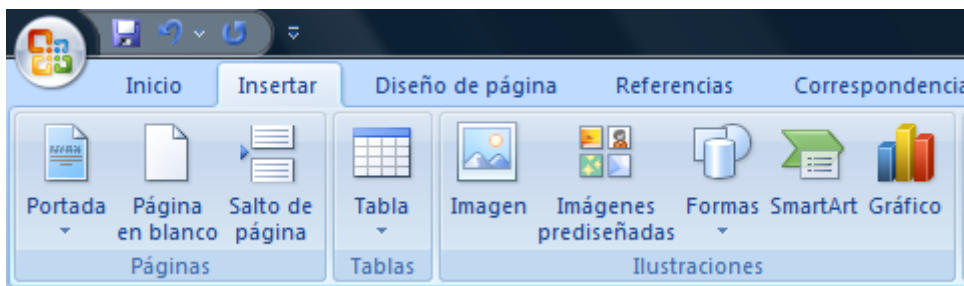
Com podem observar en la imatge superior, es pot diferenciar dues zones principals.

1. Zona de menú amb les diferents opcions de menú i zona de modificació del rang de període de temps.
2. Zona del panell central on es mostren totes les dades corresponents a la tria d'opcions del menú superior.

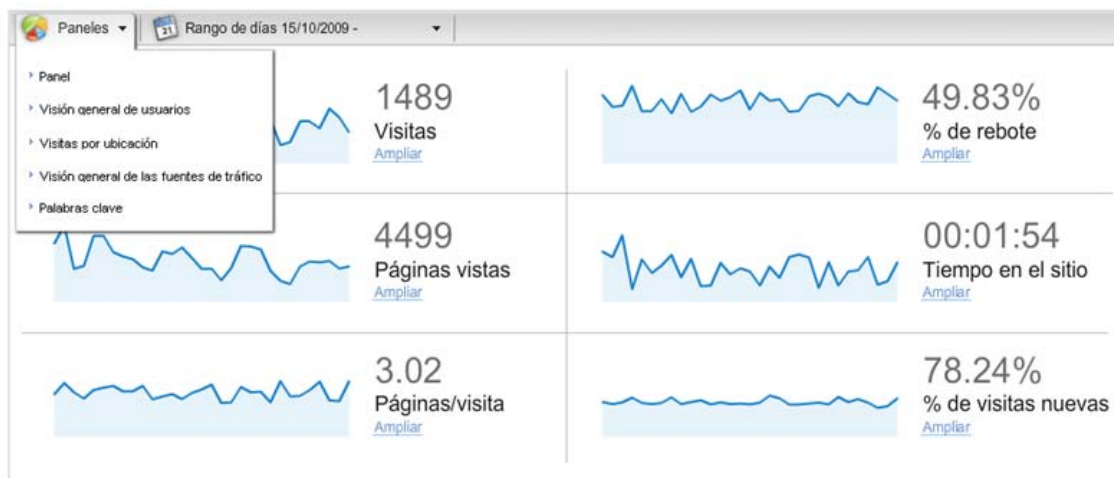
Observem com la interfície gràfica d'aquesta aplicació és molt simple e intuïtiva, per a facilitar al màxim la navegació per a l'aplicació a l'usuari que en teoria té un nivell baix de coneixements informàtics i de la matèria.



El menú superior és en forma de “menú desplegable”, ja que es va considerar l'opció de fer-lo amb pestanyes com per exemple el menú del nou Microsoft Word:



Tot i ser una forma de navegar força usable, es va considerar que era millor fer-ho d'una forma més clàssica i es va optar per a realitzar-ho en forma de menús desplegables.





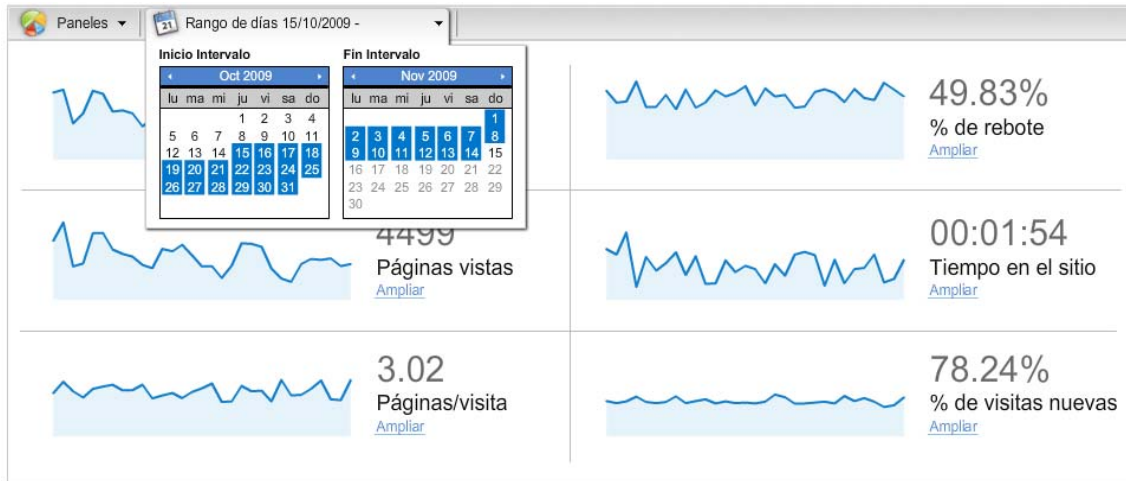


Figura 18: Calendari del panell Google Analytics en Flash

En la opció del menú de “Paneles”, es mostren totes les opcions que prèviament s’han definit per a incloure en la aplicació.

Cal recordar el fet que aquesta és una aplicació per a clients. Cada client demanarà unes opcions diferents, per tant, aquest menú serà diferent per a cada client.

Per una altre banda, en aquesta zona superior del menú, també es pot triar el dia o rang de dies del qual es vol visualitzar les dades. Quan es fa una tria de data, aquesta es mantindrà encara que es triï una altre opció del menú.

## 6.3.2 Navegant per l’aplicació

### 6.3.2.1 Panell principal

En el panell principal, es mostra un resum de les dades de les estadístiques principals de la web.

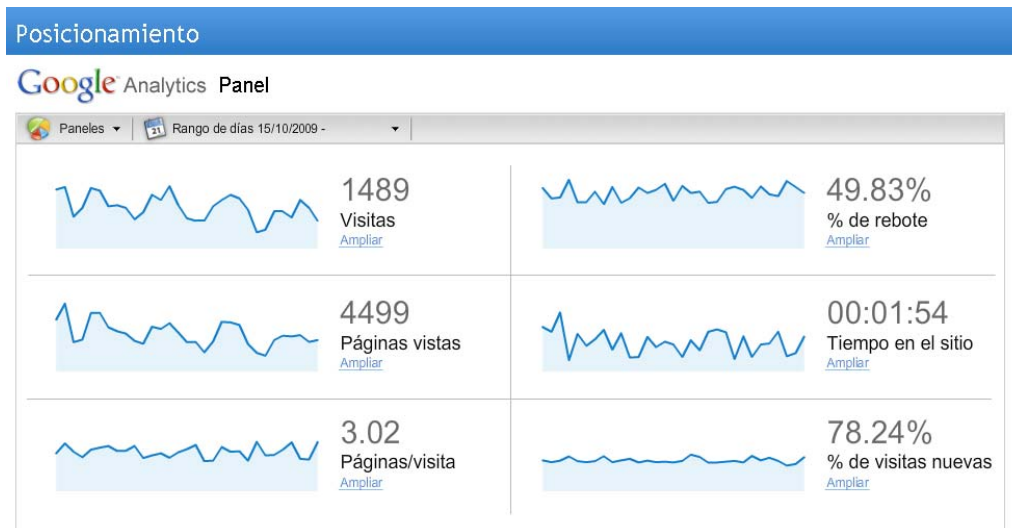


Figura 19: Panell principal Google Analytics en Flash

- **Visites:** Les visites són el nombre d'usuaris totals que accedeixen a la pàgina web. Aquestes visites poden vindre de buscadors, de altres llocs webs o simplement perquè l'usuari ha escrit la direcció web en el seu navegador.
- **Pàgines vistes:** Les pàgines vistes és el nombre total de pàgines que els usuaris que han entrat en el lloc web i han estat navegant per ell. Aquesta dada ens pot indicar si els usuaris estan interessats o no en el contingut en general que ofereix la pàgina web.
- **Pàgines/visita:** Aquesta dada mostra la mitja de pàgines vistes per a cada usuari que accedeix al lloc web. Cal entendre, que contra més alta sigui aquesta dada, significarà que l'usuari realment està interessat en el contingut de les pàgines del nostre lloc web, i per tant, es converteix en un lloc web d'interès.
- **% de rebot:** Aquest terme també pot rebre el nom de: percentatge d'abandonament. Es tracta del percentatge de visites que arriba a la web, només mira la primera pàgina i tot seguit marxa la web. Aquest és un terme una mica difícil d'avaluar, ja que es pot donar el cas que la gent entra a la web i al no trobar el que desitja, llavors marxa de la web. Però també pot donar-se el cas que sigui tot el contrari, que l'usuari al entrar a la web, enseguida trobi la informació que desitja

i per tant, abandoni la web. Per tant, segons el tipus de web que s'analitzi, s'haurà de tractar d'una forma o altra aquesta dada.

- **Temps en el lloc:** Aquest paràmetre indica el temps mig que els usuaris que visiten la pàgina web estan en el *site*. Es tracta d'un altre paràmetre que, contra major sigui, ens indicarà que l'usuari està més interessat en el contingut de la pàgina web.
- **% de visites noves:** El percentatge de visites noves és un paràmetre que indica el percentatge de nombre de visites que accedeixen a la web per primer cop. D'aquesta forma es pot analitzar si l'usuari que accedeix a la web és un visitant recurrent o si per contra, es tracta d'un usuari nou.

L'usuari, des del mateix panell principal, pot accedir al gràfic ampliat corresponent a la dada que està interessat en consultar. Únicament és necessari que accedeixin mitjançant l'opció d'ampliar que hi ha en cada apartat del panell principal.

Un cop l'usuari accedeix a la informació ampliada de la dada que l'interessa, accedeix a una pantalla amb el gràfic corresponent. Aquest gràfic, està mostrat a una mida ampliada per a poder analitzar bé la variació entre els diferents dies de dins del període establert.



Figura 20: Gràfic ampliament detallat del panell principal de Google Analytics en Flash

L'usuari podrà moure's a través del gràfic. Si es situa el cursor sobre d'un node del gràfic (punt del gràfic corresponent a un dia en concret del període de temps), es podrà visualitzar la informació detallada d'aquell dia.

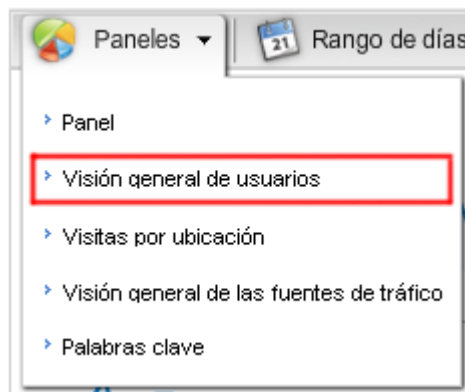


Observem com la informació que es mostra en la capsula de text, és el dia i el nombre de visites que la web ha tingut.

En qualsevol moment l'usuari pot tornar al panell principal mitjançant la opció: “*Volver al Dashboard*” situat a la part inferior dreta de la pàgina.

[← Volver al Dashboard](#)

### 6.3.2.2 Visió general d'usuaris



En aquesta segona opció del menú, accedim a un gràfic ampliat com l'explica't en el punt anterior (6.3.2.1. Panell principal).



Figura 21: Gràfic del panell visió general dels usuaris

En aquest cas, igual que en l'anterior, ens trobem amb un gràfic amb el nombre de visites que han accedit a la pàgina web en el període de temps establert.

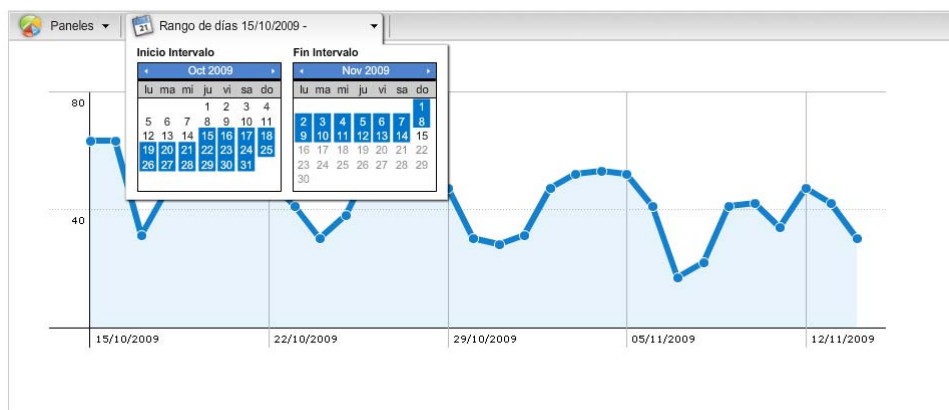
Podrem accedir al nombre de visites corresponents a cada node del gràfic.



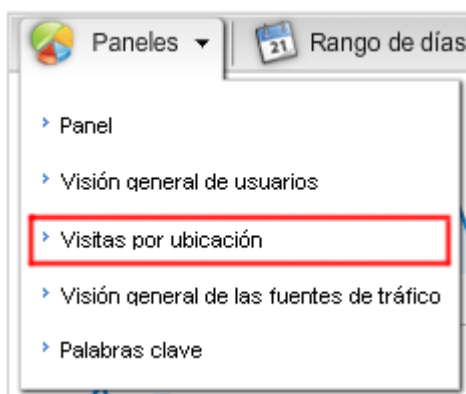
Figura 22: Detall del gràfic del panell visió general dels usuaris

En aquest cas, a diferència del gràfic de contingut ampliat del panell principal, no tindrem l'opció de "Volver al Dashboard", ja que en aquest cas ens trobem amb un gràfic únic i específic per aquesta opció del menú. S'ha considerat que la informació era prou important com per tindre-la present en una opció independent del menú.

Com en totes les opcions del menú de l'aplicació, l'usuari podrà triar el període de temps que desitgi visualitzar la informació per pantalla.



### 6.3.2.3 *Visites per ubicació*



Aquesta opció permetrà a l'usuari veure per pantalla un mapa del món amb la ubicació de cadascuna de les visites al lloc web.

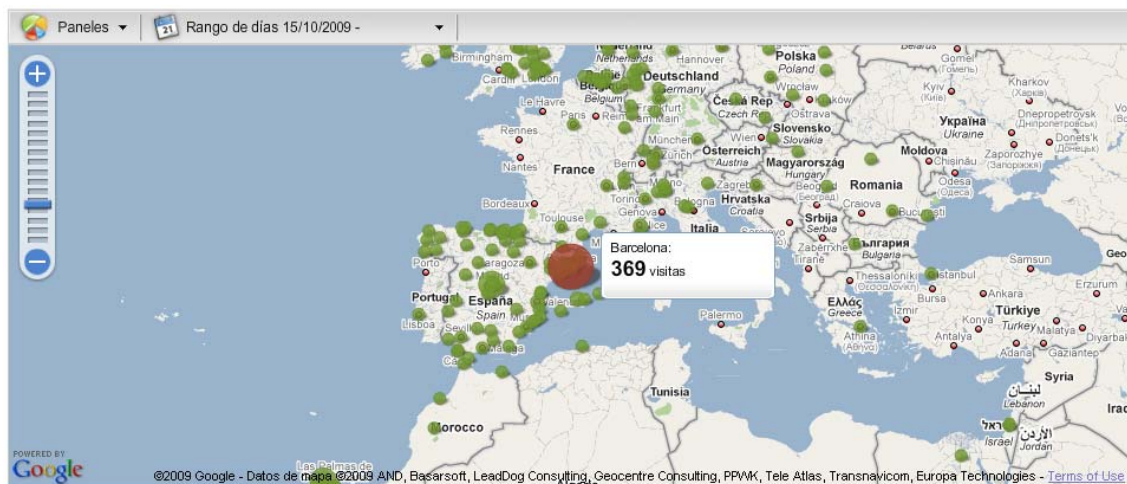


Figura 23: Mapa del panell de visites per ubicació

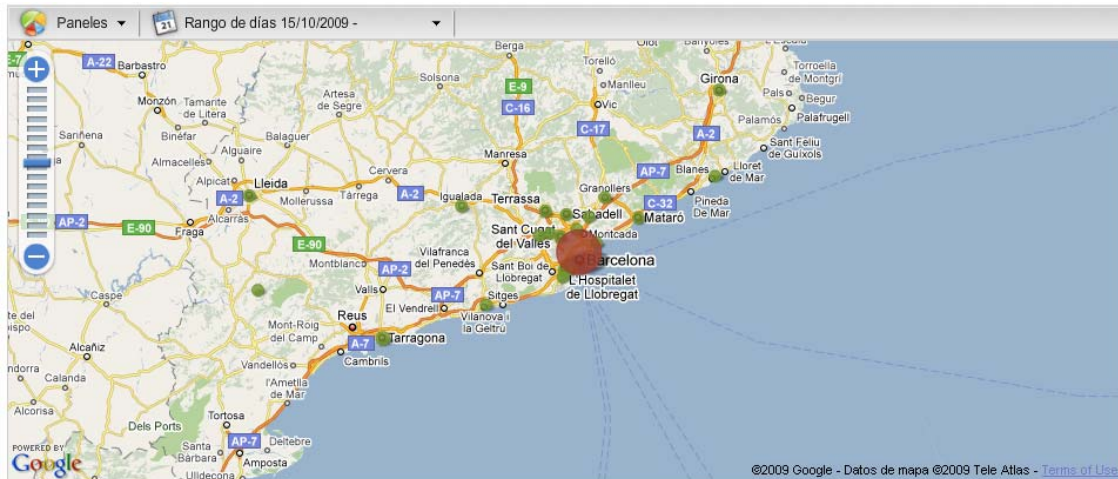
Es tracta d'un mapa de Google Maps integrat al Flash.

Observem com es marca amb un cercle de color verd tots els indrets dels quals hi ha visites. En vermell, i considerablement de més gran es marca l'indret que aporta un major nombre de visites a la web.

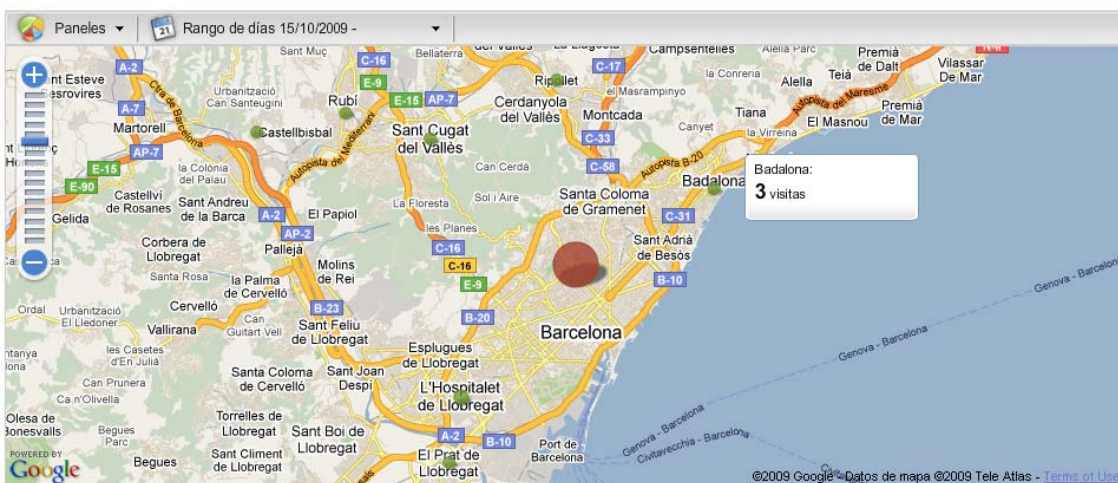
Si l'usuari situa el cursor sobre de qualsevol dels punts del mapa, es mostrarà una capa amb la informació de visites que hi ha del lloc seleccionat.



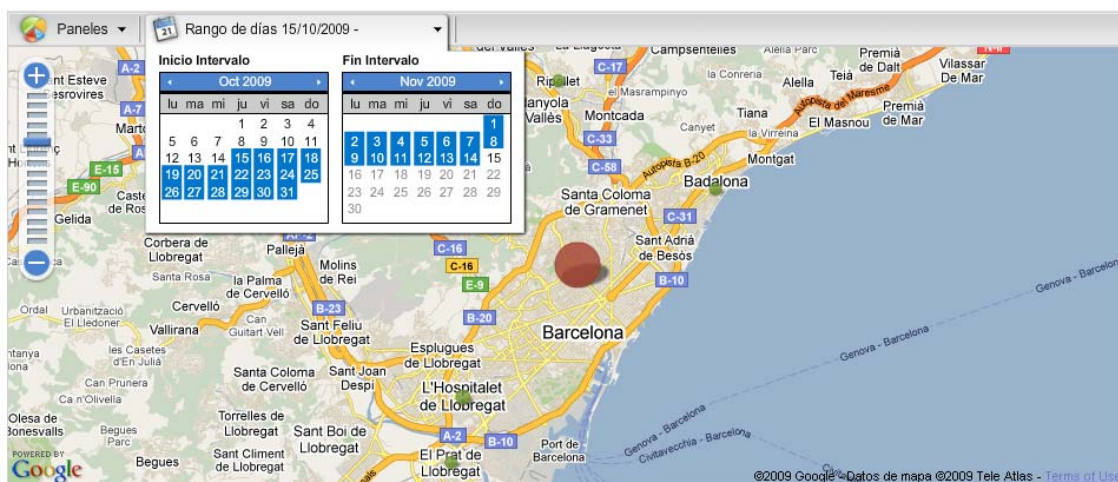
En el mapa, podem veure a la part superior esquerra un navegador. Aquest permetrà a l'usuari realitzar més o menys zoom per apropar-se o allunyar-se de cada zona per tal de veure amb més precisió la procedència de les visites.



Mitjançant el zoom es pot ampliar la zona per a poder visualitzar poblacions i el nombre de visites corresponents.



Igual que en la resta d'opcions, l'usuari podrà triar el període de temps que vol veure la ubicació de les visites de la web.





### 6.3.2.4 Visió general de les fonts de trànsit



En aquesta opció del menú, podrem observar un gràfic que ens mostrarà el total de les visites, les pàgines visitades o el temps en el lloc, en relació si les visites procedeixen de llocs web de referència (altres pàgines web que tenen un *link* directe a la nostra pàgina web), trànsit directe (l'usuari ha escrit directament la URL del lloc web en el seu navegador), o si procedeixen de motors de cerca (l'usuari ha realitzat una cerca a un buscador com, per exemple Google, i ha accedit a la web a través d'un resultat de la cerca feta).

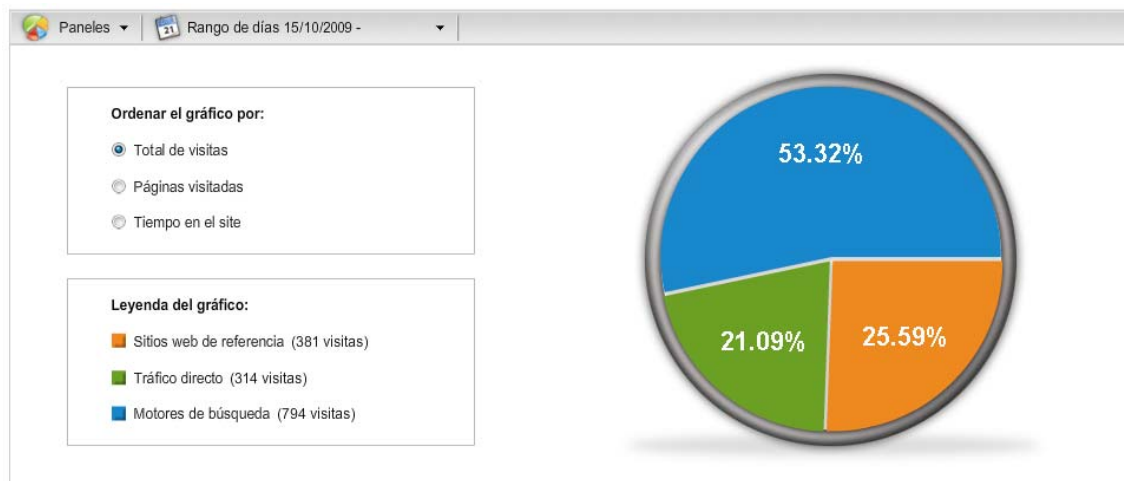
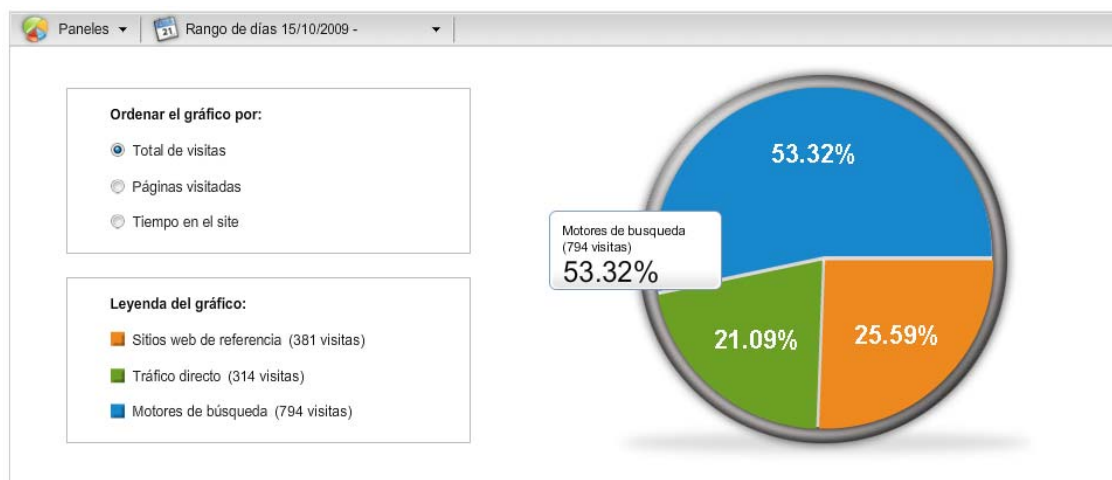


Figura 24: Gràfic del panell visió general de les fonts de trànsit

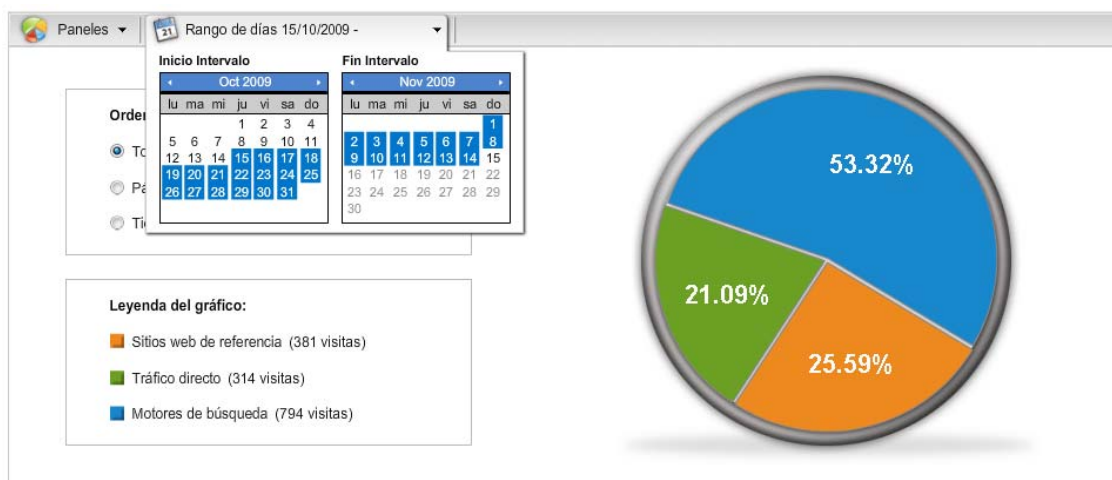
Des de la primera caixa podem triar si desitgem ordenar el gràfic segons el total de visites, el nombre de pàgines visitades o el temps en el lloc web.

En la segona caixa, ubicada just a sota de la caixa per ordenar el gràfic, ens mostra la llegenda de cadascuna de les parts del gràfic. També es pot observar el nombre de visites de cada opció.

El gràfic ens mostra el percentatge de cada part d'aquest. Si situem el cursor sobre de cada zona, se'ns mostrarà per pantalla el resum de dades d'aquesta opció. Podrem veure el nom de la opció, el nombre de visites i el percentatge; d'aquesta manera, se li facilita l'usuari la comprensió del gràfic, ja que d'un cop de vista pot veure tota la informació més rellevant.



Igual que en la resta d'opcions, l'usuari podrà triar el període de temps que vol visualitzar les diferents dades que es mostra en aquest gràfic.



### 6.3.2.5 Paraules clau



En aquesta última opció del menú, es mostra una taula amb totes les dades més importants sobre el llistat de paraules clau del lloc web.

Les paraules clau són aquells termes que els usuaris escriuran en els cercadors i permetre accedir a la pàgina web.

Posición	Keyword	Visitas	Páginas/visita	Promedio de tiempo	% visitas nuevas	% de rebote
1	alexandre hotels	180	4.08	00:02:33	60%	41.11%
2	alexandre hoteles	32	3.63	00:01:45	65.63%	34.38%
3	alexandrehotels	25	2.56	00:01:11	24%	56%
4	hoteles alexandre	18	4.39	00:01:50	55.56%	33.33%
5	hoteles con jacuzzi en la habitacion en tenerife	12	3.92	00:01:44	100%	33.33%
6	hoteles con jacuzzi en la habitacion tenerife	10	4.2	00:02:17	90%	20%
7	hotel con jacuzzi en la habitacion tenerife	9	2.78	00:00:52	77.78%	66.67%
8	www.alexandrehotels.com	9	3.33	00:02:26	77.78%	33.33%
9	habitaciones romanticas con jacuzzi	8	3.25	00:00:44	100%	50%
10	alexandrehotels.com	8	3.88	00:11:12	75%	25%

*Figura 25: Taula del panell de paraules clau*

En aquesta taula es mostra cada terme amb el nombre de visites a la web a través de la paraula cercada, el nombre de pàgines per visita mig de tota la gent que ha accedit a través de la paraula, així com el temps mig, el percentatge de visites noves i el percentatge de rebot a través de la paraula en concret.

És important tenir molt present aquests termes, ja que és la millor manera de conèixer com els usuaris accedeixen a la pàgina web. D'aquesta forma també podrem saber el que realment interessa als usuaris.

En aquesta última opció, també l'usuari podrà triar el període de temps del qual vol conèixer les paraules clau i la informació relacionada.

## 6.4 Codi font

A continuació s'explicarà pas a pas el codi font més important i rellevant de l'aplicació en Flash de Google Analytics. Per a explicar-ho d'una forma correcta i ordenada, s'explicarà de forma en que es va executant l'aplicació.

S'ha de tenir en compte que durant el procés d'execució s'alternarà programació en PHP i programació en ActionScript 3.

Cal recordar que aquesta explicació comença des de el punt que l'usuari introdueixi el seu *login* i *password* del seu compte de Google Analytics en el gestor i aquest executa el Flash.

En el moment d'incloure el Flash (arxiu SWF) es passen una sèrie de paràmetres amb la URL al Flash.

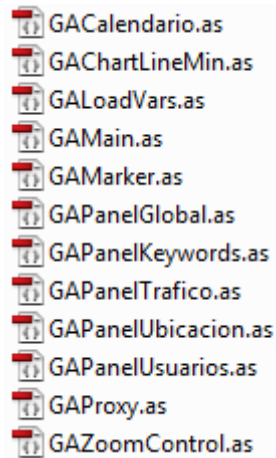
```
var mymovie = new FlashObject(
"../plugins/analytics/analytics.swf?php=../plugins/analytics/flash_analytics.php?id=<?php echo $ga_id; ?>#login=
<?php echo $ga_login; ?>#password=<?php echo $ga_password; ?>#key=<?php echo $key; ?>", "pelicula_analytics",
"100%", "100%", "7", "");
```

Concretament podem veure com es fan arribar al Flash el conjunt de dades anteriorment demanades al usuari: **\$ga\_id**, **\$ga\_login**, **\$ga\_password** i **\$key**.

Aquestes variables són recollides per el Flash i emmagatzemades en variables pròpies de ActionScript.

Quan s'inicia l'execució de l'arxiu Flash, aquest fa la crida a l'arxiu ActionScript que iniciarà tota l'execució.

Abans de continuar, és interessant presentar els arxius ActionScript més importants, que s'executaran i aportaran accions importants en el bon funcionament de l'aplicació:



Cada arxiu té una funcionalitat diferent. S'ha intentat separar al màxim cada funcionalitat per tal de realitzar una aplicació el més modular possible. Per exemple, en futures aplicacions, en el cas que es desitgi una aplicació amb el panell de *keywords*, amb el d'ubicació i amb alguna nova que el client demani, només caldrà copiar l'arxiu en el nou projecte i realitzar les peticions corresponents. D'aquesta manera, no tindrem codi no útil o repetit en l'aplicació nova.

La primera classe que es fa servir es la de **GAMain.as**, aquesta és la que rebrà la crida des del Flash principal.

```
package as3.analyticsPanel
{
    import flash.display.LoaderInfo;
    import flash.filters.DisplacementMapFilter;
    import flash.filters.DropShadowFilter;
    import flash.filters.GlowFilter;
    import flash.geom.Matrix;
    import flash.display.MovieClip;
    import flash.system.System;
    import flash.net.URLRequest;
    import flash.net.navigateToURL;
    import flash.events.MouseEvent;

    import as3.transitions.Tint;
    import as3.extras.tests.CalendarDoubleTest;
    import as3.display.graphics.Arc;
    import as3.chart.ChartLine;
    import as3.util.Debug;
    import as3.display.Linkage;
    import as3.util.Callback;

    public class GAMain extends MovieClip
    {
        protected var _mcPrincipal:MovieClip;
        protected var _mcPaneles:MovieClip;
        protected var _mcLoading:MovieClip;
        protected var _btnPaneles:MovieClip;
        protected var _btnRango:MovieClip;

        protected var _htmlParams:Object;
        protected var _folder:String;
        protected var _php:String;
    }
}
```

El primer que es fa al iniciar una classe nova, és inicialitzar el paquet en el que estarà inclosa la classe, així com importar els altres arxius i classes que s'utilitzaran. De vegades no s'utilitzen totes les classes que s'inclouen en el directori del projecte, per tant és interessant incloure-les una a una, però en altres casos sí que es fan servir totes, per tant no és necessari importar una a una, llavors es poden importar totes alhora: `import as3.*;`

Tot seguit es realitza la inicialització de la classe. El nom de la classe serà el mateix que el del arxIU AS. A continuació declararem les variables seguit del tipus de variable que es tracta. Per a portar un cert control i organització de les variables que declarem, hem establert la 'norma' que les variables generals les declararem amb un `_` davant per tal de reconèixer-les fàcilment.

Un cop declarades les variables, és necessari assignar una variable a cadascun dels paràmetres que el PHP fa arribar al Flash.

```
_htmlParams = LoaderInfo(this.root.loaderInfo).parameters;
_php = _htmlParams["php"];
_login = (_htmlParams["login"]!=null) ? _htmlParams["login"] : "";
_password = (_htmlParams["password"]!=null) ? _htmlParams["password"] : "";
_id = (_htmlParams["id"]!=null) ? "ga:" + _htmlParams["id"] : "";
_key = _htmlParams["key"];
Debug.enabled = (_htmlParams["debug"] == "1") ? true : false;
```

Observem que en aquest tipus d'assignació, donem peu a que si una variable no la rep el Flash, per defecte l'hi assignem un valor.

Sempre que ens trobem en el codi amb una variable o anotació que porti el nom o valor: **DEBUG**, es tracta de variables de control, per a observar que funciona correctament l'execució del codi durant el procés de desenvolupament de l'aplicació. Aquests missatges només es veuran quan s'executi l'aplicació des del Macromedia Flash, un cop publicat i llest per incloure l'arxiu SWF al PHP, en cap moment l'usuari podrà veure aquests comentaris de control d'execució.

Després d'anar assignant valor a diferents variables, arriba el moment de començar a passar els paràmetres a la classe **GAProxy.as**.

```
GAProxy.instance._url = _php;
GAProxy.instance._login = _login;
GAProxy.instance._password = _password;
GAProxy.instance._id = _id;
GAProxy.instance._dateStart = fechaAnt;
GAProxy.instance._dateEnd = fecha;

Debug.msg("Conectando...");
GAProxy.instance.getGeneral(loginControl, loginError);
```

Un cop assignat el valor a una sèrie de variables necessàries en la classe **GAProxy**, procedirem a la crida del mètode **getGeneral()**. Aquesta crida porta la referència de dos mètodes o funcions de la classe **GAMain**, que s'encarregaran de configurar l'aplicació, en el cas que tot el sistema de connexions i muntatge del XML amb les dades sigui correcte (**loginControl**); o en cas contrari, que s'hagi produït algun tipus d'error, es cridarà a una funció en *Javascript*, que estarà inclosa en els PHPs per tal de mostrar una alerta amb la informació del error obtingut (**loginError**).

```
private function loginControl(xm:XML):void
{
    var error:String = xm.@error;
    Debug.msg("error:" + error);
    if (error != "" && error != null) loginError();
    else {
        GAProxy.instance._auth = xm.@auth;
        GAProxy.instance._xmlGeneral = xm;
        config(xm);
    }
}

private function loginError():void
{
    Debug.msg("Error login");

    var url:String = "javascript:login_error()";
    var request:URLRequest = new URLRequest(url);
    try {
        navigateToURL(request, "_self");
    } catch (e:Error) {
        trace("Error llamada javascript");
    }
}
```

Com es pot veure, primer de tot s'han reunit totes les dades de connexió (*login*, *password*, *id*, etc) en una classe, el **GAMain**. Tot seguit, en la classe **GAProxy** s'han recopilat més dades de connexió necessàries com les dates i des de aquesta classe, com ja veurem més tard, es portarà el control de connexió.

Llavors, des del mètode **getGeneral** es realitzarà la crida a la funció: **loadVars()**. Una funció força important, ja que d'aquí en endavant el procediment que es realitzarà serà sempre el mateix, exceptuant el valor dels paràmetres que es passarà, així com es configurarà un XML diferent segons l'apartat que desitgem mostrar de l'aplicació.



```

var loadVars:GALoadVars = new GALoadVars(dimension, metric);
loadVars._url = _url;
loadVars._id = _id;
loadVars._login = _login;
loadVars._password = _password;
loadVars._dateStart = dStart;
loadVars._dateEnd = dEnd;
loadVars._maxResults = _maxResults;
loadVars._sortDimension = sortDimension;
loadVars._auth = _auth;
loadVars.onLoadOk = function(xm:XML) {
    if(calendario!=null) calendario.calActivo = true;
    loadOk(xm);
};
loadVars.onLoadError = function() {
    if(calendario!=null) calendario.calActivo = true;
    loadError();
};
loadVars.init();

```

Des d'aquesta funció, crearem un nou objecte de la classe **GaLoadVars**, assignarem les variables necessàries, que en processos anteriors hem recollit, i s'inicialitzarà l'execució de la funció **init()** de l'objecte **loadVars** que serà l'encarregada de realitzar totes les connexions amb el PHP i aquest retornarà la informació.

```

public function init():void
{
    var request:URLRequest = new URLRequest(_url);
    var loader:URLLoader = new URLLoader();
    var vars:URLVariables = new URLVariables();

    loader.dataFormat = URLLoaderDataFormat.TEXT;
    loader.addEventListener(Event.COMPLETE, loadVarsOk);
    loader.addEventListener(IOErrorEvent.IO_ERROR, loadVarsError);
}

```

És necessari crear un nou objecte de cada una de les tres variables que hem creat. Cada una tindrà una funció diferent però ahora, aniran totalment lligades unes amb les altres. La variable **request**, realitzarà l'acció d'agafar tota la informació amb una sola petició HTTP. Els objectes que recull **URLRequest** es transmeten als mètodes **load()** de la classe **URLLoader**.

La classe **URLLoader** serà l'encarregada de descarregar les dades i la informació de la URL com a dades, text o variables amb una codificació concreta. En el nostre cas serà útil per a descarregar l'arxiu XML. Cal recordar que el PHP s'ha encarregat de muntar amb tota la informació necessària per a mostrar de forma correcta l'informació.

La classe **URLVariables** permet transferir variables entre una aplicació i un servidor, emmagatzemant així dades e informació necessària.

Un cop tot aquest procés s'ha efectuat, s'indica a la variable **loader**, que si s'ha realitzat correctament l'event, es realitza la crida a la funció **loadVarsOk**; però per contra, si ha sorgit algun tipus d'error, es realitza la crida a la funció **loadVarsError**.

Si tot ha funcionat correctament, es cridarà a la funció **loadVars** de la classe **GAProxy**. Aquesta s'encarregarà de fer arribar el XML amb les dades a la funció **loginControl**. Per a no perdre la referència, s'ha anat passant com a paràmetres cada cop que hem fet una crida a una funció d'un objecte diferent. Aquesta, rebrà el XML i si tot a anat correctament, cridarà a la funció de **config()** situada en **GAMain**, per a procedir a la configuració de la pàgina que l'usuari vol veure per pantalla i mostrar la informació de forma adequada.

Observem com ha anat aquest procediment i que en tot moment s'ha conservat la referència a una funció situada en el mètode inicial de la classe **GAMain**.

- Primera crida des de la classe **GAMain**:

```
GAProxy.instance.getGeneral(loginControl, loginError);
```

- Crida des de la funció **getGeneral** de la classe **GAProxy**:

```
loadVars(dimension, metric, loadOk, loadError);
```

Observem que tot i canviar el nom de **loginControl** a **loadOk**, la referència segueix sent la mateixa.

En el primer cop que la aplicació executa el codi, automàticament es carregarà el panell número 0.

```
cargarPanel(0);
```

Cal tenir en compte que cada opció del menú de l'aplicació correspondrà a un panell diferent, per tant tindrem tants panells, com opcions de menú que el client final desitgi per a l'aplicació. D'aquesta manera, serà molt senzill incloure noves funcionalitats a l'aplicació o treure'n d'altres que no es vulguin en futures aplicacions per a clients diferents.

També cal remarcar el fet que cada panell, disposarà d'una classe pròpia. Per tant, també tindrem tants arxius AS corresponents a tantes opcions de menú com hi hagin.

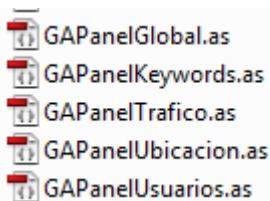
Observem com cada **if** correspon a un índex concret que crearà un panell o altre.

```

if (index == 0) {
    _panelActual = new GAPanelGlobal(GAProxy.instance._xmlGeneral);
    _mcPaneles.addChild(_panelActual);
    _panelActual.init();
}
else if (index == 1) {
    _panelActual = new GAPanelUsuarios(GAProxy.instance._xmlGeneral);
    _mcPaneles.addChild(_panelActual);
    _panelActual.init();
}
else if (index == 2) {
    if (_panelUbicacion == null) {
        _panelUbicacion = new GAPanelUbicacion(_key, GAProxy.instance._xmlUbicacion);
        _mcPaneles.addChild(_panelUbicacion);
        _panelUbicacion.init();
    }
    else {
        _mcPaneles.addChild(_panelUbicacion);
        _panelUbicacion.update();
    }
    _panelActual = _panelUbicacion;
}
else if (index == 3) {
    _panelActual = new GAPanelTrafico(GAProxy.instance._xmlTrafico);
    _mcPaneles.addChild(_panelActual);
    _panelActual.init();
}
else if (index == 4) {
    _panelActual = new GAPanelKeywords(GAProxy.instance._xmlKeywords);
    _mcPaneles.addChild(_panelActual);
    _panelActual.init();
}
}

```

Aquests són els arxius AS corresponents a cada panell diferent de l'aplicació.



- GAPanelGlobal.as
- GAPanelKeywords.as
- GAPanelTrafico.as
- GAPanelUbicacion.as
- GAPanelUsuarios.as

El fet d'haver decidit programar-ho d'aquesta forma, ens planteja l'opció de modificar l'aplicació i els seus components de menú de forma molt senzilla. Únicament serà necessari afegir una altra opció amb un índex assignat de forma correlativa i crear el nou arxIU AS amb la classe corresponent.

Per continuar explicant el funcionament de l'aplicació cal tenir en compte que cada panell de l'aplicació té una part comú i una altre específica.

La part comú en tots els panells és que totes s'inicien amb la funció **init()**. En aquesta, es realitzen càlculs específics, segons el panell en el que estem però acte seguit, sempre es crida a la funció **config()**.

A continuació s'explicarà les parts més importants del codi de cada panell.

### **PANEL·L GLOBAL (GAPanelGlobal.as):**

En el cas del panell global (el que mostra l'aplicació per defecte a l'iniciar l'execució d'aquest), observem com es realitza la crida a diferents funcions especials per a cada tipus de dades que es vol mostrar. Cal recordar que aquest panell inicial, és un breu resum de les principals dades del lloc web.

```
configNormal(0, GAProxy.instance._metricVisitas, "suma", "visitas"); // Visitas
configRebote(1); // Porcentaje Rebote
configNormal(2, GAProxy.instance._metricPaginasVistas, "suma", "paginas"); // Páginas vistas
configTiempoSitio(3); // Promedio de tiempo en el sitio
configPaginasVisita(4); // Páginas/visita
configVisitasNuevas(5); // Porcentaje de visitas nuevas
```

En cadascuna d'aquestes funcions es realitzen els càlculs necessaris per a mostrar el gràfic corresponent amb la seva dada informativa. Un cop s'han realitzat els càlculs, en tots els mètodes es crida a la mateixa funció que s'encarregarà de muntar el requadre del panell principal amb el gràfic en petit i la informació al costat.

```
initChart(i, datos, total, txt);
```

Dins d'aquest mètode, també es realitza una crida al mètode **graficaMax**. Observem com aquest mètode només s'executarà en el cas que l'usuari llenci un event de fer clic amb el *mouse* en el botó corresponent.

```
mv.btn.addEventListener(MouseEvent.CLICK, function() { graficaMax(i, datos, modoSegundos, txt); });
```

En aquest mètode és on es crearà el gràfic a mida ampliada. Al tractar-se del panell principal de l'aplicació, també es programarà un botó de tornar enrere per tal de poder tornar a la informació anterior, sense tenir que tornar mitjançant el menú.

**PANEL·L USUARIS (GAPanelUsuarios.as):**

Quan l'usuari tria aquesta opció en el menú de l'aplicació, recordem que es mostra un gràfic a pantalla completa, amb la informació de les visites durant el període de temps seleccionat per l'usuari.

En aquest cas, la funció **config()** directament realitza la crida a la funció **graficaMax()**, ja que en aquesta pantalla únicament es mostra el gràfic. Podem veure com, abans de realitzar aquesta crida i per tant crear el gràfic, es comprova si ja existeix per a esborrar-lo abans. Això serveix per quan l'usuari està veient un gràfic i sense canviar de pantalla, decideix modificar les dates per veure informació d'un altre període de temps.

```
protected function config(reinit:Boolean = false):void
{
    if(reinit && chartMax!=null) removeChild(chartMax);

    graficaMax();
}
```

En la funció **graficaMax()**, es farà el recull de dades.

```
var datos:Array = getData(GAProxy.instance._metricUsuarios);
var data:Array = new Array();

for (var j:int = 0; j < datos.length; j++) {
    data.push( {x:datos[j].key, y:datos[j].value } );
}
```

Per a crear un gràfic lineal, necessitem dos eixos de dades pel gràfic. Observem com primer es recuperen les dades de la *array* '**datos**'. Tot seguit, es declara una nova *array* amb el nom de **data**. En aquesta segona *array*, anirem separant en cada posició d'aquesta el valor per l'eix 'x' i el valor per l'eix 'y'.

Tot seguit es crearà l'objecte de la classe **ChartLine** que és la que s'encarrega de crear el gràfic.

### **PANEL·L UBICACIÓ (GAPanelUbicacion.as):**

Fins el moment, en tots els panells explicats, la funció **init()** únicament s'encarregava de cridar a la funció **config()** per a començar a configurar el nou panell, però en aquest cas, ens trobem amb un panell una mica diferent. Al tractar-se d'un mapa de Google Maps, abans de començar a configurar les dades per a mostrar-les, hem de realitzar un seguit de crides i comprovacions per tal de deixar el mapa de Google Maps al nostre gust i disseny.

```
public function init():void
{
    _map = new Map();
    _map.key = _key;
    _map.setSize(new Point(_w, _h));
    _map.addEventListener(MapEvent.MAP_READY, onMapReady);
    _map.addEventListener(MapZoomEvent.ZOOM_CHANGED, onZoomChange);
    addChild(_map);

    _mcInfo = Linkage.movie("info_window");
    _mcInfo.visible = false;
    addChild(_mcInfo);
}
```

Podem observar que una de les primeres accions que es fa, és recuperar la variable **\_key**. Aquesta variable, com ja hem explicat en apartats anteriors es tracta de la *key* de la Api del mapa de Google. És imprescindible que per cada projecte que estigui allotjat en un domini diferent, realitzar el procés d'aconseguir una nova. Google controla per dominis el tema de *apikey*, per tant només cal assignar una *apikey* al domini sobre el que treballarem.

Observem que s'assigna un event a la variable **\_map**. Quan estigui creat correctament i configurat com a tal, es procedirà a executar l'event **onMapReady**.

```

private function onMapReady(event:Event):void
{
    var zoomInicial:Number = 4;

    _map.setCenter(new LatLng(40.736072, -73.992062), 14, MapType.NORMAL_MAP_TYPE);

    _controlZoom = new GAZoomControl(zoomInicial);
    _map.addControl(_controlZoom);
    _map.setZoom(zoomInicial);
    _map.enableScrollWheelZoom();
    _map.enableContinuousZoom();

    if (_xml == null) loadData();
    else config();
}

```

Quan s'ha cridat aquest mètode, és perquè el mapa ja està preparat per a mostrar-se, únicament cal indicar-li alguns aspectes de la configuració. Per exemple, la posició per defecte que es mostrarà (**\_map.setCenter**), crear un nou objecte del control de zoom, etc. Per defecte el mapa de Google ja porta un disseny, però en segons quins casos és important redissenyar-lo per tal d'aconseguir un aspecte més amigable. Per tant, es crea un nou objecte de la classe **GAZoomControl**, i tot seguit, es realitzen tot un conjunt de configuracions d'aquest.

Finalment, es comprova primer que el XML (amb nom **\_xml**) no estigui buit (per si ha sorgit qualsevol error imprevist alhora de configurar el mapa) i en cas que el XML estigui correcte, es procedeix a la crida de la funció **config()**, per a començar a realitzar el seguit d'accions per a mostrar les dades en el mapa que ja hem programat.

En la funció **config()**, com en la resta de casos, es realitza les operacions necessàries per extreure del XML les dades i organitzar-les per ubicar-les en pantalla. En aquest cas, del XML hem d'extreure una gran quantitat de dades per a cada punt d'informació. Per a cada lloc a ubicar en el mapa, hem de extreure quatre variables: ubicació, latitud, longitud i el nombre de visites. En cas que la ubicació no sigui nul·la (que retorni un valor), ja la podem situar en el mapa; per a realitzar aquesta acció es crida a la funció **doGeocode()**, passant-li com a paràmetres les quatre variables que hem obtingut.

```

for (i = 0; i < _xml.n1.length(); i++) {
    ubicacion = _xml.n1[i].key[0];
    latitud = _xml.n1[i].n2[0].key[0];
    longitud = _xml.n1[i].n2[0].n3[0].key[0];
    visitas = _xml.n1[i].n2[0].n3[0].n4[0].valor[0];
    if(ubicacion!=null) doGeocode(ubicacion,latitud,longitud,visitas,max);
}

```

La funció **doGeocode()** és l'encarregada de col·locar el punt en el mapa i crear la capa de més informació, que mostrarà el nombre de visites quan l'usuari col·loqui el cursor sobre el punt concret.

### **PANEL·L TRÀFIC (GAPanelTrafico.as):**

Aquest panell, és el que ens mostra uns gràfic sobre les fonts de tràfic de la web. En tot moment, podem ordenar els resultats per total de visites, pàgines vistes o temps en el lloc web, mitjançant una llegenda que prèviament s'ha programat.

En aquest cas, la funció **config()** s'encarrega de configurar cadascuna de les opcions de dades a mostrar en el gràfic, (**configBtn(btn)**), crear la llegenda i actualitzar les dades informatives segons l'opció en que ens trobem (**configOpcionLeyenda(leyenda)**) i mostrar per defecte el gràfic de la primera opció (**chartChange(\_opcionActual)**).

La funció **configBtn** s'encarrega de cridar en cada moment al gràfic que l'usuari selecciona.

```
protected function configBtn(btn:MovieClip):void
{
    btn.buttonMode = true;
    btn.mouseChildren = false;
    btn.addEventListener(MouseEvent.CLICK, function() { chartChange(btn); });
}
```

La funció **configOpcionLeyenda()** s'encarrega de mostrar o amagar la capa de llegenda segons l'usuari col·loqui el cursor sobre una zona del gràfic o una altre.

```
protected function configOpcionLeyenda(btn:MovieClip):void
{
    btn.buttonMode = true;
    btn.mouseChildren = false;
    btn.addEventListener(MouseEvent.CLICK, function() { _chart.over(btn.index); });
    btn.addEventListener(MouseEvent.CLICK, function() { _chart.out(btn.index); });
}
```

Finalment, la funció **charChange** és l'encarregada de construir, configurar i mostrar el gràfic.

El que primer es fa, és construir una *array* amb el valor de totes les dades que es faran servir per a muntar el gràfic. Per a fer-ho cridem a la funció **searchMetric()**.



```
var cantidades:Array = searchMetric(metric);
```

Aquesta funció s'encarrega de buscar en el XML les dades que es necessiten.

En el cas del total de visites i en el de pàgines visitades, la construcció del gràfic és directe recollint les dades i organitzant-les per a mostrar-les en el gràfic, però en el cas del temps en el lloc web és diferent, s'ha de tractar abans les dades per tal de mostrar-les correctament.

```
if (modoSegundos) { // Tiempo Sitio
    var visitas:Array = searchMetric(GAProxy.instance._metricVisitas);
    var n1, n2, n:Number;
    for (var i:int = 0; i < cantidades.length; i++) {
        n1 = visitas[i];
        n2 = cantidades[i];
        n = Math2.roundTo((n2 / n1), 2);
        cantidades[i] = n;
    }
}
```

Aquesta comprovació serveix per saber si ja existeix un gràfic, llavors només és necessari modificar les dades, mitjançant la funció **cambiarCantidades()**; però en el cas del primer cop que s'accedeix és necessari crear-lo. Per a crear-lo, crearem un nou objecte de la classe **CharPie()**. Com a paràmetres li farem arribar les dades que s'han extret prèviament per a mostrar.

```
if (_chart == null) {
    _chart = new ChartPie(cantidades);
    addChild(_chart);
    _chart.onOver = chartOver;
    _chart.onOut = chartOut;
    _chart.colores = new Array(0xEE891F, 0x6CA024, 0x1887CC);
    _chart.x = 696;
    _chart.y = 181;
    _chart.init();
}
else {
    _chart.cambiarCantidades(cantidades);
}
```

Per últim, en aquest mètode es fan tot un seguit de crides i assignacions per tal de configurar correctament la llegenda del gràfic. Segons la opció que l'usuari hagi triat, la variable **index** tindrà un valor o un altre, i per tant, es podrà assignar a les diferents variables el valor correcte.

```
var t:String;
if (index == 0) t = "visitas";
else if (index == 1) t = "paginas";
else if (index == 2) t = "";

// cambiar textos
this.referencia.text = "("+c0+" "+t+")";
this.directo.text = "("+c1+" "+t+")";
this.búsqueda.text = "(" + c2 + " "+t+")";
```

El valor **index** ve determinat per una assignació prèvia que s'ha fet al iniciar el mètode de **charChange()**.

```
var index:Number = btn.index;
```

Aquest **btn.index** correspon a un valor que se l'hi ha assignat alhora de crear el botó per tal de diferenciar-lo de la resta i així després poder tractar-los diferent.

### **PANELL PARAULES CLAU (GAPanelKeywords.as):**

Aquest panell mostra una taula amb tot un conjunt de paraules clau amb les quals els usuaris han accedit a la pàgina web. En cada columna de la taula, tenim tot un conjunt de dades interessants a conèixer de cada paraula clau, com pot ser el nombre de visites que ha aportat aquesta paraula, el nombre mig de pàgines vistes per visites, el temps mig en el lloc web, percentatge de visites noves i el percentatge de rebot.

En aquest cas, la funció **config()** s'encarrega d'obtenir a mesura que es va creant la taula, els valors necessaris per a cada columna.

En aquest cas, ens trobem amb un tipus d'assignació diferent a la trobada fins al moment. Es tracta d'una assignació pròpia del ActionScript 3 on es realitza una cerca d'igualtat en la mateixa assignació.

```
n1 = _xml.n1[i].n2.(key[0] == GAProxy.instance._metricVisitas).valor[0];
n2 = _xml.n1[i].n2.(key[0] == GAProxy.instance._metricTiempoSitio).valor[0];
```

Un cop ja tenim totes les files preparades, s'inicia la construcció de la taula.

```
_mcTabla.sb.init(_mcFilas, "linearNone",2,true,1.5);
```

## 7. Google Analytics Mobile

### 7.1 Navegant per l'aplicació

#### 7.1.1 Introducció

A diferència del panell de Google Analytics en Flash, on l'usuari tenia un gestor al qual accedia amb un usuari i contrasenya; tot seguit tenia l'opció d'introduir el Id del seu compte de Google Analytics, i finalment, accedir al panell on es mostraven les dades mitjançant l'usuari i contrasenya del seu compte de Google que esta vinculat al compte, en l'aplicació per a dispositiu mòbil, l'usuari accedirà mitjançant la URL que es decideixi amb el client. Tot i així, per defecte serà: [www.domini-lloc-web.com/m/](http://www.domini-lloc-web.com/m/); i únicament necessitarà el seu usuari i contrasenya del seu compte vinculat al compte de Google Analytics.

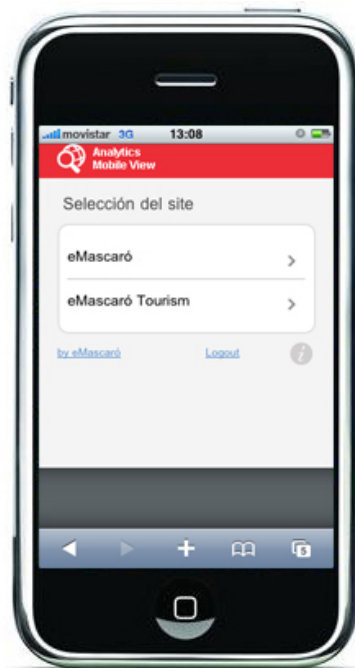


Figura 26: Google Analytics Mobile

En aquest cas, no serà necessari que l'usuari introdueixi el Id del seu compte de Google Analytics, ja que aquest anirà preprogramat en l'aplicació web. S'ha decidit fer-ho d'aquesta manera, ja que al tractar-se d'una aplicació web totalment independent és

necessari tenir un control de qui i com ho fa servir. El client perfectament podria canviar l'aplicació de servidor i començar a comercialitzar-la sense el nostre permís. Per tant, el Id del compte de Google Analytics anirà inclòs en la programació, de forma que al client que se l'hi hagi venut l'aplicació, només podrà fer servir l'aplicació amb el compte de Google Analytics que prèviament s'ha decidit.

A diferència de l'aplicació en Flash de Google Analytics, en aquest cas, es mostrarà un llistat de tots els llocs web amb Analytics vinculats al compte de correu, per tant es podrà veure les estadístiques de tots els llocs web que un compte de mail tingui vinculat.



Un fet important al tractar-se d'una aplicació per a dispositiu mòbil, era el temps de càrrega. Sovint les connexions per a dispositius mòbils són lentes si no estan connectades a una xarxa Wifi. Per tant, era molt important tenir ben controlat aquest aspecte.

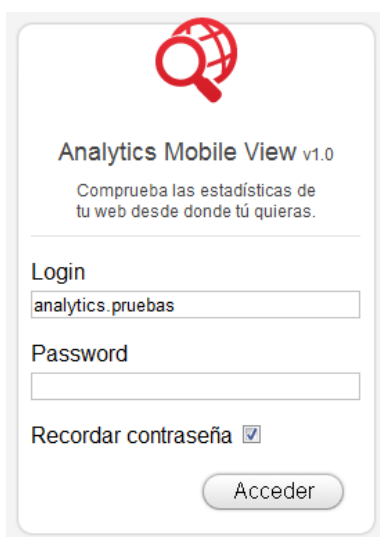
Al tractar-se d'una aplicació que ha de llegir la informació de la Api de Google, constantment era necessari realitzar consultes que suposaven un temps de càrrega considerable. Per tal d'aconseguir un temps de càrrega correcte, s'ha decidit que la millor solució serà les dades en un arxiu XML i en posterior consultes, directament llegir del XML.

Quan el primer usuari del dia accedeixi a l'aplicació, aquesta connectarà amb Google i emmagatzemarà totes les dades en un XML. Els següents usuaris que accedeixin a

l'aplicació, podran visualitzar les dades més ràpidament, ja que obtenir les dades del XML és molt més ràpid que realitzar la connexió amb Google cada cop.

### 7.1.2 Panell d'accés

L'usuari introduirà el seu *login* i *password*. En aquesta primera pantalla, pot seleccionar la casella de “*recordar contrasenya*”. D'aquesta forma, en propers cops que intenti accedir, no serà necessari introduir les dades d'identificació de nou.



Analytics Mobile View v1.0  
Comprueba las estadísticas de tu web desde donde tú quieras.

Login  
analytics.pruebas

Password

Recordar contraseña

Acceder

Figura 27: Accés a Google Analytics Mobile

### 7.1.3 Llistat de llocs web

Un cop l'usuari s'ha loguejat correctament, es mostra el llistat de llocs webs vinculats al compte amb el qual s'ha accedit. Des d'aquest llistat es triarà el lloc web del qual es vol visualitzar la informació.

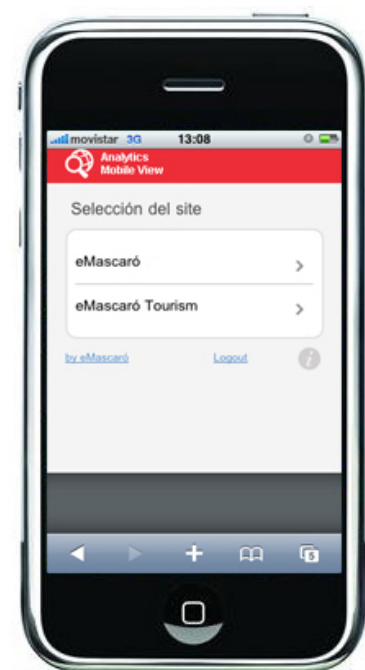


Figura 28: Llistat de llocs web Google Analytics Mobile

### 7.1.4 Panell d'estadístiques generals

Per defecte, es mostra un panell resum de les dades més importants (segons el client ha especificat prèviament).



Figura 29: Panell d'estadístiques generals Google Analytics Mobile

En aquest cas, es mostra la informació de les visites al lloc web, el nombre de pàgines vistes, el nombre mig de pàgines vistes per cada visita a la web, el temps mig en el lloc web i el tant per cent de visites que és el primer cop que accedeixen a la web.

Período: 10-01-2010	
Visitas	34
Páginas Vistas	104
Páginas / Visita	3.06
Tiempo en el Site	0:02:45
Visitas Nuevas	85%

- **Visites:** Les visites són el nombre d'usuaris totals que accedeixen a la pàgina web. Aquestes visites poden vindre de buscadors, de altres llocs webs o simplement perquè l'usuari ha escrit la direcció web en el seu navegador.

- **Pàgines vistes:** Les pàgines vistes és el nombre total de pàgines que els usuaris que han entrat en el lloc web i han estat navegant per ell. Aquesta dada ens pot indicar si els usuaris estan interessats o no en el contingut en general que ofereix la pàgina web.
- **Pàgines/visita:** Aquesta dada mostra la mitja de pàgines vistes per a cada usuari que accedeix al lloc web. Cal entendre, que contra més alta sigui aquesta dada, significarà que l'usuari realment està interessat en el contingut de les pàgines del nostre lloc web, i per tant, es converteix en un lloc web d'interès.
- **Temps en el lloc:** Aquest paràmetre indica el temps mig que els usuaris que visiten la pàgina web estan en el *site*. Es tracta d'un altre paràmetre que, contra major sigui, ens indicarà que l'usuari està més interessat en el contingut de la pàgina web.
- **% de visites noves:** El percentatge de visites noves és un paràmetre que indica el percentatge de nombre de visites que és el primer cop que accedeixen a la web. D'aquesta forma es pot analitzar si l'usuari que accedeix a la web és un visitant recurrent o si per contra, es tracta d'un usuari nou.

També es mostra en període de temps del qual es mostra la informació. Aquest període de temps es pot modificar des dels botons situats a la part superior de les dades. Al tractar-se d'una aplicació preparada per a telèfons mòbils, s'ha de fer d'una forma que sigui usable i compatible per a tots els dispositius. La millor forma era la de preparar uns botons amb uns períodes de temps predeterminats.



L'usuari podrà triar entre veure les dades del dia de ahir, dels últims 7 dies, de l'últim mes o d'un any abans, partint des de la data actual.

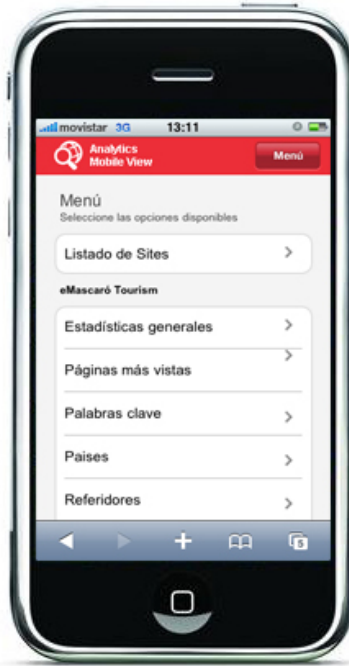
En la part superior de l'aplicació observem el botó "Menú" que permetrà accedir a la resta d'opcions de l'aplicació.





### 7.1.5 Panell de Menú

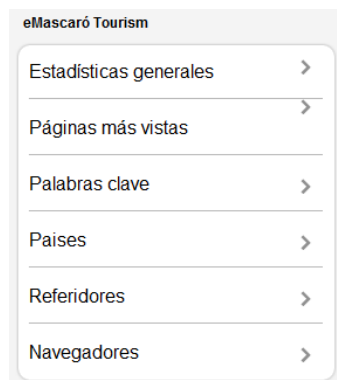
Per tal de poder navegar per tota l'aplicació sense haver de fer ús dels botons del navegador, podrem accedir a l'opció de "Menú" i accedir a totes les opcions.



*Figura 30: Panell Menú de Google Analytics Mobile*

La primera opció d'aquest menú, és la de tornar al llistat de llocs web vinculats al compte. Es tracta del llistat de llocs web que es pot observar un cop l'usuari s'ha loguejat a l'aplicació. D'aquesta forma, es dona l'opció de passar d'un lloc web a un altre, sense necessitat de tornar-se a loguejar.

A continuació es mostra el nom del lloc web en el qual estem actualment loguejats i el llistat d'opcions a las que podem accedir.



### 7.1.6 Panell de pàgines més vistes

En aquesta opció, es mostrarà un llistat amb les 10 pàgines que han obtingut més visites en el període de temps triat. Al tractar-se d'una aplicació per a dispositiu mòbil, s'ha limitat a només 10 valors per estalviar temps de càrrega. Es considera que mostrant 10 resultats ja es una xifra prou significativa per valorar i prendre decisions. Tot i així, si el client decidís que vol visualitzar més resultats, es podria ampliar el nombre de resultats sense gaire feina afegida.



Figura 31: Panell de pàgines més vistes Google Analytics Mobile

### 7.1.7 Panell de paraules clau

Aquesta altre opció, mostra el llistat de les 10 paraules clau més utilitzades. Les paraules clau són aquelles amb les que els usuaris accedeixen a la pàgina web a través dels cercadors.



Figura 32: Panell de paraules clau Google Analytics Mobile

És important aconseguir un important nombre de paraules clau. Si les paraules clau no contenen el nom de l'empresa o el domini de la pàgina web, significarà que la gent que accedeix a la web realment és un tipus de gent que l'interessa el contingut que s'ofereix. Per tant, també es podrà deduir que la pàgina web està força ben posicionada.

Les paraules clau normalment formen part del contingut de text de la mateixa pàgina web, per tant, és important tenir actualitzat el contingut de la web i que aquest realment correspongui amb el que s'ofereix en la web.

### 7.1.8 Panell de països

Des d'aquesta opció, l'usuari podrà observar el llistat de països des dels quals s'han obtingut visites. Aquesta dada és força important per a portar un control dels diferents idiomes de la web, o per si es necessari traduir la web a un idioma en concret.



Figura 33: Panell de països de Google Analytics Mobile

Estudiant bé la procedència de les visites, pot donar peu a realitzar accions concretes per afavorir el tràfic de visites des d'aquests països. Una acció molt positiva pot ser, per exemple, crear ofertes o campanyes específiques per atreure més públic.

### 7.1.9 Panell de referidors

Els referidors, en aquest cas, es tracta de la forma amb la qual els usuaris han accedit a la pàgina web. Ja pot ser des d'un cercador, des d'una altra pàgina web o escrivint directament la URL en el navegador.

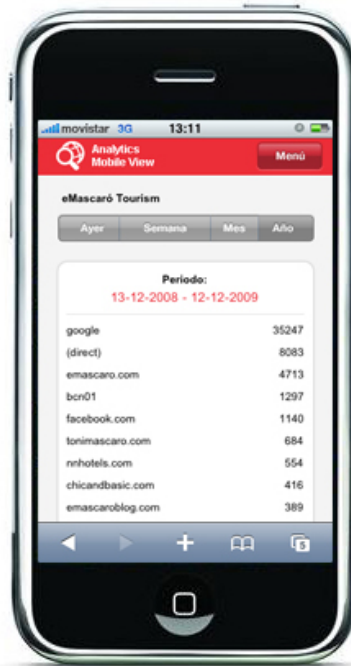


Figura 34: Panell de referidors de Google Analytics Mobile

És important conèixer la procedència de les visites a la web, ja que així es pot analitzar el nivell d'importància de la web.

### 7.1.10 Panell de navegadors

És necessari conèixer des de quins navegadors accedeixen els usuaris a la web. Sovint una mateixa pàgina web no es visualitza de la mateixa forma en un navegador o altre. Per tant, si es dona el cas que s'obtenen moltes visites d'un navegador concret, serà de vital importància optimitzar al màxim la pàgina per aquell navegador web.

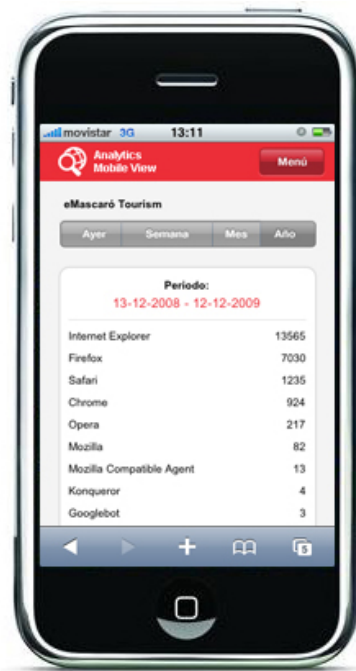


Figura 35: Panell de navegadors de Google Analytics Mobile

Actualment el navegador més utilitzat segueix sent el *Internet Explorer*, però poc a poc el *Firefox* i el *Chrome* tenen més importància. Per tant, és molt important que una pàgina web sigui 100% compatible amb *Internet Explorer* (i les diferents versions d'aquest), *Firefox* i *Chrome*.

### 7.1.11 Logout

Finalment la última opció és la de tancar la sessió.

## 7.2 Codi Font

### 7.2.1 Introducció

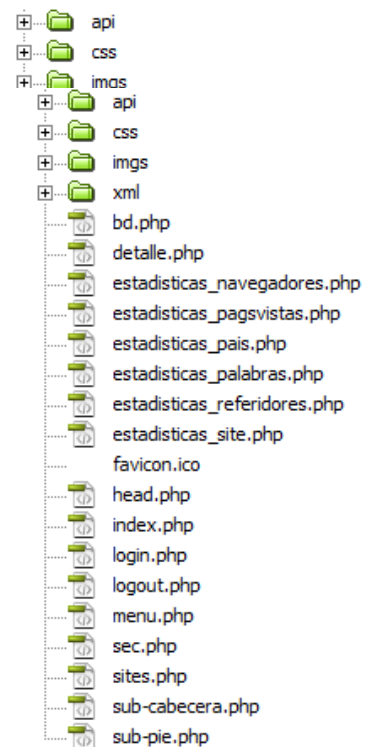
A continuació s'explicarà pas a pas el codi font més important i rellevant de l'aplicació per a telèfon mòbil de Google Analytics. Per a explicar-ho d'una forma correcta i ordenada, s'explicarà de forma en que es va executant a l'aplicació.

Cal recordar que per la programació d'aquesta s'ha fet servir els següents llenguatges i metodologies:

- PHP
- HTML
- CSS
- XML

Per tenir organitzat de forma correcte els arxius que formen part de l'aplicació, s'ha organitzat per carpetes els arxius.

- **Arrel:** a l'arrel tindrem els arxius PHP, que corresponen alguna de les pàgines de l'aplicació.
- **Carpeta api:** en aquesta hi figuraran els arxius necessaris per a fer la connexió amb la api de Google Analytics.
- **Carpeta img:** totes les imatges del projectes les tindrem agrupades en aquesta ubicació.
- **Carpeta css:** en aquesta carpeta estaran guardats els diferents arxius dels estils de la pàgina.
- **Carpeta xml:** en aquesta carpeta s'aniran creant i emmagatzemant els arxius XML que es creïn amb les dades de l'aplicació.



L'arxiu **analytics\_api.php** és el que conté totes les funcions necessàries per a realitzar les connexions amb la Api de Google Analytics.

A mesura que s'expliquin les diferents pàgines i parts de l'aplicació, es farà referència a diverses funcions d'aquest arxiu.

Un altre arxiu important i necessari per el bon funcionament de l'aplicació és el **bd.php**. Aquest conté un gran nombre de funcions útils que en nombroses ocasions es cridaran.

Per exemple, la funció: **dom\_to\_simple\_array**, realitza un conjunt de crides i accions per transformar el contingut d'un XML en una *array*. D'aquesta manera, podrem treballar i tractar les dades del XML en forma de variables en una *array* de forma molt més senzilla. Per una altre banda, la funció **simple\_array\_to\_dom** realitza l'acció contrària, crea un XML a través d'una array de dades.

```
function dom_to_simple_array($contenido)
{
    $aData=array();
    // Quitamos el carácter de salto de línea que impide decodificar correctamente el XML
    $contenido=str_replace(chr(0x0D),"",$contenido);
    // Busca el contenido entre los dos tag
    $pos=strpos($contenido, "\r\n");
    $primera=$contenido;
    if (strpos($primera,"encoding=\"UTF-8\"")===false) $utf=false;
    else $utf=true;
    $contenido=substr($contenido, $pos+2);
    $contenido=elimina_comments($contenido);
    procesa($aData, $contenido, $utf);
    return $aData;
}

function simple_array_to_dom ($aData, $utf = true, $xsl="")
{
    if ($utf) $enc="UTF-8";
    else $enc="ISO-8859-1";
    $xml="<?xml version=\"1.0\" encoding=\"\$enc\" ?>\r\n".$xsl."\r\n";
    $xml=transforma($aData, $xml, 0, $utf);
    return $xml;
}
```



## 7.2.2 Panell d'accés

L'aplicació s'inicia des de l'arxiu **index.php**. en aquest ens trobarem amb un formulari amb el camp de introducció de *login* i *password*.



El primer que es fa és comprovar si l'usuari anteriorment ja ha accedit a l'aplicació i havia marcat l'opció de “Recordar contrasenya”.

```
<?php
if (@$_COOKIE["analytics"])
{
echo "<SCRIPT LANGUAGE=\"JavaScript\"> location.href='login.php'; </SCRIPT>";
}
?>
```

Aquesta comprovació, la podem fer gracies a que el primer cop que l'usuari ha accedit hem guardat una o més *cookies* en PHP, amb la informació necessària per a realitzar el procés de loguejar i recuperar les dades de Google Analytics. Observem com en el codi, si la *cookie*: **analytics** existeix, llavors automàticament indiquem al navegador que es dirigeixi a l'arxiu **login.php**.

En cas que l'usuari no hagi accedit anteriorment, l'usuari omplirà les dades en el formulari i el fer clic sobre el botó de “Acceder”, el formulari enviarà totes les dades a l'arxiu **login.php**.

```
<form name="form" method="post" action="login.php" style="margin:0px; padding:0px">
```

Observem que per indicar l'acció del formulari, només és necessari indicar-ho en el tag **action** en el moment de inicialitzar el formulari.

A continuació s'explicarà el conjunt d'accions que es realitzen en l'**arxiu login.php**.

El que primer es fa, és recuperar les dades enviades per el formulari, es recuperen amb el mètode **POST**, ja que des del formulari del panell d'accés s'havia indicat aquest mètode. Cal recordar que hi ha dues formes d'enviar les dades, per **POST** i per **GET**.

```
// es recuperen les dades enviades per el formulari
$login=@$_POST["login"];
$password=@$_POST["pass"];
$recordar=@$_POST["recordar_psw"];
```

En el cas que l'usuari hagi marcat la casella de "*Recordar contrasenya*", és necessari crear *cookies* per emmagatzemar aquestes dades. Per tant, el que farem serà crear un conjunt de *cookies* amb la informació de *login* i *password* de l'usuari.

```
if($recordar=="si")
{
// creo las cookie
setcookie ("us_analytics", $login, time () + 604800);
setcookie ("pass_analytics", $password,time () + 604800);
setcookie ("analytics", "cookies", time () + 604800);
}
```

També s'ha de tenir en compte si es dona el cas que l'usuari ha accedit mitjançant la *cookie* que tenia guardada, en aquest cas, és necessari recuperar les dades de diferent forma. Només és necessari assignar-li el valor a la nova variable **\$login** i **\$password**.

```
// si existeix la cookie, recuperem login i password
if($_COOKIE["analytics"]){
    $login=@$_COOKIE["us_analytics"];
    $password=@$_COOKIE["pass_analytics"];
}
```

Un cop ja tenim recuperat el valor del *login* i el *password*, estem en disposició de començar el procés de loguejat enviant les dades a Google.

Crearem primer, un objecte de la classe **analytics\_api()**, d'aquesta forma ja podrem fer crides a funcions ubicades dins del **analytics\_api.php**.

```
$api = new analytics_api();
if($api->login($login, $password))
```

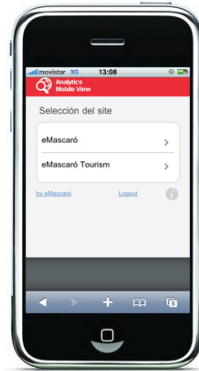
Com podem veure, dins del **if**, trobem la crida a la funció *login*. Aquesta funció s'encarrega de realitzar la connexió amb Google Analytics i retornar una variable String anomenada **auth** que posteriorment serà necessària per a recuperar les dades.

```
if($api->login($login, $password)) {
    $entrar = true;
    @session_register("login");
    @session_register("password");
    $_SESSION["login"]=$login;
    $_SESSION["password"]=$password;
    $_SESSION["auth"]=$api->auth;
    $_SESSION["autorizacion"]=true;
    header("location:sites.php");
}else{
    $_SESSION["autorizacion"]=false;
    header("Location:index.php?msg=1");
}
```

Per tant, com podem veure, si la funció *login* retorna un String llavors és correcte el procés de loguejar-se, i s'emmagatzemen un conjunt de variables a la sessió per poder-les fer servir posteriorment. Finalment, s'indica al PHP que redirigeixi a l'usuari a la pàgina **sites.php**. En el cas que el procés de loguejat no a retornat cap String, es redirigeix a l'usuari a la pàgina **index.php** mostrant-li un missatge d'error.

### 7.2.3 Llistat de llocs web

A continuació, s'accedeix a la pàgina amb el llistat de llocs web vinculats al compte amb el qual s'ha accedit.



El primer que es realitza en aquest PHP, és crear un objecte de la classe Api un altre cop, i assignar-li el valor del String d'autorització que en el procés anterior de loguejat hem obtingut. En aquest cas, el recuperem de la variable de sessió que hem creat anteriorment, així no és necessari tornar a realitzar el procés de loguejat.

```
$api = new analytics_api();  
$api->auth = $_SESSION["auth"];  
$xmlsites="xml/sites.php";
```

Cal recordar que al tractar-se d'una aplicació per a telèfon mòbil, i que la connexió pot ser més aviat lenta, és molt important realitzar totes les accions possibles per estalviar llargs temps d'espera en connexions lentes.

Tot seguit també crearem una nova variable amb la ruta del XML que crearem amb el llistat de *sites* vinculats al compte de email.

A continuació, si l'autorització és vàlida, procedirem a la crida de la funció **load\_accounts()** i tot seguit crearem una nova *array* anomenada **\$xml** per anar emmagatzemant les dades que s'extreguin.

```
if($_SESSION["autorizacion"]) {  
    $api->load_accounts();  
  
    $xml=array();
```

```
foreach($api->accounts as $account){
    $xml["root"][0]["account"][]=$account;
}

escribe_xml($xmlsites, simple_array_to_dom($xml));
```

Mitjançant aquest bucle, anirem assignant a cada posició de l'*array* un compte, i tot seguit, mitjançant la funció **escribe\_xml()** crearem el XML amb totes aquestes dades.

Observem que la funció **escribe\_xml()** s'encarrega de guardar el XML. Mitjançant la funció **simple\_array\_to\_dom()**, procedirem a la creació del XML amb les dades guardades en l'*array* que hem creat en el bucle anterior.

```
function escribe_xml($fichero, $xml)
{
    $fp=fopen($fichero,"w");
    fwrite($fp, $xml);
    fclose($fp);
}
```

És necessari crear aquest arxiu XML per emmagatzemar les dades i que en posteriors cops que l'usuari accedeixi al llistat, no sigui necessari tornar a realitzar la consulta. Un cop més estem intentant estalviar temps de càrrega.

Un cop ja s'ha creat el XML i tenim totes les dades per a mostrar per pantalla, es comença el procediment de presentar la informació.

En aquest cas, utilitzarem la funció **dom\_to\_simple\_array()** i guardarem el resultat en una variable **\$sites**. La funció **dom\_to\_simple\_array()** fa l'acció contrària a la funció **simple\_array\_to\_dom()**, s'encarregarà d'extreure les dades d'un XML i preparar-les per a ser tractades com una *array*.

```
if(!isset($xml))$xml=dom_to_simple_array(abre_xml($xmlsites));

$sites=$xml["root"][0]["account"];
```

Per últim, només serà necessari recórrer aquesta *array* de comptes i començar a mostrar-la per pantalla.

```
<?php echo get_simple($site["title"]);?>
```

Observem que per a mostrar la informació correctament, fem servir la funció **get\_simple()**. Aquesta s'encarrega de mostrar correctament el contingut de la cadena de text que volem mostrar, tenint en compte la codificació adequada per a un projecte web compatible amb tots els navegadors i servidors.

```
function get_simple($txt)
{
    $txt=utf8_decode($txt);
    $str=strtr(strtolower($txt), "áàâãäåèéêëëïíîïóôõöùúüüñ &?ñ\"'€,%ç:;!¿?°²");
    $str=str_replace("--", "-", $str);
    return $str;
}
```

Quan un usuari faci clic sobre un dels seus *sites*, automàticament s'accedirà a la pàgina amb les estadístiques generals. Per fer-ho de forma correcte i que la pàgina de destí (en aquest cas: **detalle.php**) pugui mostrar correctament la informació, per defecte li passarem com a paràmetres el nom de l'hotel, l'acció que correspondrà a **"datos"** (paraula assignada per a mostrar el panell d'estadístiques generals) i com a període de temps, per defecte mostrarem les dades del dia anterior.

```
<a href="detalle.php?hotel=<?php echo get_simple($site["title"]);?>&accion=datos&dia=ayer">
```

Per a mostrar-se qualsevol de les pàgines amb diferent tipus de dades, sempre es crida a la pàgina **detalle.php**. Serà en aquesta, on segons el paràmetre **"accion"** que se l'hi passi, es cridarà a una altre pàgina PHP.

D'aquesta manera, s'ha aconseguit optimitzar el codi. Només existirà una única plantilla de pàgina de detall de la informació, però segons l'acció que se l'hi passi per paràmetre, es cridarà a un PHP concret. Aquest tindrà segons l'opció escollida, el conjunt d'accions i crides necessàries per mostrar la informació.

En **detalle.php**, el primer que es fa és recollir les dades del hotel, l'acció i el dia que s'han fet arribar anteriorment en el moment que l'usuari fa clic en un enllaç.

```

$hotel=$_POST["hotel"];
$accion=$_POST["accion"];
$dia=$_POST["dia"];

```

Per a poder accedir a la informació que la Api de Google proporciona, entre altres dades, es necessita un Id de Google. Aquest és únic per a cada compte que s'estigui realitzant un seguiment de les estadístiques amb Google Analytics.

Aquest id nosaltres ja el tenim. Anteriorment, quan hem accedit als *sites* vinculats al compte, hem emmagatzemat en el XML tota la informació del compte (només informació de gestió del compte, la informació estadística s'obté posteriorment amb crides determinades).

```
<account title="eMascaró Tourism" tableId="ga:000000" accountId="11111111" accountName="eMascaró Tourism" profileId="000000">
```

El primer que farem, serà obrir el XML on tenim guardada tota la informació dels comptes, convertir el contingut d'aquest en una *array* i finalment assignar el contingut a una variable anomenada: **\$sites** per a poder treballar amb les dades.

```

//Buscamos IG GOOGLE
if(!isset($xml))$xml=dom_to_simple_array(abre_xml($xmlsites));
$sites=$xml["root"][0]["account"];

```

A continuació recorrerem l'*array* per a buscar les dades del compte que estem intentant accedir.

```

foreach ($sites as $i=>$site){
    $varhotel=get_simple($site["title"]);
    if($hotel==$varhotel){
        $id_google="ga:".$site["profileId"];
        $hotel_actual=$site["title"];
        $idhotel=$i;
    }
}

```

Quan correspongui el valor de la variable **\$hotel** (aquesta variable l'havíem recuperat anteriorment: `$hotel=$_POST["hotel"];`) amb el del hotel que recorrem de l'*array*, llavors guardarem el valor del seu Id: `$id_google="ga:".$site["profileId"];`

A continuació, podem veure el conjunt de codi que farà que es mostri una informació o una altra, segons l'acció que l'usuari triï en el menú. Farem un “**include**” de la pàgina amb el codi corresponent a les consultes i accions necessàries per a emmagatzemar en *arrays* les dades a mostrar.

```
<?php
if($accion=="datos") include ("estadisticas_site.php");
else if($accion=="países") include ("estadisticas_pais.php");
else if($accion=="pagsvistas") include ("estadisticas_pagsvistas.php");
else if($accion=="palabras") include ("estadisticas_palabras.php");
else if($accion=="referidores") include ("estadisticas_referidores.php");
else if($accion=="navegadores") include ("estadisticas_navegadores.php");
?>
```

Per a no escriure més codi del necessari, en els diferents arxius PHP amb la programació per a cada situació, es guardaran les dades en arrays amb el mateix nom. Com només s'executarà un *include* en cada situació, mai es podrà donar el cas de sobreescrivre variables. Sempre estarem segurs que es mostra la informació adequada.

```
<?php for ($i=0; $i<count($titular); $i++)
{
?>
<!-- item -->
<tr>
<td class="">
<table cellpadding="0" cellspacing="0">
<tbody>
<tr>
<td class="pdtop10 txtnegro12 tleft"><?php echo utf8_decode($titular[$i]);?></td>
<td class="pdtop10 txtnegro12 tright"><?php echo $numeros[$i]?></td>
</tr>
</tbody>
</table>
</td>
</tr>
<?php } ?>
<!-- fin item -->
```

Les *arrays* **\$titular** i **\$numeros** tindran tota la informació tractada i preparada per a mostrar-la. Farem un bucle per a recórrer tots els valor i anar construint la taula de resultats.



## 7.2.4 Panell d'estadístiques generals

A continuació, s'accedeix a la pàgina el panell d'estadístiques generals. Aquesta és un conjunt d'informació bàsica però de gran utilitat.



Per accedir a aquesta informació, l'usuari ha d'haver triat l'opció de “*Estadístiques generales*” del menú de l'aplicació.

Al fer clic sobre aquesta opció el paràmetre **\$accion** tindrà el valor de: “**datos**” i per tant accedirà la pàgina **estadisticas\_site.php** per a recollir les dades.

```
if($accion=="datos") include ("estadisticas_site.php");
```

En l'arxiu **estadisticas\_site.php** el primer que es fa es recollir el valor del *login* i el *password* de la variable de sessió per a poder fer-la servir per recuperar les dades.

```
$login=$_SESSION["login"];
$password=$_SESSION["password"];
```

Tot seguit, és necessari preparar la ruta on guardarem l'arxiu XML amb les dades. Per a portar un control amb els noms dels arxius XML que guardaran la informació, sempre el nom de l'arxiu serà: **datos\_** i a continuació escriurem el Id identificatiu del compte al qual pertanyen les dades.

```
$xmldataos=$id;
$xmlldataos2=explode(":", $xmldataos);
$xmlldataos2=$xmldataos2[1];
$xmlldataos="xml/datos_".$xmldataos2.".php";
```

Ara ja es pot començar el procediment de recollida de dades. Serà necessari un altre cop crear l'objecte de la classe **analytics\_api()** i assignar el valor de **auth** que tenim a la variable de sessió a la corresponent del mètode.

```
$api = new analytics_api();  
$api->auth = $_SESSION["auth"];
```

Com que hem de tenir informació de diferents períodes de temps, serà necessari crear una array amb els períodes que volem.

```
$consultas=array("ayer","semana","mes","anyo");
```

En aquest cas, al tractar-se d'una consulta genèrica, la consulta extraurà tota la informació dels diferents períodes de temps automàticament, però en les altres tipus de consultes, serà necessari tractar el període de temps.

Com en altres casos farem un bucle per a recórrer tots els períodes de temps.

A continuació, guardarem en la variable **\$data** el valor del resultat de la crida de la funció **get\_summary()**.

```
foreach($consultas as $consulta){  
    $data = $api->get_summary($id, $consulta);
```

En aquesta funció, es realitzarà la crida a Google Analytics i retornarà tot un conjunt de dades. Ara es necessari tractar les dades o recuperar-les en variables per deixar-les llestes per a mostrar-les.

```
$result=array();  
$result["visitas"]=$data["ga:visits"];  
$result["paginasvistas"]=$data["ga:pageviews"];  
$result["tiempovisita"]=$data["average_time_on_site_formatted"];  
$result["paginasvisita"]=$data["pages_per_visit"];  
  
$data2 = $api->data($id, '', 'ga:bounces,ga:newVisits,ga:visits,ga:pageviews,ga:uniquePageviews','', $consulta);  
$result["visitasnuevas"]=$data2["ga:newVisits"];  
  
$xml["root"][0][$consulta]=array($result);
```

Com podem veure, les visites, pàgines vistes, temps de la visita i les pàgines per visita, són dades que s'obtenen directament amb l'anterior consulta. En canvi, per a obtenir el nombre de visites noves, és necessari realitzar una altra consulta amb paràmetres nous.

Anirem guardant els valors en el XML abans predefinit.

```
escribe_xml($xmldata, simple_array_to_dom($xml));
```

Finalment extreure'm les dades guardades en el XML i assignem els valors a unes determinades variables.

```
$datos=dom_to_simple_array(abre_xml($xmldata));
$visitas = $datos["root"][0][$dia][0]["visitas"];
$paginasvistas = $datos["root"][0][$dia][0]["paginasvistas"];
$tiempovisita = $datos["root"][0][$dia][0]["tiempovisita"];
$paginasvisita = $datos["root"][0][$dia][0]["paginasvisita"];
$visitasnuevas = round(($datos["root"][0][$dia][0]["visitasnuevas"]*100) / $datos["root"][0][$dia][0]["visitas"])."%";
```

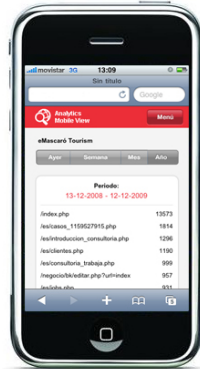
Per últim, és necessari crear les *arrays* **\$titular** i **\$numeros** amb les dades extretes per a tindre-les llestes per a mostrar-se per pantalla.

```
$titular=array("Visitas", "Páginas Vistas", "Páginas / Visita", "Tiempo en el Site", "Visitas Nuevas");
$numeros=array($visitas,$paginasvistas, $paginasvisita, $tiempovisita, $visitasnuevas);
```

En l'*array* **\$titular**, en cadascuna de les seves posicions, crearem un camp amb el nom del camp que estarà relacionat amb el valor de la mateixa posició de l'*array* **\$numeros**.

### 7.2.5 Resta de panells

A continuació, explicarem el funcionament del panell de les pàgines més vistes. Tot i així, l'explicació serà la mateixa per a tota la resta de panells, ja que el procediment d'extracció de dades i de mostrar-lo es exactament el mateix.



Per accedir a aquest informació, l'usuari ha d'haver triat l'opció de “Pàgines més vistes” del menú de l'aplicació.

Al fer clic sobre aquesta opció el paràmetre `$accion` tindrà el valor de: “**pagsvistes**” i per tant, accedirà la pàgina **estadisticas\_pagsvistes.php** per a recollir les dades.

```
else if($accion=="pagsvistes") include ("estadisticas_pagsvistes.php");
```

En l'arxiu **estadisticas\_pagsvistes.php** el primer que es fa es recollir el valor del *login* i el *password* de la variable de sessió per a poder fer-la servir per recuperar les dades.

```
$login=$_SESSION["login"];  
$password=$_SESSION["password"];
```

Tot seguit, és necessari preparar la ruta on guardarem l'arxiu XML amb les dades. Per a portar un control amb els noms dels arxius que guardarem la informació, el nom de l'arxiu sempre serà: **datos\_pagsvistes\_** i a continuació escriurem el Id identificatiu del compte al qual pertanyen les dades.

```
$xmldatos=$id;
$xmldatos2=explode(":",$xmldatos);
$xmldatos2=$xmldatos2[1];
$xmldatos_pagsvistas="xml/datos_pagsvistas_".$xmldatos2.".php";
```

Ara ja es pot començar el procediment de recollida de dades. Serà necessari, un altre cop, crear l'objecte de la classe **analytics\_api()** i assignar el valor de **auth** que tenim a la variable de sessió a la corresponent del mètode.

```
$api = new analytics_api();
$api->auth = $_SESSION["auth"];
```

Com que hem de tenir informació de diferents períodes de temps, serà necessari crear una *array* amb els períodes que volem.

```
$consultas=array("ayer","semana","mes","anyo");
```

Per aquest tipus de consultes, és necessari establir un rang de període de temps per a cada consulta. Per tant es crida a les funcions **get\_inicio()** i **get\_fin()**.

```
$inicioconsulta=get_inicio($consultas[$var]);
$finconsulta=get_fin($consultas[$var]);
```

Aquestes dues funcions s'encarreguen d'establir una relació entre el valor de la variable **\$dia** que hem recuperat de la variable de sessió i retornar un valor en format data.

```
function get_inicio($start){
    if($start == 'today') {
        $start = date('Y-m-d');
    }
    elseif($start == 'ayer') {
        $start = date('Y-m-d', strtotime('yesterday'));
    }
    elseif($start == 'semana') {
        $start = date('Y-m-d', strtotime('1 week ago'));
    }
    elseif($start == 'mes') {
        $start = date('Y-m-d', strtotime('1 month ago'));
    }
    elseif($start == 'anyo') {
        $start = date('Y-m-d', strtotime('1 year ago'));
    }
    else {
        if(!$start) $start = date('Y-m-d', strtotime('1 month ago'));
    }
    return $start;
}
```

Com en altres casos farem un bucle per a recórrer tots els períodes de temps.

A continuació, guardarem en la variable `$data` el valor del resultat de la crida de la funció **data()**.

En aquesta funció, es realitzarà la crida a Google Analytics i retornarà tot un conjunt de dades. Ara és necessari tractar les dades o recuperar-les en variables per deixar-les llestes per a mostrar-les.

```
foreach($consultas as $consulta){
    if($consultas[$var]!="ayer"){
        $data = $api->data($id, 'ga:pagePath', 'ga:pageviews', false, $inicioconsulta, $finconsulta, 10, 1, '');
    } else {
        $data = $api->data($id, 'ga:pagePath', 'ga:pageviews', false, $inicioconsulta, $inicioconsulta, 10, 1, '');
    }
}
```

És necessari remarcar el fet que, la única diferència entre tots els panells de informació, són els paràmetres que se l'hi passen a la funció **data()**.

A continuació, tenim un resum de les diferents crides per a cada panell de la funció **data()**.

Panell de **pàgines més vistes**:

```
$data = $api->data($id, 'ga:pagePath', 'ga:pageviews', false, $inicioconsulta, $inicioconsulta, 10, 1, '');
```

Panell de **paraules clau**:

```
$data = $api->data($id, 'ga:keyword', 'ga:pageviews', false, $inicioconsulta, $inicioconsulta, 11, 1, '');
```

Panell de **països**:

```
$data = $api->data($id, 'ga:country', 'ga:visits', false, $inicioconsulta, $inicioconsulta, 10, 1, '');
```

Panell de **referidors**:

```
$data = $api->data($id, 'ga:source', 'ga:pageviews', false, $inicioconsulta, $inicioconsulta, 11, 1, '');
```

Panell de **navegadors**:

```
$data = $api->data($id, 'ga:browser,ga:browserVersion', 'ga:visits', false, $inicioconsulta, $inicioconsulta, 1000, 1, '');
```

Com podem observar la diferencia entre les diferents crides, és el tipus de “*metric*” o “*dimension*” que es passen.

Una altre dada que podem observar en aquesta crida, és el número 10 que correspon a que es vol obtenir 10 resultats de cada crida. És important no voler obtenir molts resultats si no són necessaris, ja que el temps de descàrrega de la crida serà superior.

Un cop explicada aquesta petita diferencia, ja podem continuar amb l’explicació.

Ara és necessari guardar els valors en el XML abans predefinit.

```
escribe_xml($xmldatos_pagsvistas, simple_array_to_dom($xml));
```

Finalment extreure’m les dades guardades i assignem els valors a unes determinades variables.

```
$datos=dom_to_simple_array(abre_xml($xmldatos_pagsvistas));
for ($i=0; $i<count($datos["root"][0][$dia]); $i++){
    $pagina = $datos["root"][0][$dia][$i]["pagina"];
    $visitas = $datos["root"][0][$dia][$i]["visitas"];
}
```

Per últim, és necessari crear les *arrays* **\$titular** i **\$numeros** amb les dades extretes per a tindre-les llestes per a mostrar-se per pantalla.

```
$titular[$i]=$pagina;  
$numeros[$i]=$visitas;
```

En l'*array* **\$titular**, en cadascuna de les seves posicions, crearem un camp amb el nom del camp que estarà relacionat amb el valor de la mateixa posició de l'*array* **\$numeros**.

A partir d'aquí, la informació ja està preparada per mostrar-se per pantalla.



## 8. Conclusions

La realització d'aquest projecte m'ha suposat moltes hores dedicades, tant en horari laboral, com en hores realitzades a casa. Des d'un principi vaig haver de prendre la decisió de comprometre'm amb l'empresa i organitzar-me per a distribuir la feina i complir la data d'entrega, tant amb el client que realitzava la petició de les aplicacions, com amb els terminis d'entrega de la universitat.

Aquest projecte ha estat més complicat del que en un principi m'imaginava. El fet de trobar dos tipus d'usuaris ben diferenciats, va suposar haver de realitzar dues aplicacions diferents i haver d'invertir encara més hores al desenvolupament.

Tot i així, a títol personal, la realització d'aquest projecte m'ha obligat a aprendre un llenguatge nou de programació (AS3), perfeccionar el PHP i sobretot, conèixer de més aprop les necessitats dels clients, alhora de demanar un producte.

També m'he adonat que a vegades és millor consultar un dubte a un company, que passar-te hores i hores intentant solucionar-ho tot sol.

Finalment, puc dir que estic content d'haver acabat exitosament el projecte, que el client per a qui anava destinada l'aplicació estigués completament satisfet i que l'empresa eMascaró Crossmedia, pugui seguir oferint aquest producte a altres clients.



## 9. Bibliografia

### Referències bibliogràfiques:

[1] Joey Lott, *ActionScript 3. Patrones de diseño*, Ed. Anaya Multimedia, 2007.

### Localitzacions URL

[1] <http://www.php.net>, Documentació oficial de PHP

[2] <http://www.maestrosdelweb.com/>, Blog amb informació i tutorials de programació web

[3] <http://www.desarrolloweb.com/manuales/>, Manuals de PHP i JavaScript.

[4] <http://www.cristalab.com/tutoriales/>, Manuals de Flash i ActionScript

[5] <http://code.google.com/intl/es-ES/apis/analytics/docs/gdata/gdataDeveloperGuide.html>,  
Documentació Api de Google Analytics



## 10. ANNEX: Contingut del CD

El CD que acompanya la documentació del projecte inclou els següents arxius:

Contingut	Format	Carpeta
Documentació memòria	Word / PDF	Documentació
Article del projecte	Word / PDF	Article
URLs, usuaris i contrasenyes	Word / PDF	acces

