

# Reconstruyendo para un mundo fragmentado

Construcción de un nuevo sitio  
web para la Carrera

Marco González Luengo  
Para la Ayudantía de  
Mantenimiento de la Página  
Web de la carrera

# Antes de que los teléfonos tuvieran inteligencia

## Introducción

### Sitios web para móviles

El año 2011 el W3C hizo el último llamado para revisión de las tecnologías que estarán alrededor del nuevo estándar HTML5<sup>1</sup>, tecnología que se lleva desarrollando desde el 2008, que se espera esté en su recomendación estable para el 2014, sin embargo el estándar se encuentra funcional en la mayoría de los navegadores modernos.

Alrededor de HTML5 se consolidan tecnologías como CSS3<sup>2</sup>, JavaScript<sup>3</sup> y SVG<sup>4</sup>, entre muchas otras, que favorecen la interoperabilidad y consistencia de las páginas web entre diferentes navegadores y procesadores de contenido, así como una separación entre el contenido, la presentación y la lógica. Con esto en mente, se propicia una compatibilidad a futuro con navegadores que soporten el estándar, además de aprovechar las ventajas que las diferentes tecnologías en sus especificaciones ofrecen.

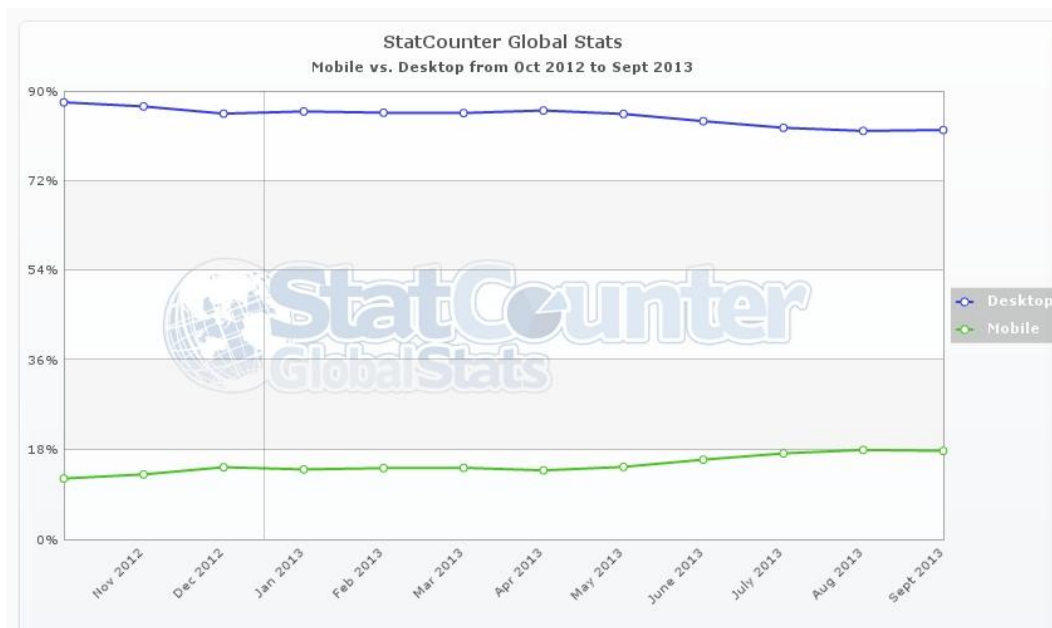


Ilustración 1: Presencia de navegadores de escritorio versus móviles en los últimos 12 meses

<sup>1</sup><http://www.w3.org/TR/html5/>

<sup>2</sup><http://www.w3.org/Style/CSS/current-work>

<sup>3</sup><http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

<sup>4</sup><http://www.w3.org/TR/SVG/>

## Interfaz de programación de aplicaciones

Los sitios web fuertemente comunicados consisten en que, aparte de la presentación habitual del sitio, se generan interfaces que permiten la entrega del contenido del sitio a otras aplicaciones. Tecnologías como RSS<sup>5</sup> y JSON<sup>6</sup> permiten y facilitan la alimentación de contenido a aplicaciones y servicios, además de funcionar como una RESTful API<sup>7</sup> para aplicaciones que necesiten acceder a esta información.

## Arquitectura MVC

El concepto de **Modelo-Vista-Controlador** existe hace bastante tiempo en la informática, pero ha sido durante los últimos años donde ha cobrado bastante relevancia en el desarrollo de proyectos y aplicaciones, aprovechando el paradigma de Programación Orientada a Objetos, también trasladado a la web.

Esto viene de la mano con los frameworks<sup>8</sup>, conjuntos de bibliotecas que se han hecho para poder favorecer la rapidez de escritura de código, conteniendo rutinas habituales de programación accesibles mediante métodos definidos en su API. Así se evita *reinventar la rueda* cada vez que se elaboran nuevos sistemas.

El conjunto de la programación objetual, junto con un framework y la arquitectura MVC ayudan a generar software mejor escrito, más rápido en tiempos de construcción y más fácil de mantener a futuro, ya que refuerzan una mejor organización del código y la adopción de buenas prácticas de programación.

## Problemática

Se ha planteado crear un sitio web para la carrera completamente nuevo, tanto en front-end como en back-end. La idea es tener un sitio web que pueda ser visualizable en navegadores de escritorio, navegadores móviles, además de tener **capacidades de aplicación web** y servir diferentes tipos de datos, para favorecer la interoperabilidad con otros sistemas a futuro.

Esto implica la creación de un nuevo CMS que heredará el modelo de datos del CMS anterior, pero creado bajo los cánones actuales de aplicaciones web.

---

<sup>5</sup><http://en.wikipedia.org/wiki/RSS>

<sup>6</sup><http://en.wikipedia.org/wiki/JSON>

<sup>7</sup>[http://en.wikipedia.org/wiki/REST\\_API#RESTful\\_web\\_APIs](http://en.wikipedia.org/wiki/REST_API#RESTful_web_APIs)

<sup>8</sup>[http://en.wikipedia.org/wiki/Software\\_framework](http://en.wikipedia.org/wiki/Software_framework)

# Situación previa



Ilustración 2: Captura de pantalla del sitio web actual



Ilustración 3: Captura de pantalla del sitio web actual visto en un móvil Android

El sitio web<sup>9</sup> actual está construido con técnicas que se arrastran desde los tiempos de HTML3, cuando CSS aún no se había propuesto como estándar, además de utilizar plugins propietarios como Flash. Esto, por consecuencia, impide la interoperabilidad del sitio, sobre todo en móviles.

Además de esto, la arquitectura del sitio está enfocada de manera tal que el acceso al modelo de datos, la lógica de negocios y las vistas de usuario no están debidamente separadas, lo que dificulta la comprensión del código y su consiguiente mantenimiento.

<sup>9</sup><http://www.inf.jmc.utfsm.cl>

# Manos a la obra

## Tecnologías consideradas

Por dentro

Framework

Utilizaremos CodeIgniter 2.1.4<sup>10</sup> como nuestro framework, debido a la flexibilidad y tolerancia que tiene con diferentes prácticas de desarrollo en PHP. Tiene un buen abanico de *helpers* y *libraries* que facilitan varios aspectos de hacer una aplicación web, como validación de formularios, paginación, envío de correos, acceso mediante diferentes motores de bases de datos y un amplio etcétera, además de ser fácilmente extensible. Cuenta con una curva de aprendizaje rápida, amplia comunidad de apoyo y con un manual de uso claro e integrado en la instalación.

Requiere tener instalado PHP 5.1.6 o superior en el servidor, se recomienda PHP 5.3.

Replicar el entorno de producción

Una de las tareas más complicadas es lograr replicar el entorno de producción de una aplicación web. Conjuntos de aplicaciones como XAMPP, WAMP, MAMP o AppServ, si bien son útiles, no reflejan fielmente un entorno de producción, lo que provoca problemas de compatibilidad del código e interpretación de las capacidades del servidor con la plataforma.

Para subsanar este problema es que se usarán técnicas de virtualización que mantendrán una imagen idéntica al servidor de producción. Esto se logrará mediante el uso de Oracle VirtualBox<sup>11</sup> y Vagrant<sup>12</sup>, que apoyan y facilitan la creación rápida de máquinas virtuales provisionables con el conjunto de programas que se encontrarán en el servidor de producción.

---

<sup>10</sup><http://ellislab.com/codeigniter>

<sup>11</sup><http://www.virtualbox.org/>

<sup>12</sup><http://www.vagrantup.com/>

Por fuera

## Front-end framework

Uno de los requerimientos es que el sitio web sea responsivo<sup>13</sup>, lo que implica la creación de una grilla de columnas que se adapte al ancho de la pantalla que visualiza y que se tenga un enfoque *mobile first* a la hora de desarrollar. En el sitio web actual esto es imposible debido a que el *layout* está hecho con tablas HTML lo que impide lograr responsividad, no es adecuado para la semántica<sup>14</sup> y puede significar problemas en el SEO a futuro.

Para lograr este objetivo nos apoyaremos en Bootstrap 3<sup>15</sup>, que tiene soporte para CSS Media Queries, además de aportar con una propuesta visual consistente en todos los elementos de la UI y otros componentes como carruseles, *jumbotron*, barras de navegación, *dropdowns* y más.

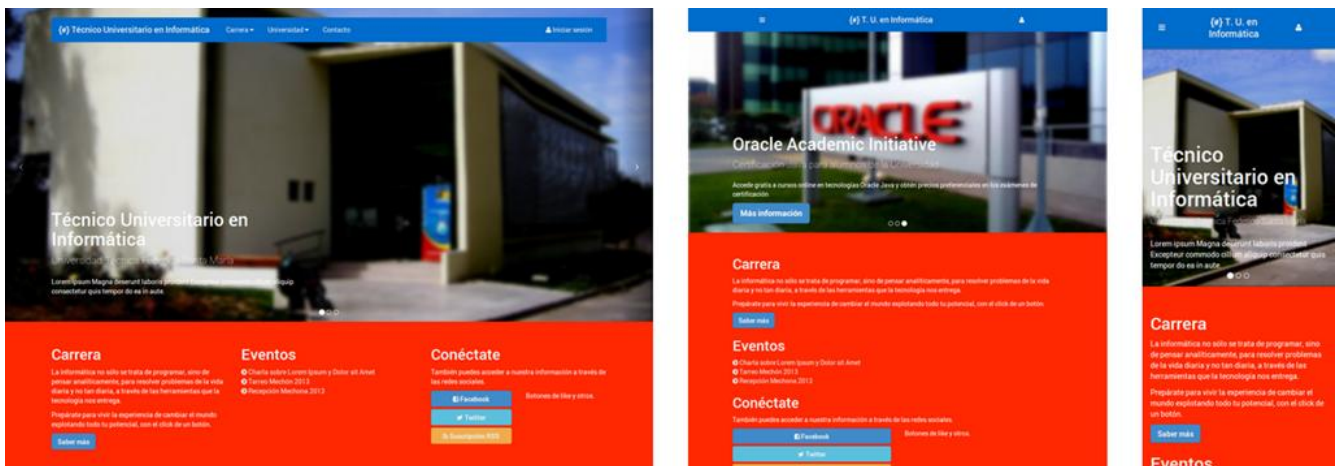


Ilustración 4: Comparativa de las 3 presentaciones principales: escritorio, tablet y móviles

<sup>13</sup><http://alistapart.com/article/responsive-web-design>

<sup>14</sup><http://webmasters.stackexchange.com/questions/6036/why-arent-we-supposed-to-use-table-in-a-design/>

<sup>15</sup><http://getbootstrap.com>

## Complementos

Para apoyar el desarrollo y para variados efectos se utilizarán bibliotecas como jQuery<sup>16</sup> y html5shiv<sup>17</sup>, entre otras, encargadas de brindar una mejor experiencia de usuario al navegar el sitio.

También, rescatando parte de lo desarrollado anteriormente para el sitio, se implementará en paralelo el acceso a la información del sitio a través de tecnología RSS y una RESTful API para entrega de contenido encapsulado con JSON. Esto presenta ventajas a la hora de alimentar lectores RSS como feedly y permite a futuro el desarrollo de aplicaciones para web, escritorio o móviles que puedan extraer información mediante la RESTful API.

Uno de los objetivos principales es lograr una comunicación expedita entre el sitio web de la carrera y la página oficial de la carrera<sup>18</sup> en Facebook, propiciando la difusión de la información a través de distintos medios.

## Desarrollo actual

Uno de los aspectos con mayor urgencia de implementación era la comunicación entre el sitio web y una aplicación de Facebook que permitiese *cross-posting*. Para lograr esto se elaboró un generador de RSS y una aplicación en Facebook que leyera ese RSS y lo publicara como posts de timeline de Facebook.

### Generar RSS

Después de revisar las especificaciones de un RSS tipo, se creó una plantilla que ocupara los mismos métodos de obtener información de la DB que la página principal. La idea es obtener una lista de títulos, el contenido de las noticias, fecha de publicación y la URL del post.

### Conectar con Facebook

Para lograr que el RSS se publique en Facebook debemos utilizar una aplicación llamada RSS Graffiti<sup>19</sup> que automatiza el proceso de obtención del RSS, parseado y publicación en Facebook.

---

<sup>16</sup><http://jquery.com/>

<sup>17</sup><http://code.google.com/p/html5shiv/>

<sup>18</sup><http://www.facebook.com/tuinformatica.usm>

<sup>19</sup><http://www.rssgraffiti.com>

# Apéndice

soul/feed/index.php

```
<?php
header("Content-type: text/xml");
date_default_timezone_set("America/Santiago");
include 'config.php';
include 'opendb.php';
include 'querydb.php';

$zonahoraria = (date("T") == "CLST") ? " -0300" : " -0400" ;
if (isset($_GET["mostrar"])) {
    $cantidad = $_GET["mostrar"];
} else {
    $cantidad = 5;
}
echo "<?xml version='1.0' encoding='UTF-8' ?>\n";
?>

<rss version="2.0"

xmlns:atom="http://www.w3.org/2005/Atom">
<channel>
    <title>USM - Técnico Universitario en Informática</title>
    <atom:link href="http://www.inf.jmc.utfsm.cl/" rel="self"
type="application/rss+xml" />
    <link>http://www.inf.jmc.utfsm.cl/</link>
    <lastBuildDate><?php echo date("D, d M Y H:i:s",
strtotime(fechaUltimoArticulo())) . $zonahoraria ?></lastBuildDate>
    <description>Canal oficial de noticias de la carrera de Técnico
Universitario en Informática de la Universidad Técnica Federico Santa María,
Sede Viña del Mar</description>

<?php
    foreach (listarNoticias($cantidad) as $noticia) :
?>
        <item>
            <title><?php echo $noticia{'not_titulo'} ?></title>
            <pubDate><?php echo date("D, d M Y H:i:s",
strtotime($noticia{'not_fecha_grabacion'} . ' ' .
$noticia{'not_hora_grabacion'})) . $zonahoraria ?></pubDate>
            <link>http://<?php echo $_SERVER['SERVER_NAME']
?>/noticia_desplegar.php?id=<?php echo $noticia{'codigo_noticia'} ?></link>
            <guid>http://<?php echo $_SERVER['SERVER_NAME']
?>/noticia_desplegar.php?id=<?php echo $noticia{'codigo_noticia'} ?></guid>
            <description><?php echo "<![CDATA[" . nl2br($noticia{'not_descripcion'})
. "]]>" ?></description>
        </item>
<?php
    endforeach;
    include 'closedb.php';
?>
</channel>
</rss>
```



# Configurar RSS Graffiti

**Source Settings: USM - Técnico Universitario en Inf**

**Basic** **Advanced**

**FEED SETUP**

Feed Title: USM - Técnico Universitario en Informática

Feed URL: http://www.inf.jmc.ut fsm.cl/soul/feed/

Source Name Override (not required)

Source URL Override (not required)

Maximum Posts per Update: System Maximum

Post Order per Update: Publish older posts first

**FACEBOOK POST PREVIEW**

Live Data Preview

Placeholder text: Lorem ipsum dolor sit amet, consectetur adipiscing elit. In auctor tempor ligula, dignissim faucibus tortor malesuada non. Nulla in augue at est dictum porttitor.

Feed item title

Save Changes Cancel

Ilustración 4: Configuración de RSS origen

**Target Settings**

**Basic** **Bit.ly Options**

**TARGET SETUP**

Choose Target: Técnico Universitario en Info

Publish on Behalf of: Técnico Universitario en Info

**POST STYLE**

The following options affect how the post appears in Facebook. For more information on post style, visit our support page.

|   |  |   |
|---|--|---|
| <input checked="" type="radio"/> <b>Standard</b><br>Recommended for most feeds. Publishes a text excerpt and any available media (image, audio or video.) | <input type="radio"/> <b>Compact</b><br>Intended to publish short stories from microblogs without changing your Facebook status. | <input type="radio"/> <b>Status Updates</b><br>Similar to "Compact" style, but instead of just writing on your wall it will also update your Facebook status. |
| <input type="checkbox"/> Move Text to Box<br>Shorten to 250   |  | <input type="checkbox"/> Append URL   |

**FACEBOOK POST PREVIEW**

Live Data Preview

Placeholder text: Lorem ipsum dolor sit amet, consectetur adipiscing elit. In auctor tempor ligula, dignissim faucibus

Save Changes Cancel

Ilustración 5: Configuración de posts objetivo en Facebook