

# Ratchet: Framework Para Desarrollo Web App Para Móvil

[Ratchet](#) es un [Framework](#) para el desarrollo front-end de aplicaciones web para móviles, basadas en [HTML](#), [CSS](#) y [JavaScript](#). [Ratchet 2.0 fue lanzado hace poco](#), así que es un buen momento para empezar a aprender a construir aplicaciones con este Framework. Si lo que desea es ver a **Ratchet** en la acción, haga clic en este [vinculo de demostración](#) para su dispositivo móvil.



## ¿Que Es Ratchet Framework?

Como sabemos, en la actualidad, si no sabes usar un **Framework** no sabes mucho :S Un **Framework**, te ayuda a realizar determinada tarea en menos tiempo, ya que este tiene incorporado «la mayoría de trabajo». Solo tendras que saber su

sintaxis y listo, tendrás tu proyecto casi finalizado. En el caso de **Ratchet**, te provee la facilidad de desarrollar el [front-end](#) para dispositivos móviles en poco tiempo.

**Ratchet** tiene como objetivo proporcionar a los diseñadores y desarrolladores un marco para la creación de **aplicaciones web** para móviles y que funcionen como si fueran nativas. El **Framework** es desarrollado por el mismo equipo que creó el popular [Bootstrap](#), lo que significa que **Ratchet** es de alta calidad. Si usted desea orientar el desarrollo en este marco para iOS, Android, o ambos, **Ratchet 2.0** está listo para el trabajo.

## Primeros Pasos Con Ratchet

### Paso 1: Descargar Ratchet

Para descargar **Ratchet** , dirígete al [sitio de descarga](#) y haz clic en el gran [botón de descarga](#). Para nada difícil este paso, ¿verdad?

### PASO 2 : Iniciar Un Servidor

A pesar de que **Ratchet** ayuda de gran manera en el código de **front-end**, necesitamos ejecutar el proyecto en un servidor web, como es de costumbre en este tipo de desarrollo. Si estás en **Windows**, recomendamos usar **WAMP** para poner en marcha un servidor para el desarrollo local. Si estás en [Mac OS X](#), el uso de un servidor **Python** es trivial. Abre la aplicación Terminal y navega hasta el directorio donde están los archivos de **Ratchet** utilizando el comando de cambio de directorio. Tendrá que modificar la ruta de a cuerdo a como se asigna a su directorio de proyecto, en nuestro caso: `$ cd ~/dev/MyRatchetProject.`

**Mac OS X** viene con **Python** instalado previamente , por lo que sólo tienes que escribir este comando para iniciar el servidor

: `$ python -m SimpleHTTPServer` By default, este debe iniciar un servidor en el puerto **8000** Escriba la siguiente dirección en su navegador y ya está bueno para ir: **`http://localhost:8000.`**

## **PASO 3: Habilite Eventos Táctiles**

**Ratchet** no funcionará correctamente en un navegador de escritorio con la configuración predeterminada, pero hay algunas cosas que usted puede hacer para crear un entorno de desarrollo. El método más fácil es simular eventos de toque, ya sea en [Chrome](#) o [Firefox](#). Sin embargo, recomendamos desarrollar utilizando un [emulador de SmartPhones](#).

En **Mac OS X**, puede instalar **XCode** y utilizar el emulador de [iPhone](#) para ver su sitio como si estuviera en **Mobile Safari**. Alternativamente, usted podría incluso sólo tiene que utilizar un teléfono real. Si está conectado a la misma red Wi-Fi como el equipo, escriba la dirección IP de su máquina de desarrollo junto con el puerto que está utilizando para desarrollar, y usted debería ver su sitio. Su dirección IP y el puerto varían, pero la dirección se verá algo como esto: **`http://192.168.1.0:8000.`**

## **PASO 4: Ejemplos De Descarga**

Se ha combinado varias de las plantillas de **Ratchet** y componentes en un ejemplo, esto nos facilita las cosas, teniendo una buena base para echar a andar el proyecto.



El proyecto de ejemplo que incluye **Ratchet 2.0.2** ( la última versión hasta el día de hoy), pero si quieres hacer algo más que experimentar, es mejor que veas los ejemplos que hay en su [sitio web](#).

## Estructura Básica De Una App Con Ratchet

Cada vista en **Ratchet** es una página **HTML** independiente que puede conectarse con las demas como las páginas web normales. En cada página de **Ratchet**, hay tres reglas a tener en cuenta:

- Si desea agregar una barra fija ( utilizando la clase `.bar`) a la página, asegúrese de que sea lo primero que en el interior del `<body>`
- Todo lo que no está en la clase `.bar` debe ir dentro de un div con la clase `.content`
- Asegúrese de añadir las etiquetas `<meta>` propias ( incluidos en los ejemplos ).

**Ratchet** tiene un estilo básico decente, pero si usted quiere que su aplicación se vea como una aplicación nativa de **iOS** o de **Android**, que puede incluir adicionalmente al **CSS**. Sólo

tienes que añadir ya sea **ratchet-theme-ios.css** ó **ratchet-theme-android.css** según sea necesario. He aquí un fragmento de código que le da un ejemplo de lo que debe ir en el interior del elemento `<body>` :

```
[html]

<body>

<!-- Make sure all your bars are the first things in your
<body> -->
<header class="bar bar-nav">
<a class="icon icon-left-nav pull-left" href="/index.html"
data-transition="slide-out"></a>
<h1 class="title">Typography</h1>
</header>

<!-- Wrap all non-bar HTML in the .content div (this is
actually what scrolls) -->
<div class="content">
<div class="content-padded">
<h3>h3. Heading</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna
aliqua.</p>
</div>
</div>

</body>

[/html]
```

## Componentes De Ratchet

**Ratchet** incluye una gran variedad de **componentes pre-construidos** que vienen para ser usados como clases de **CSS**. A continuación son sólo algunos de los aspectos más destacados. Para la lista completa de los componentes, echa un vistazo a la [documentación oficial de Ratchet](#).

# TIPOGRAFÍA

Casi cada Framework para front-end incluye una buena tipografía predeterminada y **Ratchet** no es una excepción. Sólo tiene que utilizar los elementos principales y los párrafos como lo haría normalmente. Además, si desea que el contenido se muestre en un formato específico de Ratchet, puede envolver el párrafo en la siguiente clase: **.content-padded**.

## VISTAS DE TABLA

Para crear una vista de tabla, agregue la clase **.table-view** a una lista desordenada y agregue la clase **.table-view-cell** a cada elemento de la lista. De manera predeterminada se creará una lista que se extiende por un borde al otro, pero se puede envolver opcionalmente la lista desordenada en la clase **.card**, para añadir algo de padding. He aquí un fragmento de código de ejemplo del proyecto que se mencionó al principio de este post.



```
[html]
```

```
<div class="card">  
<ul class="table-view">
```

```
<li class="table-view-cell">
<a class="push-right" href="examples/typography.html">
<strong>Typography</strong>
</a>
</li>
<li class="table-view-cell">
<a class="push-right" href="examples/tab-bar.html">
<strong>Tab Bars</strong>
</a>
</li>
</ul>
</div>

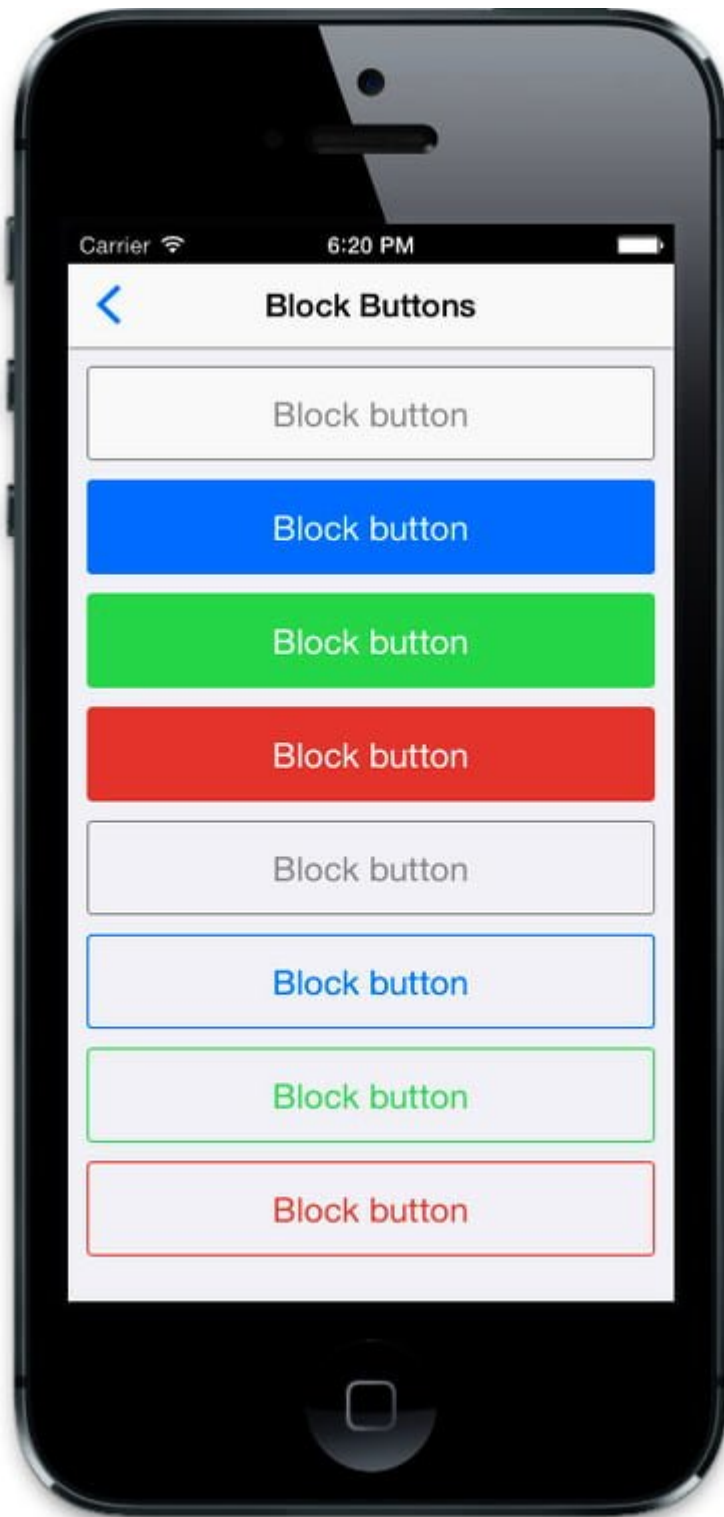
[/html]
```

## TRANSICIONES DE PAGINA CON PUSH.JS

**Push.js** (incluido en **Ratchet**) podrá escuchar los eventos de los clics en la página y aplicar transiciones **automáticamente**. Para añadir una transición, sólo tiene que añadir el atributo **data-transition** para anclar elementos con uno de tres posibles valores: **slide-in**, **slide-out**, ó **fade**. Usando el ejemplo anterior, podemos hacer que las páginas se deslicen, solo añadiendo el siguiente enlace: **<a href=»examples/tab-bar.html» data-transition=»slide-in»>**

## BOTONES EN BLOQUE

**Ratchet** incluye diferentes estilos de botón. El estilo «block button» es para los botones que debe abarcar el ancho del contenedor primario. Justo así se vera esta propiedad:



[html]

```
<!-- Wrap all non-bar HTML in the .content div (this is  
actually what scrolls) -->
```

```
<div class="content">
```

```
<div class="content-padded">
```

```
<button class="btn btn-block">Block button</button>
```

```
<button class="btn btn-primary btn-block">Block
```



```
button</button>
<button class="btn btn-positive btn-block">Block
button</button>
<button class="btn btn-negative btn-block">Block
button</button>

<button class="btn btn-block btn-outlined">Block
button</button>
<button class="btn btn-primary btn-block btn-outlined">Block
button</button>
<button class="btn btn-positive btn-block btn-outlined">Block
button</button>
<button class="btn btn-negative btn-block btn-outlined">Block
button</button>
</div>
</div>

[/html]
```

## SLIDERS

**Ratchet** tiene un componente integrado para el manejo de Sliders que permite a los usuarios navegar por las imágenes usando **gestos swipe**. La construcción de un control deslizante implica el uso de varias clases **CSS**. El uso adecuado de estas clases, da como resultado un buen efecto como si fuera nativo en la app, vale la pena saber usarlo:



## Finalmente!

Este artículo fue hecho simplemente con la intención de ofrecer un breve resumen de **Ratchet**. Si desea obtener más información, echa un vistazo a la [documentación de Ratchet](#), donde se puede leer acerca de todos los componentes de este

**Framework.** Conocías esta herramienta? o has usado algún otro **Framework** para el mismo fin?