

Proyecto de Titulación: Alternativas de Solución
“Aplicación móvil basada en WebSocket, para
aulas SCALE-UP”

Presentado por:	Cristóbal Ganter
ROL UTFSM:	2604155-4
Profesor guía:	Werner Creixell
Fecha:	16.12.2013

Alternativas posibles de solución

El trabajo propuesto en los informes anteriores [1] [2], y en general el desarrollo de aplicaciones web, contempla la integración de diversas tecnologías. Una aplicación web consiste de dos partes claramente diferenciadas: el front-end y el back-end. El front-end es la parte de la aplicación con la cual interactúa el usuario y suele estar compuesto por código HTML y, opcionalmente código CSS y Javascript. El back-end es la parte que está encargada de atender los requerimientos de los navegadores; generar y transmitir el front-end; y administrar el contenido ya sea generado por los usuarios o por el administrador del servicio. El front-end y el back-end interactúan normalmente a través del protocolo HTTP. Sin embargo en este trabajo también se utilizará la comunicación a través de WebSocket.

Como se puede observar el desarrollo de una aplicación web consta de la integración de diferentes tecnologías. Lo cual implica que deben seleccionarse las mejores alternativas para cada parte de la aplicación. En concreto, evaluaré las alternativas de solución para dos apartados diferentes: **frameworks** para el back-end y para el front-end.

Back-end

Para el back-end no existe un estándar que defina un lenguaje específico con el cual escribirlo. Este componente de una aplicación web puede ser escrito en cualquier lenguaje de programación que soporte la comunicación a través del protocolo TCP. Debido a esto se han desarrollado frameworks para diversos lenguajes de programación, que facilitan el desarrollo de back-ends.

En este trabajo solo evaluaré frameworks para el lenguaje de programación Python. Esta es una decisión personal fundamentada en el hecho de que Python es el lenguaje de programación con el cual mas experiencia tengo. Además existe bastante variedad de frameworks escritos en Python para el desarrollo web. Algunos de ellos utilizados en aplicaciones ampliamente conocidas.

En Wikipedia es posible encontrar una lista con los frameworks disponibles para Python [3]. De esta lista se han analizado todos los frameworks que han sido actualizados durante el último año. De esta forma se omiten frameworks que puedan estar en desuso. Entre todos los frameworks analizados se han seleccionado tres para ser comparados. Estos son: Django, Flask y Tornado.

Django

Django es un framework de alto nivel, basado en la arquitectura Modelo-Vista-Controlador, que permite crear paginas web de manera rápida y efectiva. Según la pagina web de django: “Django se centra en la automatización tanto como sea posible ...” [4]

Algunas paginas web producidas con Django son: [Pinterest](#), [Instagram](#) y [The New York Times](#).

Flask

Flask es un microframework para Python basado en Werkzeug y Jinja 2 [5]. Está estructurado como una librería para Python que incluye un servidor web, plantillas HTML Jinja2 y soporte para el manejo de cookies.

Algunas paginas web producidas con Flask son: [Indico](#) y [Battlefield3 Development News Aggregator](#).

Tornado

Tornado es un framework web y biblioteca de red asíncrona para Python, desarrollado originalmente en FriendFeed. Mediante el uso de conexiones no bloqueantes, Tornado puede escalar a decenas de miles de conexiones abiertas, lo que es ideal para long polling, WebSocket, y otras aplicaciones que requieren una conexión de larga duración para cada usuario. [6]

Algunas paginas web producidas con Django son: [IPython Notebook](#), [bitly](#) y [Fetchnotes](#).

Front-end

Diseñar la interfaz de una aplicación web no es una tarea sencilla. HTML y CSS inicialmente fueron creados para escribir documentos de texto. Hoy en día el uso de estos lenguajes a cambiado mucho. Si bien HTML5 y CSS3 han traído grandes avances para crear interfaces cada vez mas complejas, aún sigue siendo difícil escribir interfaces para aplicaciones. Éstas son mas cercanas a las interfaces que conocemos hoy para equipos de escritorio que a la de una pagina web común. Debido a esto existen variados frameworks que permiten crear interfaces de aplicaciones web de manera sencilla.

Entre las características deseables en una interfaz de aplicación web, que dificultan su creación, se encuentran:

- Soporte para múltiples tamaños de pantalla
- Soporte para múltiples navegadores
- Adaptación del aspecto visual al dispositivo actual
- Utilización de controles complejos, como menús desplegables o sliders

En la elaboración de este informe se han analizado una amplia variedad de frameworks para la creación de interfaces web. Gran parte de ellos estaba basado en Javascript, lo cual no es deseable puesto que se pierde la separación entre contenido, estilo y funcionalidad (HTML, CSS y Javascript respectivamente). Otra parte de ellos era inestable, no funcionaba bien en mis dispositivos o estaban en desarrollo.

Finalmente entre todas las alternativas disponibles, se han seleccionado JQuery Mobile y Bootstrap. También se considerará no utilizar un framework. Creo que a pesar de las dificultades, me sería posible abordar el diseño de una interfaz de aplicación web utilizando solamente HTML, CSS y Javascript.

JQuery Mobile

JQuery Mobile es un sistema de interfaz de usuario unificado, basado en HTML5, para todas las plataformas de dispositivos móviles populares, construido sobre los sólidos fundamentos de jQuery y jQuery UI. [7]

Bootstrap

Bootstrap es un framework para front-ends móviles. Es elegante, intuitivo y potente, permitiendo un desarrollo web más rápido y fácil. [8]

Bootstrap incluye clases CSS que permiten manejar el layout de una página, un toolkit e incluso iconos.

No utilizar un framework

La utilización de los frameworks propuestos (sobre todo JQuery Mobile) agregan un overhead que podría ser innecesario para este proyecto. Por lo cual estimo que una buena alternativa es crear la interfaz de usuario desde cero utilizando solo HTML, CSS y Javascript.

Ventajas y desventajas

Back-end

Django

Ventajas:

- Viene con un mapeador objeto-relacional incorporado con el que se puede crear y acceder una base de datos directamente desde Python.
- Es un framework bien estructurado, lo cual permite un alto nivel de estandarización entre diferentes aplicaciones.
- Viene con un sistema de plantillas que permite la modularización del código HTML e incorporar información dinámica al momento de responder a los requerimientos.
- Viene con una gran cantidad de componentes de front-end y back-end pre-programados.

Desventajas:

- Por ser tan estructurado requiere una mayor cantidad de tiempo de aprendizaje.
- Viene con muchas cosas que probablemente no se usen en este proyecto.
- No incluye soporte para Websocket por omisión. Existen cuatro paquetes diferentes que implementan la comunicación por Websocket.

Flask

Ventajas:

- Viene con un sistema de plantillas que permite la modularización del código HTML e incorporar información dinámica al momento de responder a los requerimientos.
- La estructura de una aplicación Flask destaca por su orden y simplicidad.
- Está muy bien documentado.
- Requiere muy poco tiempo de aprendizaje

Desventajas:

- No incluye soporte para Websocket por omisión. Existe un paquete que implementa la comunicación por Websocket.
- Los controladores se definen utilizando funciones en vez de clases, por lo que no es posible implementar controladores complejos, herencia, encapsular variables de estado, etc.

Tornado

Ventajas:

- Viene con un sistema de plantillas que permite la modularización del código HTML e incorporar información dinámica al momento de responder a los requerimientos.
- El sistema de plantillas permite escribir Python directamente.
- El sistema de plantillas permite incorporar módulos con sus propios estilos y Javascript.
- Incluye una capa fina de abstracción para el manejo de bases de datos.
- Es un framework poco estructurado, lo cual permite un alto nivel de flexibilidad.
- Viene con algunos componentes de back-end pre-programados.
- Soporta nativamente Websocket. Los controladores Websocket se definen de la misma manera que un controlador HTTP.
- Los controladores se definen utilizando clases, por lo cual es posible utilizar herencia.
- Este framework es reconocido por su buen rendimiento.

Desventajas:

- Por ser poco estructurado, las aplicaciones escritas con tornado pueden llegar a ser muy diferentes entre ellas.
- Tiene una mala documentación. Algunas veces es necesario recurrir al código fuente para entender ciertas funcionalidades.
- No tiene muchos componentes de front-end pre-programados.

Front-end

JQuery Mobile

Ventajas:

- Viene con una gran variedad de controles.
- Permite cambiar temas fácilmente.
- Es de fácil implementación. Usa clases CSS.
- Está altamente automatizado.

Desventajas:

- Está altamente automatizado.
- Cambia agresivamente el comportamiento estándar del sistema.
- Algunos componentes tienen un mal rendimiento en mi dispositivo móvil.

Bootstrap

Ventajas:

- Viene con una razonable variedad de controles.
- Es de fácil implementación. Usa clases CSS.
- El contenido se ajusta dinámicamente al tamaño de página.
- Viene con iconos.
- Es una implementación poco obstrusiva.

Desventajas:

- Agrega overhead a la carga de la página.

No utilizar un framework

Ventajas:

- No agregar overhead a la pagina.

Desventajas:

- Se incluyen solo los controles disponibles en los formularios.
- Hay que crear las hojas de estilo desde cero.
- Hay que procurar que el diseño sea adaptativo.
- Hay que procurar que el diseño sea compatible con los navegadores mas populares.

Características de comparación

Back-end

Característica	Ponderación
Capa de abstracción para bases de datos	10%
Posibilidad de estandarización de la aplicación	3%
Calidad del sistema de plantillas	11%
Calidad de componentes pre-programados	10%
Tiempo de aprendizaje (a menor tiempo mejor nota)	11%
Overhead (a menor overhead, mejor nota)	15%
Calidad del soporte para Websocket	30%
Documentación	4%
Herencia en los controladores	6%

Front-end

Característica	Ponderación
Variedad de controles	5%
Apariencia	10%
Dificultad de implementar	5%
Overhead	20%
Rendimiento	20%
Diseño adaptativo	20%
Compatibilidad	20%

Bibliografía

- [1] Resumen Proyecto de Titulación: “Aplicación móvil basada en WebSocket, para aulas SCALE-UP”, <http://goo.gl/gBNXX2>
- [2] Estado del Arte: “Aplicación móvil basada en WebSocket, para aulas SCALE-UP”, <http://goo.gl/Tul9nA>
- [3] Wikipedia: Comparison of web application frameworks, <http://goo.gl/MYpyYe>
- [4] Pagina oficial de Django, <https://www.djangoproject.com/>
- [5] Pagina oficial de Flask, <http://flask.pocoo.org/>
- [6] Pagina oficial de Tornado, <http://www.tornadoweb.org/en/stable/>
- [7] Pagina oficial de jQuery Mobile, <http://jquerymobile.com/>
- [8] Pagina oficial de Bootstrap, <http://getbootstrap.com/>