TECNOLÓGICO DE COSTA RICA

AplicaciónContactoMapa

Aplicaciones Web

Prof Andréi Fuentes

Julio Agüero Sánchez

13/11/2012

Contenido

Descripción del problema

Se tiene la necesidad de gestionar contactos con sus datos personales. específicamente referencias de: Nombre, Apellido, Correo Electrónico, Dirección, Empresa.

También, se debe poder mostrar las direcciones de cada contacto en un mapa geográfico. Para poder ver su dirección de manera inequívoca y precisa.

Este problema es algo que puede acaecer a personas que necesiten tener una agenda grande de contactos y además desean tenerlos en un mapa para poder ubicarlos rápidamente, ya sea ellos mismos o algún tercero que quiera saber sobre su posicionamiento global.

Como solución se propone la programación de una aplicación web en Django, framework para uso con el lenguaje de programación Python. En esta aplicación se cubrirán los requerimientos indicados por el cliente, brindando así una plataforma tecnológica para el uso en las actividades relacionadas con ubicación y manejo de contactos personales.

Diseño del programa

Decisiones de diseño

Como se mencionó anteriormente, se utiliza el framework Django, por lo que se sigue un patrón arquitectónico de Model View Controller. Esto principalmente para aprovechar la agilidad de desarrollo que brinda el framework. Además por la familiaridad con el patrón, lo cual permite un mejor desarrollo con mayor calidad.

También, será una aplicación web, por lo que se puede acceder con sólo poseer un browser, por lo que es *cross-platform*, una ventaja en la actualidad por la existencia de tantos sistemas operativos.

Para la autenticación de los usuarios se brindará la opción de realizarlo por medio de una cuenta de twitter haciendo uso de la biblioteca OAuth y las prestaciones del API de Twitter.

Se utilizará sqlite 3 como base de datos, ya que se trata de una versión en desarrollo y no se necesita de un motor con caracteristicas superiores por el momento, en próximas entregas y dependiendo de la aceptación de la aplicación, se podrá usar otro DBMS. Se usó Ubuntu 11.04 como sistema operativo para el desarrollo.

Para Django, como para muchos otros frameworks, existen herramientas para el desarrollo rápido como lo es la generación de código. Se decidió utilizar la biblioteca django-generate-scaffold para aprovechar las ventajas que ofrece y hacer un desarrollo más veloz para cumplir con la fecha de entrega al cliente.

Para lograr que los mapas de google fueran mostrados correcctamente, se recurrió al API que brinda google y sus funcionalidades de java-script.

Para esta versión en desarrollo, se mantuvo una interfaz de usuario que fuera sencilla, usable y que proveyera de lo necesario.

Diagrama de clases

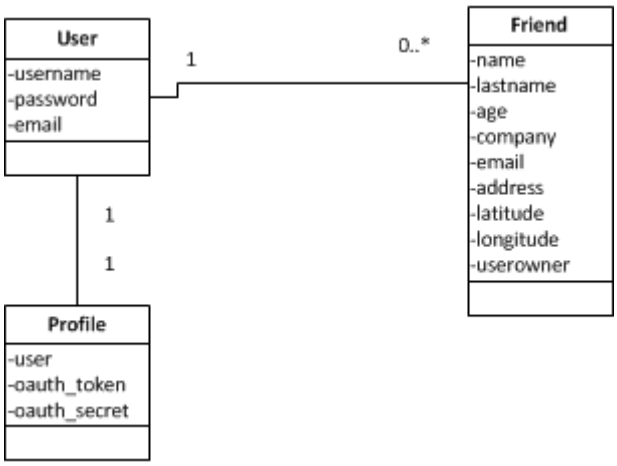
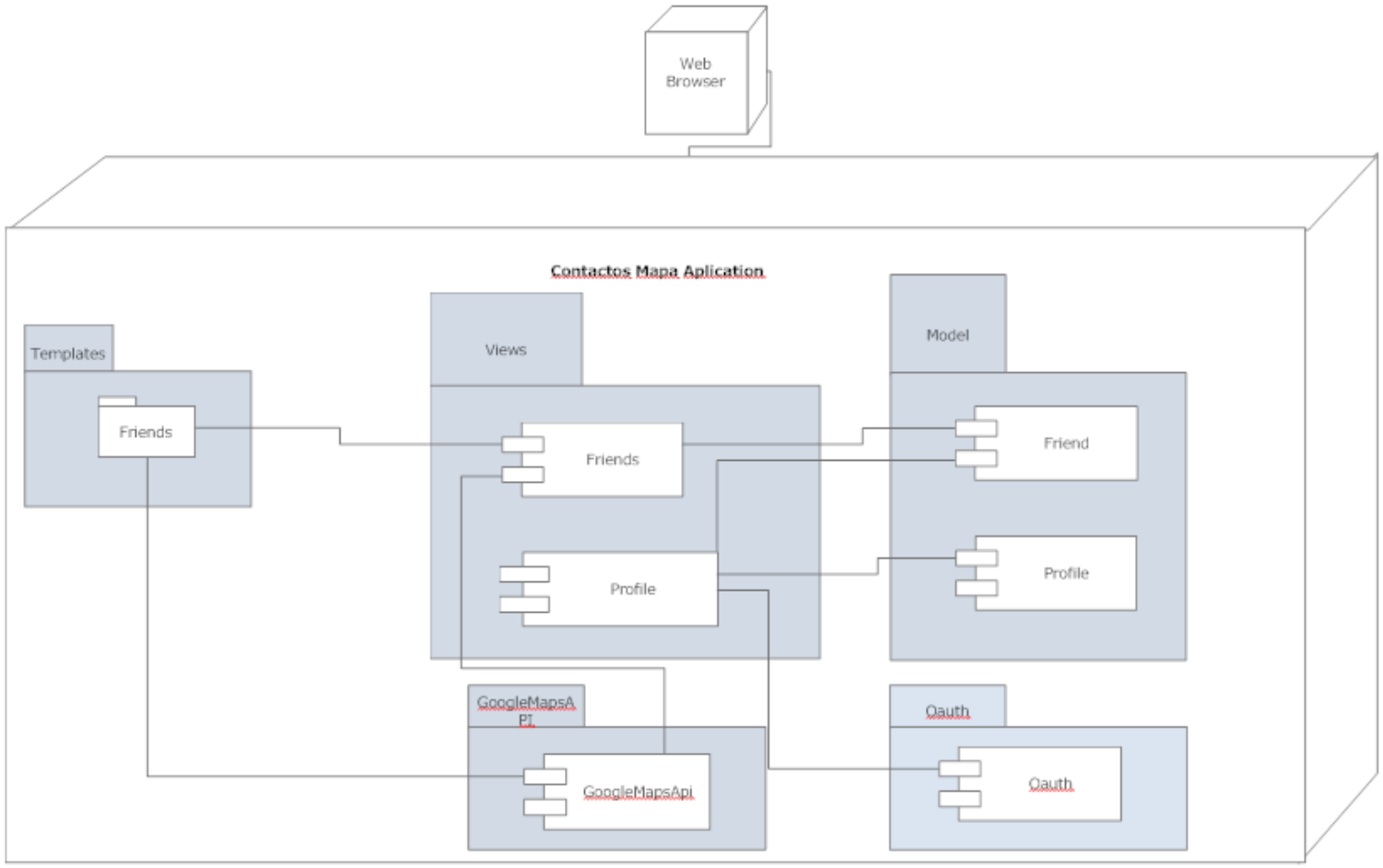


Diagrama de arquitectura

Se detallan los componentes principales y sus relaciones:



Uso de Django

En general es un Framework que ayuda al desarrollo de aplicaciones al elevar el nivel de abstracción para el usuario, haciendo que sea más sencilla la programación. Por ejemplo, abstrae el uso de la base de datos, la simplifica a códigos sencillos y fáciles de entender.

Aparte tiene una división característica en un proyecto que contiene aplicaciones.

Manejo de Views

En el caso de Django, aunque siga un patrón Model View Controller, las Views son los controladores, son los encargados de controlar el flujo de la aplicación y aplicar las funcionalidades correspondientes para lograr los objetivos de la aplicación.

Para esta aplicación se tienen los controladores de *Friends* que manejan las solicitudes sobre los objetos Friend (que representa los contactos personal). Para lo que son las acciones de creación, actualización, vista de contactos creados por fecha, entre otros.

A los controladores se les añadieron tanto métodos como también se modificó su funcionalidad existente para poder cumplir con los requerimientos del programa.

Manejo de Templates

En el caso de django, los templates serían lo que son las vistas en muchos otros frameworks.

Para el formato de vistas se utilizó html. Al utilizar un generador de texto se generaron estilos y un base_html por aplicación para mostrar una base de diseño en todas los templates de la aplicación.

También, en los templates se lleva a cabo la validación por parte del cliente de los datos ingresados en los distintos formularios de la aplicación.

Manejo de Models

Se crearon dos modelos: Friend y Profile. Estos fueron creados por medio de la herramienta scaffold.

Como se puede notar en el diagrama de clases, Usuario tiene campos nombre, apellido, email, empresa, dirección; Profile tiene a una instancia de User y los tokens para poder autenticarse por medio de Twitter.

Hay que considerar que se hizo uso de los modelos que tiene django para autenticación, como lo es el User de su aplicación de Auth.

Y también, las reglas para las validaciones por parte del servidor se deberían realizan en los modelos.

Manejo de URLs

Para poder tener acceso a un controlador -view en este caso- se tiene que agregar en el archivo de urls.py de cada aplicación y del proyecto. Se podría decir que los URLs se manejan a nivel de proyecto y luego a nivel de aplicación.

Para poder agregar una acción a un controlador se tiene que asegurar que esta esté incluida en el archivo de ruteo de la aplicación, de lo contrario no se podrá utilizar. Se debe escribir la ruta que viene después del nombre de la acción por medio de una expresión regular para comunicarle a django cuál debe ser la acción a usar con el request enviado.

Existen maneras de utilizar vistas genéricas en las cuales el archivo de urls.py viene a tomar un mayor papel para poder asignar valores a variables para enviarlas a los views

Archivos de configuración

Se tiene el archivo settings.py a nivel de proyecto. En este se cambia todo lo que tiene que ver con aplicaciones utilizadas, por ejemplo si se crea una aplicación se debe agregar a la variable INSTALLED_APPS. La configuración a la conexión a la base de datos, el middleware que se usa, entre todos los aspectos de configuración se establecen en este archivo.

Bibliotecas externas

Se trabajó con oauth2 -Open Authentication- como biblioteca para la autenticación con Twitter, ya que es la que Twitter acepta desde el 2010. Se instaló por medio de *pip*. Se decidió utilizar esta por la mayor sencillez y eficiencia que presentaba. Por medio de esta se interactúa con el API de Twitter y sus tokens de authenticación, autorización.

Es importante notar que ya no se envía el password de las cuentas de twitter hacia la aplicación. Todo se hace por medio de tokens codificados por Twitter, para poder identificar a la aplicación que quiere solicitar permiso de utilizar datos del usuario de Twitter y para poder identificar y autenticar al usuario también. Esto es una gran mejora en la seguridad, al no compartir información privada con terceros.

Por otra parte, se utilizó también el API que da Google a los desarrolladores. Específicamente el API V3 para google maps, en el cual se obtuvo también un token para autenticar y autorizar al desarrollador de la aplicación presente y que de esta manera se pudiera hacer uso de los mapas de google. Estos se manejan por medio de java-script, tienen varias funcionalidades y el API las muestra muy claramente. Es muy sencillo de integrar con las páginas web, al simplemente integrar un script para esa funcionalidad específica.

Análisis de resultado

Aspecto	Estado
Autenticación con Twitter[1]	Implementado con éxito
Gestión de Contactos	Implementado con éxito
Documentación externa	Implementado con éxito
Integración con API GoogleMaps[2]	Implementado en un 25%
Búsqueda avanzada[3]	No implementado
Validaciones de parte del cliente[4]	Implementado en un 50%
Documentación interna[5]	Implementado en un 50%
Documentación externa	Implementado con éxito

[1] La autenticación con Twitter dió muchos problemas por cuestión de conflicto de nombre de usuarios, el superusuario con el nombre de usuario de twitter.
[2] No se pudo implementar por cuestiones de tiempo y más sobre todo por problemas al utilizar un generador de código que encapsulaba estructuras importantes para poder acceder y establecer datos. Además de problemas con los estilos y su interacción con javascript
[3] No se implementó por cuestiones de tiempo.
[4] Ídem
[5] Se implementó en donde se consideró necesaria para el desarrollo.

Manual de usuario

Configuración

Instalar Python:

```
apt-get update; apt-get install python2.7
```

Bajar Django en la página web oficial, https://www.djangoproject.com/download/ y correr los siguientes comandos para instalarlo. Correr desde donde está el archivo bajado:

```
tar xzvf Django-1.4.2.tar.gz

cd Django-1.4.2

sudo python setup.py install
```

Instalación

Colocar la estructura de archivos del proyecto en algún lugar de su disco duro y estará lista la instalación.

Se debe realizar la instalación de algún web browser como lo es Google Chrome, Mozilla Firefox o Opera. Se recomienda que sea una versión actualizada.

Uso

Desde la carpeta del proyecto correr el comando:

```
python manage.py runserver
```

Seguidamente abrir una ventana del explorador y dirigirse a la dirección de dominio http://127.0.0.1/friend y a partir de ahí se puede navegar a través del menú diseñado para este fin.

Conclusiones

- Se debe tener en cuenta al cliente en todo momento al desarrollar una aplicación. Priorizar las funcionalidades que más le generen valor y desarrollarlas de primero. De esta manera, el cliente verá avance y se sentirá más confiado con la aplicación, por lo que si llega el momento de tener que pedir un aplazamiento de tiempo será mejor visto por el cliente.
- Establecer métricas de "estiramiento": Esto es, si acaso hay un problema que no prospera y se tiene dificultad para avanzar a pesar de consultar fuentes expertas, se debe avanzar con las funcionalidades de la aplicación. Sino, se cae en el riesgo de no poder cumplir con la fecha de entrega al cliente, lo que debilita la relación desarrollador-cliente y puede llevar a problemas en el futuro.
- Se debe tener máxima cautela con los generadores de código porque siempre van a introducir líneas de código innecesarias, encapsular funcionalidad necesaria de modificación y muchas veces confunden al programador al no mostrar claramente cómo realizan su función.
- La incursión en el uso de un nuevo framework, siempre representa una curva de aprendizaje exponencial. En especial, Django fue difícil de aprender para mí, mayormente por su manejo de vistas, enrutamiento y por además basarse en un lenguaje no conocido para mí, Python.
- Las aplicaciones web tienen una gran ventaja ya que no están ligadas a una plataforma específica.
- El uso de un framework permite tener estandarización hasta cierto punto, por las convenciones sobre nombres, la estructura de la aplicación y hasta la estructura de las carpetas en las que se contiene el código fuente.
- El uso de un sistema de control de versiones distribuido es de gran ayuda para tener un seguimiento de las acciones realizadas en la aplicación. Inclusive si se realizó un error, se puede regresar a una versión funcional sin mayor problema.
- Hay que tener claro que trabajar con APIs externos siempre traerá dificultades. Se debe realizar un procedimiento de mucha investigación y lamentablemente al tener documentación pobre, se debe recurrir a la prueba y el error.