

geoAlertas.es

Tus alertas según tu posición



geoAlertas

Tus Alertas según tu Posición

geoAlertas.....	2
Autor.....	3
Links.....	3
Descargas.....	3
Licencia.....	3
Intenciones.....	3
Ejemplos.....	3
Los módulos.....	3
Módulo web.....	4
Módulo script/cron.....	7
El algoritmo de colisión.....	8
API de Localizame.....	8
API Interfaz HTTP de envío de SMS.....	8
API de GoogleMaps.....	9
Herramientas Open Source.....	9
En el futuro.....	9

Autor

Fernando Guillén Suarez (<http://fernandoguillen.info>)

Links

- <http://geoalertas.blogspot.com/>
- <http://geoalertas.es/>
- <http://open.movilforum.com/node/357>

Descargas

- <http://code.google.com/p/geoalertas/downloads/list>

Licencia

Apache 2.0

Intenciones

geoAlertas intenta ser un sencillo sistema de gestión de **recordatorios** según tu **localización**.

Podrás dar de alta tantas alertas como desees asignándoles una posición geográfica, el sistema intentará seguir tu posición y enviarte las alertas según te acerques a los puntos calientes.

Existen muchas herramientas para gestionar alertas con una fecha y horario determinado, pero no existe ninguna que gestione las alertas según tu posición actual.

Esta posibilidad es increíblemente útil porque muchas veces necesitamos recordar algo no en un *determinado momento* sino en un *determinado lugar*.

Ejemplos

Ahora que pasas cerca de la panadería, recuerda comprar una hogaza.

Estás cerca de casa de Juan, ¿por qué no le haces una visita?.

La librería está cerca, compra la cartulina negra que siempre se te olvida

Los módulos

La aplicación está compuesta por dos módulos completamente diferenciados que comparten sólo el modelo y la base de datos.

Por un lado tenemos el **módulo web** que proporciona una interface amigable para que los usuarios puedan gestionar sus cuentas y sus alertas y por otro lado tenemos el

módulo script-cron que se encargará de ir solicitando la posición actual de cada usuario y cotejarla con las alertas que tiene activas para ver si tiene que enviarle alguna de ellas.

Módulo web

Se basa en una plataforma web basada en RubyOnRails con todas las funciones necesarias para que los usuarios puedan gestionar sus cuentas y sus alertas.

En este módulo se ha puesto un gran esfuerzo en la sencillez, en la reducción de las funcionalidades a las mínimas más útiles, siguiendo los populares patrones 'KISS' (http://es.wikipedia.org/wiki/Principio_KISS) y 'menos es más' (<http://www.lacasagiratoria.com/archivos/2005/08/26/la-filosofia-de-menos-es-mas.php>).

Gestión de cuentas

geoAlertas.es
Tus alertas según tu posición

usuario 669477485asd tus alertas nueva alerta localizaciones tu cuenta salir

actualizar cuenta

Aquí puedes configurar los datos de tu cuenta. Recuerda que antes de poder acceder al servicio geoAlertas debes de estar dado de alta en los servicios movistar: [localizame](#) y [SMS](#). Lee las [instrucciones](#) para más información.

número de tu móvil
600600600

tu clave localizame
...

tu clave SMS
....

email
asd

clave para esta aplicación
....

actualizar cuenta

wiki blog autor código

El usuario puede modificar sus datos, necesitamos varias claves para poder acceder a sus cuentas en las APIs de Movistar.

Gestión de alertas

Crear/Modificar alertas

geoAlertas.es
Tus alertas según tu posición

usuario 669477485asd tus alertas nueva alerta localizaciones tu cuenta salir

nueva alerta

Aquí puedes dar de alta tu nueva alerta. Simplemente indica la dirección, los metros de distancia a los que quieres que el sistema te avise y la nota que quieres que se te envíe.

dirección
Plaza de España, Madrid

radio
100 metros

nota
Estás cerca de casa de Juan, ¿por qué no le hacemos una visita?

interruptor
☒ activada ☐ desactivada

enviada
No enviada ninguna vez todavía.

crear alerta


wiki blog autor código

Para definir una alerta el usuario debe indicar una dirección y un **radio**, esto modificará el algoritmo de colisión utilizado para saber si su posición actual coincide con la dirección de esta alerta.

En el campo nota se indica el contenido del **SMS** que será enviado al usuario cuando esta alerta se dispare.

Una alerta puede estar **activada** o **desactivada** según se desee.

Listado de alertas



Tus alertas según tu posición

usuario 600600600 tus alertas nueva alerta localizaciones tu cuenta salir

ver alertas

Acceso correcto, ya puedes gestionar tus alertas

Aquí puedes ver todas las alertas que has creado ordenadas por la actualizada más recientemente. Las alertas se mostrarán tanto si están activas como si no.

Nicolás Morales 34, 28019, Madrid (100m)



Estás cerca de la panadería, acuérdate de comprar una hogaza

Última actualización el Mon May 12 20:03:34 +0200 2008
Último envío el Mon May 12 20:03:35 +0200 2008
Esta alerta está **activa**

editar

Plaza de España, Madrid (100m)




Estás cerca de casa de Juan, ¿por qué no le hacemos una visita?

Última actualización el Mon May 12 20:03:34 +0200 2008
Último envío el Mon May 12 20:03:35 +0200 2008
Esta alerta está **activa**

editar

plaza antón martin, Madrid (100m)



Mira a ver que ponen en el cine ya que estás ahí

Última actualización el Mon May 12 20:03:34 +0200 2008
Ningún envío todavía
Esta alerta está **no activa**

editar

[wiki](#) [blog](#) [autor](#) [código](#)

En la pantalla de listar alertas el usuario puede ver todas sus alertas, tanto las activas como las que no.

En los detalles se muestra la dirección, la geolocalización de esta alerta incrustada en un **GoogleMaps**, el texto de la alerta, la fecha de última actualización, la fecha del

último envío que ese hizo y el estado actual.

Desde este listado podemos acceder a modificar cualquiera de las alertas.

Localizaciones



Esta pantalla es meramente informativa, aquí el usuario puede ver el histórico de localizaciones que el sistema ha hecho de su teléfono.

Módulo script/cron

Este módulo se ha programado independiente del módulo web pero comparten el modelo y la base de datos.

Se trata de una colección de clases Ruby que se pueden ejecutar desde un cron de sistema y se encargan de recopilar la posición de todos los usuarios de la aplicación y cotejarla con las direcciones y radios de sus alertas, en el caso de que se detecte una colisión se enviará un SMS.

La clase core

AlertsCron que se encuentra en `/lib/alerts_cron.rb`

Prueba de ejecución

`AlertsCron.new.run`

La configuración

Lo suyo es que se configure en el servidor un cron que haga la llamada oportuna a la clase cada intervalo de tiempo que creamos oportuno.

El algoritmo de colisión

No es fácil calcular la distancia entre dos coordenadas cartesianas.

Se ha conseguido gracias a un fork en ruby desarrollado explícitamente para esta aplicación de la función aquí descrita: <http://snipplr.com/view/2531/calculate-the-distance-between-two-coordinates-latitude-longitude>

```
puts( '-----' )
puts( 'distancia entre nicolas morales 38 y plaza de españa' )
puts( '-----' )
puts( GeoAlertsUtil.distance( 40.390754, -3.732509, 40.423549, -3.713337 ) )

puts( '-----' )
puts( 'colisión de 100metros Entre nicolas Morales 38 y 40' )
puts( '-----' )
puts( GeoAlertsUtil.collission( 40.390754, -3.732509, 40.390757, -3.732715, 100 ) )
```

API de Localizame

La API de **Localizame** es un poco dura de roer. Al final se ha conseguido hacer la llamada a la API y recuperar la localización en lenguaje humano que esta API retorna.

```
puts( "-----" )
puts( "accediendo a API" )
puts( "-----" )
location = MovistarLocalizame.get_location( '600600600', 'xxxx' )
puts( "LOCATION:" + location )
```

Se ha utilizado ruby, Hpricot y un poco de ingenio .

Al final como lo retornado por este API es demasiado genérico se ha encontrado una coincidencia en todos los tipos de respuesta y es el **Código Postal**.

Así que por ahora nos conformamos con el Código Postal para automatizar la localización del usuario.

```
cp = MovistarLocalizame.parse_cp( location )
puts( "CP:" + cp )
```

Recordar que para que la aplicación pueda localizarte el usuario debe enviar **un SMS al 424 con la palabra 'clave' cada 30 minutos**. Esto es un requisito de la propia API.

API Interfaz HTTP de envío de SMS

Esta API ha sido muy fácil de usar pues ya teníamos la implementación de la llamada escrita en Ruby.

Lástima que a día de hoy no parezca que funcione bien pues el servidor responde con

'Ok' pero el mensaje nunca llega.

En cualquier caso la aplicación está preparada para que cuando este API vuelva a reaccionar funcione sin ningún cambio.

API de GoogleMaps

Nuestra estimada amiga API de GoogleMaps nos ayuda tanto como para mostrar las direcciones de las alertas y de las localizaciones de usuario en un mapa interactivo como para convertir las direcciones insertadas por el usuario al crear/modificar alertas en coordenadas geográficas.

Para convertir una dirección en una coordenada cartesiana se usa la gema google-geocode:

```
require 'rubygems'
require 'google_geocode'

gg = GoogleGeocode.new GOOGLE_API

address = 'nicolas morales 38, madrid'
location = gg.locate address
puts 'address:' + address
puts 'address:' + location.address
puts 'lat:' + location.latitude.to_s
puts 'lng:' + location.longitude.to_s
```

Herramientas Open Source

Todas las herramientas tanto de desarrollo, como servidores, como Sistemas Operativos, como herramientas de diseño gráfico que se han utilizado son **Open Source**.

En el futuro

Mejoras en la interface web

Validaciones de formulario

Mejoras en la interpretación de la localización del servicio Localizame

Listar localizaciones según cronología

Mejorar mapas GoogleMaps: iconos personalizados, globos con información en iconos.