

**Pliego técnico para la contratación del servicio de asistencia técnica para el desarrollo de un asistente de movilidad virtual para el proyecto de MetroGuagua de Guaguas Municipales, S.A.**

**1. Objeto del contrato**

Asistente virtual de movilidad conectado a un sistema de información que procese distintas fuentes de datos y las comunique a través de distintos canales de forma programada.

**2. Desarrollo y fases del proyecto**

El proyecto se dividirá en dos fases dentro de las cuales se desarrollarán y evolucionará tanto los puntos de contacto del asistente (web y comunicaciones) como el propio sistema de información:

Fase 1 - Objetivos:

- a. Establecer la base del sistema de información y asistente
- b. Cubrir el las obras con un punto de información accesible y online
- c. Desplegar un canal de alertas básico y segmentado sobre movilidad
- d. Ofrecer los canales de suscripción de alertas e información: SMS y eMail
- e. Comenzar a analizar comportamiento de usuarios

Fase 2 - Objetivos:

- a. Procesar fuentes de información dinámicas y ofrecerlas al usuario
- b. Evolucionar la comunicación con usuarios a través de comunicaciones más avanzadas
- c. Exponer la información del asistente a través de una API pública para terceros
- d. Actualizar el sistema de alertas en tiempo real en base a las fuentes dinámicas
- e. Segmentación de usuario basada en información de perfil y puntos de interés
- f. Permitir consultas guiadas básicas al asistente

**3. Diseño**

El objetivo es diseñar cada uno de los elementos que se van a implementar dentro del punto de contacto de la web así como de las comunicaciones que se lanzan al usuario. Los elementos a diseñar son:

- A. Diseño de la página de perfil de usuario donde podrá realizar las siguientes tareas:
  - a. Modificación de datos personales
  - b. Gestión de canales de alertas
  - c. Gestión de puntos de interés dentro de la ciudad
  - d. Baja del sistema
  - e. Ver alertas
- B. Diseño de los correos electrónicos:
  - a. Notificaciones y alertas
  - b. Verificación de cuenta
  - c. Baja del sistema
  - d. Actualización de datos
- C. Textos de SMS:

- a. Notificaciones y alertas
- b. Baja del sistema
- c. Actualización de datos
- D. Diseño de cada una de las páginas nuevas de la web y sus textos:
  - a. Acerca del asistente
  - b. Preguntas Frecuentes
  - c. Integración de banners
  - d. Documentación de la API
- E. Diseño de banners: Se diseñarán cada uno de los banners que se podrán integrar en webs de terceros, los tamaños mínimos a diseñar son:
  - a. 300x250
  - b. 728x90
  - c. 160x600
  - d. 250x250
- F. Rediseño de la página de mapa de movilidad para incluir las siguientes funcionalidades:
  - a. Alertas geopositionadas
  - b. Puntos de interés del usuario
  - c. Área de acción del usuario
  - d. Nuevas obras de MetroGuagua
- G. *Diseño de feed de Alertas*
- H. *Diseño de aplicaciones de asistente para nuevas páginas y estados web*

#### 4. Web Pública

El objetivo es evolucionar el punto de entrada a la web pública del asistente para los ciudadanos, dicha web conllevará las siguientes tareas y funcionalidades:

1. Desarrollo HTML y CSS de las páginas nuevas:
  - a. Acerca del asistente
  - b. Preguntas frecuentes
  - c. Integración banners
  - d. Nuevos filtros en el mapa de movilidad
  - e. Página de perfil
  - f. Imágenes de Asistente
  - g. Feed de alertas a modo Chat
2. Desarrollo JavaScript:
  - a. Implementación de nuevas URLs de secciones
  - b. Parametrización de templates
  - c. Extensión del uso de la API de Google Maps para mostrar alertas e información sobre mapas
  - d. Implementación de errores globales y específicos de servicios
  - e. Autenticación OAuth
  - f. Log-in
  - g. Integración de nuevos servicios internos de API para la gestión de alertas, información de usuario y mapas
  - h. Sistema de actualización de metas
  - i. Implementación de objetos Schema
  - j. Sistema de analíticas por eventos y vistas de página
  - k. Caché de nuevas templates de páginas



- l. Implementación de tareas en Grunt
- m. Implementación funcionalidad de compartir mapa y contenido de la web
- n. Integraciones de servicios API:
  - i. Alertas
  - ii. Contenido
  - iii. Usuarios
  - iv. Información en base a geoposición
- o. Sistema de cookies de usuario: Usuarios recurrente
- p. Sistema de A/B Testing para la mejora de la experiencia de usuario+
- q. Mapa de movilidad
  - i. Gestión de capas de información
    - 1. Guaguas
    - 2. Sagulpa
    - 3. Policía
    - 4. Eventos
    - 5. Obras
    - 6. Alertas
  - ii. Funcionalidad para poder compartir capas de información
  - iii.
- r. Multi-idioma
- s. Animaciones y micro-interacciones

## 5. Back-end

El objetivo sería crear toda la capa de back-end y back-office con los siguientes modelos, funcionalidades y configuraciones:

- 1. Datos:
  - a. Modelos y serializadores para fuentes de informaciones externas
  - b. Modelos y serializadores para contenido de cortes por eventos
  - c. Modelos y serializadores para contenido de obras
  - d. Modelos y serializadores para alertas
  - e. Modelos y serializadores para analíticas
  - f. Modelos y serializadores de perfiles de usuario
  - g. Modelos y serializadores de contenido geoposicionado
- 2. Fuentes de información:
  - a. Integración, procesado y normalización Twitter de la Policía
  - b. Integración, procesado y normalización de API Guaguas
  - c. Integración, procesado y normalización de API Sagulpa
  - d. Integración, procesado y normalización de desvíos de líneas Guaguas por eventos
- 3. Implementación vistas panel de administración:
  - a. Gestión de contenido geoposicionado de obras
  - b. Gestión de contenido geoposicionado de eventos
- 4. Envío de alertas y comunicaciones:
  - a. Alertas y comunicaciones segmentadas por posición
  - b. Alertas y comunicaciones segmentadas por recurrencia
  - c. Alertas y comunicaciones segmentadas por horas
  - d. Post a redes sociales automático
  - e. Categorización de alertas

- f. Integración de nuevas plantillas de correo y SMS con los servicios
- 5. API:
  - a. Servicios internos:
    - i. Banners
    - ii. Web
  - b. Servicio para terceros
    - i. Petición de todas las alertas
    - ii. Petición de alertas e información por código postal
    - iii. Petición de alertas e información por latitud y longitud
    - iv. Petición de alertas e información por dirección

- 6. Segmentación de usuarios
  - a. Procesado de perfiles de usuario
  - b. Procesado de preferencias de usuario
  - c. Procesado de puntos de interés del usuario
  - d. Conexión a base de datos de analíticas

## 6. Canales

Suponen los canales de comunicación a los que el usuario se puede suscribir y a través de los cuales recibirá la información procesada por el asistente:

- 1. Email
  - a. Integración de cada una de las plantillas
  - b. Parametrización de contenido por usuario
- 2. SMS
  - a. Integración de cada una de las plantillas de envío
  - b. Parametrización de contenido por usuario

## 7. Servidores y entornos

Crear, configurar y mantener los entornos donde se van a desplegar cada una de las versiones del asistente a nivel back y front-end

- 1. Creación de entornos
  - a. Creación del entorno de test
  - b. Creación del entorno de pre-producción
  - c. Creación del entorno de producción
- 2. Configuración de servidores
  - a. Configuración de sistema operativo
  - b. Configuración de claves de acceso a los servidores
  - c. Proxys de seguridad y firewall para bloqueos de acceso
  - d. Instalación de agentes para el control de métricas
- 3. Despliegue continuo
  - a. Configuración de proceso de despliegue escalonado
  - b. Configuración del sistema de failover para permitir rollbacks en despliegue fallidos
  - c. Preparación y migración de certificados de servidores hacia Jenkins

- d. Scripts de despliegue distribuido entre entornos
- e. Scripts de test automatizados
- f. Métricas de cobertura de código
4. Servidor de logs centralizados
  - a. Instalación de agentes para centralización de logs
  - b. Configuración de alertas en logs
  - c. Configuración del dashboard de alertas sobre los logs de los servidores
5. Configuración de métricas
  - a. Preparación de alarmas en CloudWatch para aviso de incidencias o mal funcionamiento de los sistemas
  - b. Configuración de alarmas de escasez de memoria en discos y posterior auto-resize
6. Configuración de certificados y firmas
  - a. Instalación de los certificados en los servidores
  - b. Configuración del sistema de distribución automática entre servidores de los certificados

#### 8. Otras consideraciones técnicas

- Django 1.9 o superior con dashboard de administración
- Django Rest Framework
- GDAL libraries (Geospatial Data Abstraction Library)
- Celery
- Servicio de envío SMS basado en coste por envío
- Servicio de envío Email basado en coste por envío
- Redis Caché
- S3 Storage
- Database (AWS RDS): PostgreSQL + PostGIS extension
- Autenticación OAuth2
- Integración continua y entornos (desarrollo, stage, producción):
  - Docker
  - Jenkins
  - Load Balancers
  - Route53
  - AWS Elastic Beanstalk
  - AWS EC2
  - AWS Lambdas
  - AWS Step Functions
  - AWS CloudWatch
- Sistema centralizado de logs (Logentries)
- Front-end desarrollado en framework JS Angular 1.4
- HTML5 y CSS 3.0
- Node.js y Grunt.js
- Documentación API basada Swagger

**Las Palmas de Gran Canaria a 6 de noviembre de 2017**

**EI GERENTE**

**Miguel A. Rodríguez Ramírez**



