

Universidad Autónoma Metropolitana Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación  
Modalidad: Experiencia Profesional

Desarrollador Senior

Roberto Emmanuel Ramírez Islas  
205204716

Jefe directo  
M. en C.A. Jesús Miguel de la Rosa Estañol  
Director general y fundador  
Aplika Soluciones Precisas

Trimestre 2014 Primavera  
14 de marzo de 2014

Yo, M. en C.A. Jesús Miguel de la Rosa Estañol, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Roberto Emmanuel Ramírez Islas, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco. que deberá entregar al Coordinador de Estudios a más tardar el día programado para la evaluación global o de recuperación, según sea el caso, para su firma y sello.

## **Resumen ejecutivo**

El presente documento describe la posición que a la fecha ejerzo en la empresa Aplika Soluciones Precisas. En el se encuentran una descripción de la empresa y las actividades asociadas a mi puesto, así como una relación de los proyectos en los que he estado envuelto. En estos últimos se enuncian las tecnologías utilizadas, mi responsabilidad y una descripción técnica de lo que logré durante mi participación en ellos. Por último se profundiza en el proyecto con nombre código Mahís.

# Índice

<b>1. Descripción de la empresa y actividades a las que se dedica</b>	<b>5</b>
1.1. Metodología de trabajo . . . . .	5
<b>2. Departamento de adscripción</b>	<b>5</b>
<b>3. Descripción técnica de las actividades asociadas al puesto</b>	<b>5</b>
<b>4. Relación de proyectos</b>	<b>6</b>
4.1. Nombre código: Azul (liberado) . . . . .	6
4.2. Nombre código: Pez (liberado) . . . . .	6
4.3. Nombre código: Frankie (no liberado) . . . . .	7
4.4. Nombre código: Manami (liberado) (aplicación para terceros) .	7
4.5. Nombre código: Muéstrate (liberado) . . . . .	8
4.6. Nombre código: Mahís (liberado) . . . . .	8
4.7. Proyecto para la Secretaría de Energía (no liberado) (aplica- ción para terceros) . . . . .	9
4.8. Reportetelefonico.com (liberado) (fallecido) . . . . .	9
4.9. Nombre código: Squall (no liberado) . . . . .	10
<b>5. Proyecto a profundidad: Nombre código: Mahís</b>	<b>10</b>
5.1. Análisis de requisitos . . . . .	11
5.2. Especificación . . . . .	11
5.3. Arquitectura . . . . .	11
5.4. Programación . . . . .	12
5.5. Pruebas . . . . .	12
5.6. Mantenimiento . . . . .	13
<b>6. Nota</b>	<b>13</b>

# 1. Descripción de la empresa y actividades a las que se dedica

Aplika Soluciones Precisas es una compañía mexicana enfocada al desarrollo de aplicaciones móviles e Internet. Con raíces desde el año 1998 y fundada en el año 2005, ha crecido a un ritmo sostenido y ha desarrollado continuamente soluciones tecnológicas en una variedad de campos. El éxito de la compañía se debe a una visión clara y a un ritmo adecuado de crecimiento.

<http://www.aplika.mx/>

## 1.1. Metodología de trabajo

Utilizamos principalmente la metodología de desarrollo ágil, y para ello nos apoyamos en varias aplicaciones de Atlassian (JIRA, Confluence, Stash, Fisheye/Crucible, Bamboo y Crowd).

# 2. Departamento de adscripción

Departamento de desarrollo de soluciones.

# 3. Descripción técnica de las actividades asociadas al puesto

A continuación enlisto las funciones de mi puesto.

- Análisis de requisitos. Habilidad para reconocer, especificar, analizar y capturar requisitos, así como plasmarlos en diagramas.
- Especificación. Interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software mediante caso de uso e historias de usuario.
- Arquitectura. Diseño de componentes de aplicaciones mediante diagramas de clases, diagramas de base de datos, diagrama de despliegue, diagrama de secuencia y herramientas CASE.
- Programación. Desarrollo en lenguajes de programación, tales como PHP y Java, y definición de unidades de prueba.

- Prueba. Comprobar que el software realice las cosas tal y como están descritas con procesos tan claros que cualquiera puede entenderlos.
- Mantenimiento. Mantener y mejorar el software para corregir errores descubiertos e incorporar nueva funcionalidad.

## 4. Relación de proyectos

### 4.1. Nombre código: Azul (liberado)

Sistema para mercadeo de proximidad mediante Bluetooth.

**Tecnologías** Bluetooth, Java, MySQL, SQLite.

**Responsabilidad** Desarrollador en jefe (aplicación heredada).

#### Descripción técnica del trabajo realizado

- Mantenimiento. Creación de un sistema de bitácoras de eventos de la aplicación, así como la reestructuración del código fuente para poder ser más legible.

### 4.2. Nombre código: Pez (liberado)

Sistema de mensajería SMS mediante interfaz web.

**Tecnologías** PHP, MySQL.

**Responsabilidad** Desarrollador en jefe.

#### Descripción técnica del trabajo realizado

- Análisis de requisitos. Juntas en las cuales se discutieron requisitos. Estos últimos han sido percibidos y propuestos (technology push).
- Especificación. Clasificación y especificación de los requisitos anteriores mediante casos de uso.
- Arquitectura. Diseño de la base de datos, así como del flujo general del sistema. Diseño de los protocolos de intercomunicación entre la aplicación y el servidor de envío de mensajes, así como la metodología de gestión de colas.

- Programación. Desarrollo total desde cero.
- Prueba.
- Mantenimiento. Corrección de errores.

### **4.3. Nombre código: Frankie (no liberado)**

Sistema web de gestión de servicios de paquetería.

**Tecnologías** PHP, MySQL.

**Responsabilidad** Desarrollador del módulo de servicio web.

#### **Descripción técnica del trabajo realizado**

- Arquitectura. Diseño de la especificación del servicio web.
- Programación. Desarrollo total desde cero del módulo de servicio web.
- Prueba.
- Mantenimiento. Corrección de errores.

### **4.4. Nombre código: Manami (liberado) (aplicación para terceros)**

Analizador (parser) y exportador de cuestionarios escolares.

**Tecnologías** Java.

**Responsabilidad** Desarrollador.

#### **Descripción técnica del trabajo realizado**

- Programación. Programación total desde cero.
- Prueba.
- Mantenimiento. Corrección de errores.

#### 4.5. Nombre código: Muéstrate (liberado)

Interprete modular para generación de aplicaciones móviles

**Tecnologías** JavaME.

**Responsabilidad** Desarrollador en jefe.

##### Descripción técnica del trabajo realizado

- Análisis de requisitos. Juntas en las cuales se discutieron requisitos. Estos últimos han sido percibidos y propuestos (technology push).
- Especificación. Clasificación y especificación de los requisitos anteriores mediante casos de uso.
- Arquitectura. Diseño del flujo general del sistema, especificación del motor gráfico, especificación de la estructura de los archivos de interpretación y arquitectura de módulos enchufables (plugins).
- Programación. Programación total desde cero.
- Prueba. Pruebas en conjunto con jefe directo.
- Mantenimiento. Corrección de errores y creación de nuevos módulos enchufables (plugins).

#### 4.6. Nombre código: Mahís (liberado)

Sistema de gestión comercial con frentes web y móvil conectados en modelo cliente-servidor con soporte fuera de línea.

**Tecnologías** PHP, PostgreSQL, Android, SQLite.

**Responsabilidad** Desarrollador en jefe.

##### Descripción técnica del trabajo realizado

- Análisis de requisitos. Juntas en las cuales se discutieron requisitos. Estos últimos son tanto percibidos y propuestos (technology push), como obtenidos directamente de los clientes actuales del sistema (market pull).

- Especificación. Clasificación y especificación de los requisitos anteriores mediante casos de uso.
- Arquitectura. Diseño de la base de datos general (servidor), así como la del cache local de los clientes, incluyendo la especificación de protocolos de sincronización entre ambas. Diseño del flujo general de ambos frentes, así como el de su interacción (cliente-servidor), la cual incluye el diseño de servicios web.
- Programación. Programación total desde cero.
- Prueba.
- Mantenimiento. Corrección de errores y nueva funcionalidad en base a una estrategia Push-Pull.

#### **4.7. Proyecto para la Secretaría de Energía (no liberado) (aplicación para terceros)**

Directorio nacional de energía solar con módulos de registro y exportación de datos.

**Tecnologías** PHP, PostgreSQL.

**Responsabilidad** Arquitecto del sistema.

##### **Descripción técnica del trabajo realizado**

- Arquitectura. Diseño de la base de datos, así como del flujo general del sistema.
- Programación. Programación total desde cero.
- Prueba. Pruebas en conjunto con jefe directo.
- Mantenimiento. Corrección de errores.

#### **4.8. Reportetelefonico.com (liberado) (fallecido)**

Sistema de reporte y análisis de consumo de servicios de telefonía celular.

**Tecnologías** PHP, SQLite.

**Responsabilidad** Arquitecto del sistema.

#### **Descripción técnica del trabajo realizado**

- Arquitectura. Diseño de base de datos utilizada como cache temporal, así como del flujo general del sistema.
- Programación. Programación total desde cero.
- Prueba.
- Mantenimiento. Corrección de errores.

### **4.9. Nombre código: Squall (no liberado)**

Aplicación gráfica de mensajería MMS.

**Tecnologías** JavaME.

**Responsabilidad** Arquitecto del sistema

#### **Descripción técnica del trabajo realizado**

- Arquitectura. Diseño del flujo general del sistema y especificación del motor gráfico.
- Programación. Programación total desde cero.
- Prueba.
- Mantenimiento. Corrección de errores.

## **5. Proyecto a profundidad: Nombre código: Mahís**

El proyecto Mahís es una solución que permite tomar decisiones comerciales de forma instantánea. Ideal para gestionar equipos en campo y administración de representantes médicos. Durante sus inicios este fue pensado como una aplicación genérica de recepción de datos y generación de reportes en base a estos, pero fue evolucionando en un producto enfocado a la recolección de información empresarial, especialmente acotado al momento de la escritura de este documento a remisiones y monitoreo de productos.

Los puntos fuertes de el son su robustes en su parte móvil, y sus reportes detallados en el frente web.

## 5.1. Análisis de requisitos

En un principio el equipo de trabajo fue reunido para discutir las necesidades percibidas y propuestas, concepto también conocido como *technology push*, o “a donde queremos llegar”. Durante estas juntas los participantes decimos lo que pensamos que debe de hacer el producto, limitamos su funcionalidad en base a tiempos y, con la ayuda de un pizarrón, así como con la ayuda de notas propias, esbozamos y diagramamos todo lo discutido.

Un punto muy importante de estas juntas fue la idea de que la información debería de poder ser capturada mediante un dispositivo móvil, y recibida en un servidor. El primero debería de ser lo más robusto posible para poder dejar hacer la captura sin contratiempos. A nadie le gusta dejar de trabajar si no hay conexión, y mucho menos perder datos.

Como parte del ciclo de la aplicación estas juntas no cesan, y después de la liberación de la primera versión ahora contiene retroalimentación de los clientes actuales del sistema, llamada también *market pull* o “a donde nos llevan”. Debido a esto último la aplicación fue tomando tintes más bien empresariales, llegando a su forma actual y dejando atrás se generalidad.

## 5.2. Especificación

En base a las notas del análisis empecé a trabajar cada punto, detallando y puliendo las ideas. Esto apoyándome en diagramas de caso de uso para determinar más a fondo cual iba a ser la interacción del usuario con los dos frentes del sistema. Después la labor fue dividir el trabajo en épicos, historias y tareas. Un épico es una idea grande que engloba varias historias, las cuales a su vez son requisitos técnicos del sistema y que contienen varias tareas. Esta última disección se debe a que en Aplika Soluciones Precisas trabajamos mediante la metodología ágil, y separando el trabajo en varios puntos nos da la oportunidad de poder liberar frecuentemente y corregir a tiempo.

Para esta labor me apoyé principalmente en un programa para diagramación llamado *Dia*, así como en *JIRA*, un programa para gestión de proyectos por parte de la compañía australiana *Atlassian*.

## 5.3. Arquitectura

El siguiente paso fue crear la base de datos. Este es un punto clave, pues el objetivo principal del sistema, como en la gran mayoría de todos,

es almacenar datos y mostrar reportes, obteniendo así información. Para su concepción me apoye de nuevo en *Dia*, y debido a la premura solo la lleve hasta su forma normal Boyce–Codd, dejando para después las siguientes.

Para el flujo del sistema seguí un patrón Modelo-Vista-Controlador, el cual permite un diseño claro y eficiente, así como un crecimiento sencillo. Dentro del modelo he plasmado principalmente la gestión de datos y reglas de negocio, dejando en las vistas y controladores una carga muy liviana. Es en el primero también en el que a la hora de programar he escrito todas las consultas a la base de datos.

Finalmente, diseñé los protocolos de comunicación y sincronización entre la parte móvil y el frente web. A estos les presté mucha atención, pues la robustez del sistema es fundamental. Para esto último diseñé un cache en la parte móvil que permite al usuario trabajar sin conexión, el cuál se sincroniza en cuanto recupera la conectividad.

## 5.4. Programación

Una vez terminado el diseño proseguí a escribir en código fuente el sistema. Esto en los lenguajes de programación *Java para Android* y *PHP*, con la ayuda de *Eclipse* para el primero y *Netbeans* para el segundo. La respuesta al por qué dos entornos de desarrollo es amor por el último y resignación por el ulterior.

Otra herramienta que me ayudo mucho fue *pgAdmin* para la creación de las consultas *SQL*. Hay algunas demasiado complejas debido a la normalización de la base de datos, a las cuales es difícil seguirles la pista sin coloración. De igual manera su visualizador gráfico del planeador de consultas ayuda mucho a la hora de optimizarlas.

Por último, utilizamos *Git* para versionar nuestro código fuente, el cual almacenamos en *Stash* de *Atlassian*.

## 5.5. Pruebas

Durante el proceso de programación hago pruebas estáticas de caja blanca, y durante juntas con el equipo se llevan a cabo pruebas de caja negra. En estas últimas nos juntamos varios elementos de la empresa con diferentes perfiles con el simple objetivo de romper la aplicación. En caso de encontrar problemas me regreso un punto atrás, y realizo las correcciones y mejoras pertinentes.

## **5.6. Mantenimiento**

En este punto me encargo de corregir los errores en el sistema que se nos escaparon en la etapa de pruebas y que fueron descubiertos por nuestros clientes. Cabe mencionar que aquí también reinicia el ciclo de desarrollo, pues día a día recibimos peticiones de nueva funcionalidad, la cual envuelve a todos los puntos anteriores.

## **6. Nota**

Este reporte carece de diagramas e imágenes relacionados con la empresa en la que actualmente laboro o los proyectos en que participé debido a que estos son propiedad de Aplika Soluciones Precisas.