[Título del documento]

[Subtítulo del documento]

Contenido

[2 Estructura de la base de datos. 3](#_Toc450050783)

[3 Modelos,vistas y controladores. 5](#_Toc450050784)

[4 Diagrama de flujo. 7](#_Toc450050785)

[5 Diagrama de arquitectura. 9](#_Toc450050786)

[6 Requisitos. 9](#_Toc450050787)

[7 Setup del código. 9](#_Toc450050788)

[8 Credenciales para pruebas. 9](#_Toc450050789)

El objetivo es diseñar e implementar una plataforma de suscripción e integrar con un webservice (ver documentación del webservice adjunta). Dicho webservice se usará para el envío de SMS y cobros.

La aplicación debe:

-Exponer métodos para el alta y baja de los usuarios en la plataforma.

-NO debe permitir el alta si el usuario no tiene fondos.

-Sistema para cobrar periódicamente a los usuarios suscritos.

-Soportar gran volumen de suscripciones y transacciones diarias (5-10 millones). Guardar un registro detallado de las altas, bajas, cobros y envío de SMS.

-Guardar registro de la comunicación con el webservice (peticiones / respuestas).

-La aplicación principal debe desarrollarse en PHP, haciendo uso del framework MVC Codeigniter.

-El uso de tecnologías auxiliares queda a criterio del desarrollador.

# Modelos,vistas y controladores.

La plataforma tiene los siguientes tres modelos:

-Plataforma\_model.php:

-users\_model.php:Contiene funciones para interactuar con las bases de datos para por ejemplo validar los datos introducidos al iniciar sesión.

-Webservices\_model.php:Contiene las funciones para “generar”, enviar el xml al webservice y guardar el recibido en un array el cual por una parte se devuelve al controlador y por otra guarda su contenido en los logs de la base de datos.

La plataforma tiene las siguientes vistas:

-alta.php: Nos muestra un texto y un checkbox para aceptar la condiciones del texto anterior el cual es obligatorio de marcar en caso de querer darse de alta y luego nos redirecciona al panel de control del usuario.

-baja.php:Nos muestra un texto y un checkbox para aceptar la condiciones del texto anterior el cual es obligatorio de marcar en caso de querer darse de baja y luego nos redirecciona al panel de control del usuario.

-login.php: Muestra un formulario para iniciar sesión, el nick o nombre de usuario es el número de teléfono,ademas muestra un enlace para registrarnos en caso de no tener cuenta, si nos autenticamos con éxito nos redirecciona al panel de control del usuario.

-signup.php: Muestra un formulario para registrar un usuario el cual consiste en el número de teléfono y la contraseña, una vez realizado el registro nos redirecciona al panel de control del usuario.

-menu.php: Muestra diferentes botones según si se ha iniciado la sesión o no, ademas nos marca con un color verde para indicar en que “página” nos encontramos actualmente, esta view se incluye siempre en el resto de las views.

-panel\_control.php: Muestra el número de teléfono del usuario con la sesión iniciada,un formulario para cambiar de contraseña el cual en el caso de estar correcto nos muestra una vista para cambiar la contraseña.

Por último en caso de estar no estar dado de alta en el servicio nos mostrara un enlace para darnos de alta en caso de tener fondos, en caso de no tener fondos y de algun error nos los mostrara con un mensaje en lugar del enlace, en caso de estar dada de alta nos saldra un link para dar de baja la suscripción.

-passwordUpdate.php:Muestra un formulario donde introducir la nueva contraseña y un botón para enviarla y luego nos redirecciona al panel de control del usuario.

-about.php:Muestra un texto el cual es un faq sobre el funcionamiento del servicio para el usuario.

La plataforma tiene los siguientes controladores:

-plataforma.php:Contiene funciones para cargar las vistas principales, dar de alta o baja a los usuarios en el servicio de suscripción.

-users.php:Contiene funciones para registrarse,validar los datos al iniciar sesión,cambiar la contraseña

-webservices.php: Contiene funciones para obtener tokens, peticiones de cobro y sms, ademas guarda lo anterior en logs en la bbdd, ademas tiene una funcion para comprobar si el usuario tiene saldo realizando un bill request para permitir que se de de alta y otra funcion que se ejecuta diariamente mediante un scheduler como Windows task scheduler o Crontab el sistemas Linux.

# Sistema para el cobro periodicos de los usuarios suscritos.

Para realizar los cobros periodicamente se ha creado una tarea con el programador de tareas de windows, está es ejecutada cada día a las 15:00 y lo que realiza es llamar a un archivo bat que ejecuta un navegador y carga la url que lanza la funcion que se encarga de realizar todo los necesario para cobrar a los usuarios.

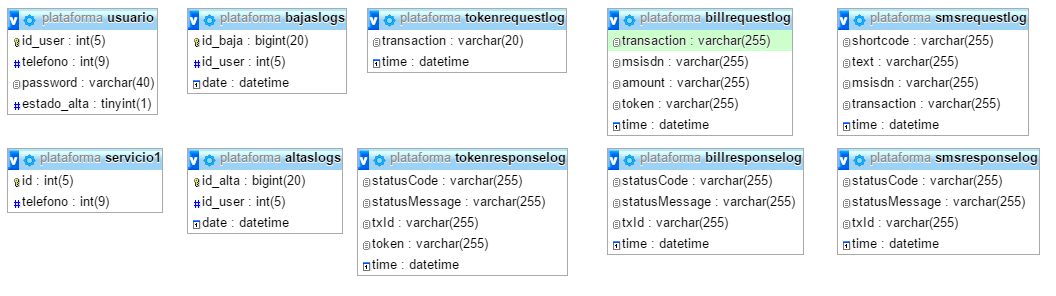
En caso de usar un sistema Linux se podria hacer lo mismo con crontab en lugar del programador de tareas de windows y un script shell en lugar del archivo bat.



# Estructura de la base de datos.

La base de datos tiene la siguiente estructura:





Las tablas de la base de datos siguen las siguientes estructuras:

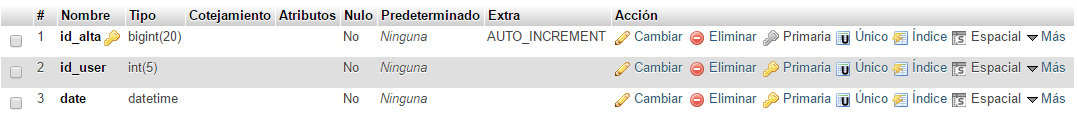
Tabla “altaslogs”, en ella guardamos un log el que registramos cuando un usuario se da de alta en el servicio.

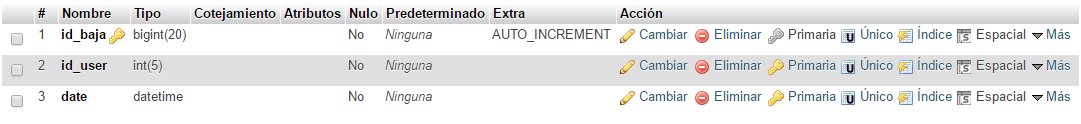
Tabla “bajaslogs”, en ella guardamos un log en el que registramos cuando un usuario se da de baja en el servicio.

Tabla “usuario”, en ella guardamos los datos de los usuarios para iniciar sesión y el estado de si la suscripción esta activada o desactivada.

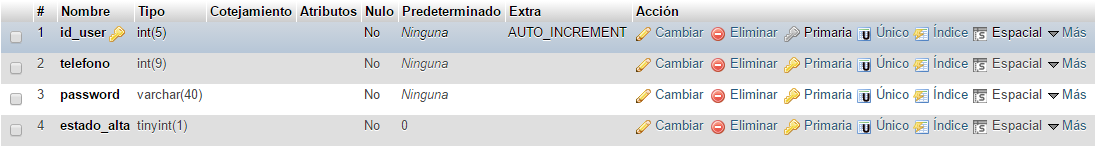


Tabla “servicio”, en ella guardamos los usuarios que estan dados de alta, la usamos para relizar los transacciones periodicas de forma cómoda.

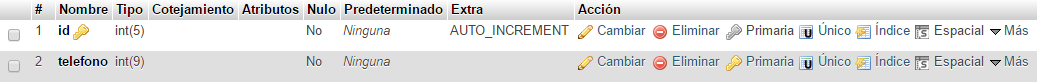


Tabla “tokenrequestlog”,en ella guardamos un log con el contenido del xml que se le envia al webservice para hacer el request del token.

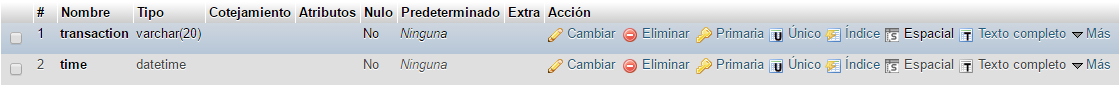


Tabla “tokenresponselog”,en ella guardamos un log con el contenido del xml que se recive del webservice despues de enviar el request del token.

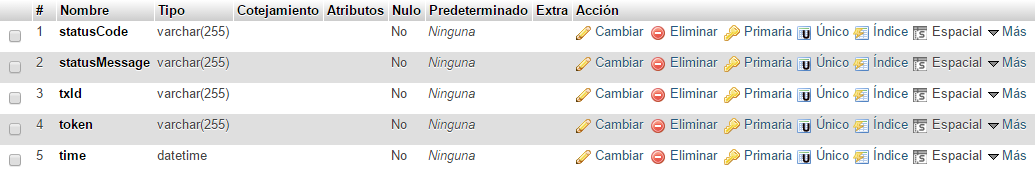


Tabla “billrequestlog”,en ella guardamos un log con el contenido del xml que se le envia al webservice para hacer el request de la bill.



Tabla “billresponselog”, en ella guardamos un log con el contenido del xml que se recive del webservice despues de enviar el request de la bill.

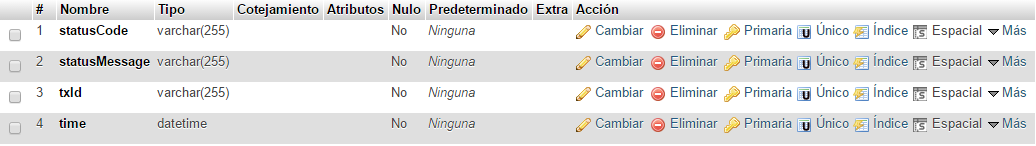
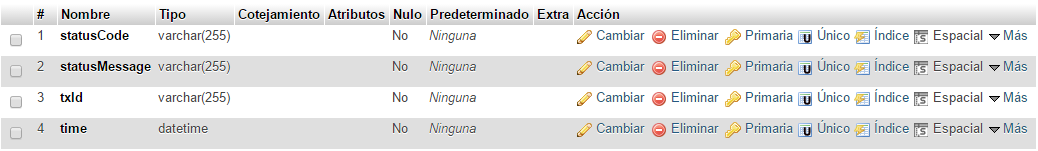


Tabla “smsrequestlog”, en ella guardamos un log con el contenido del xml que se le envia al webservice para hacer el request del sms.



Tabla “smsresponselog”, en ella guardamos un log con el contenido que se recive del webservice despues de enviar el request del sms.



# Diagrama de flujo.

Diagrama de flujo del sistema de registro y alta:

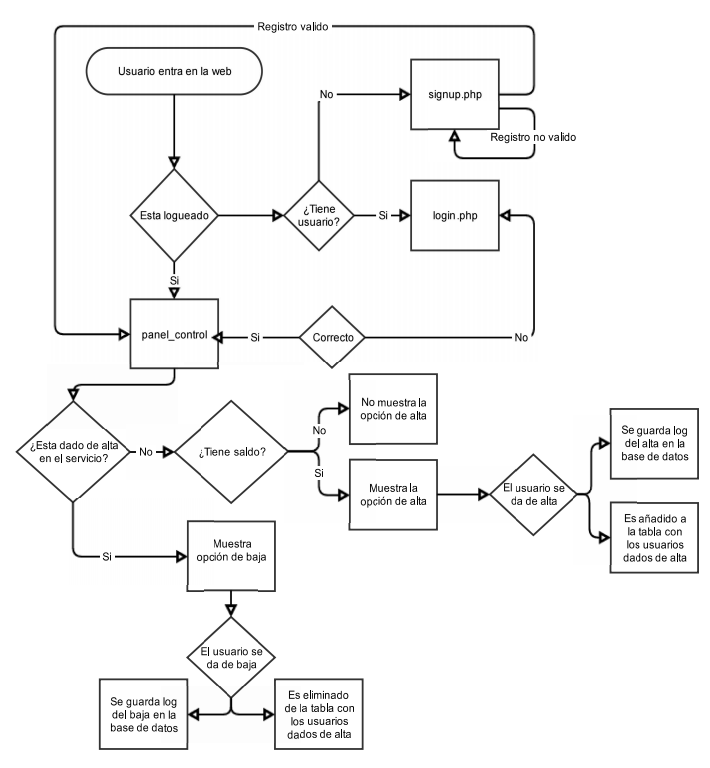
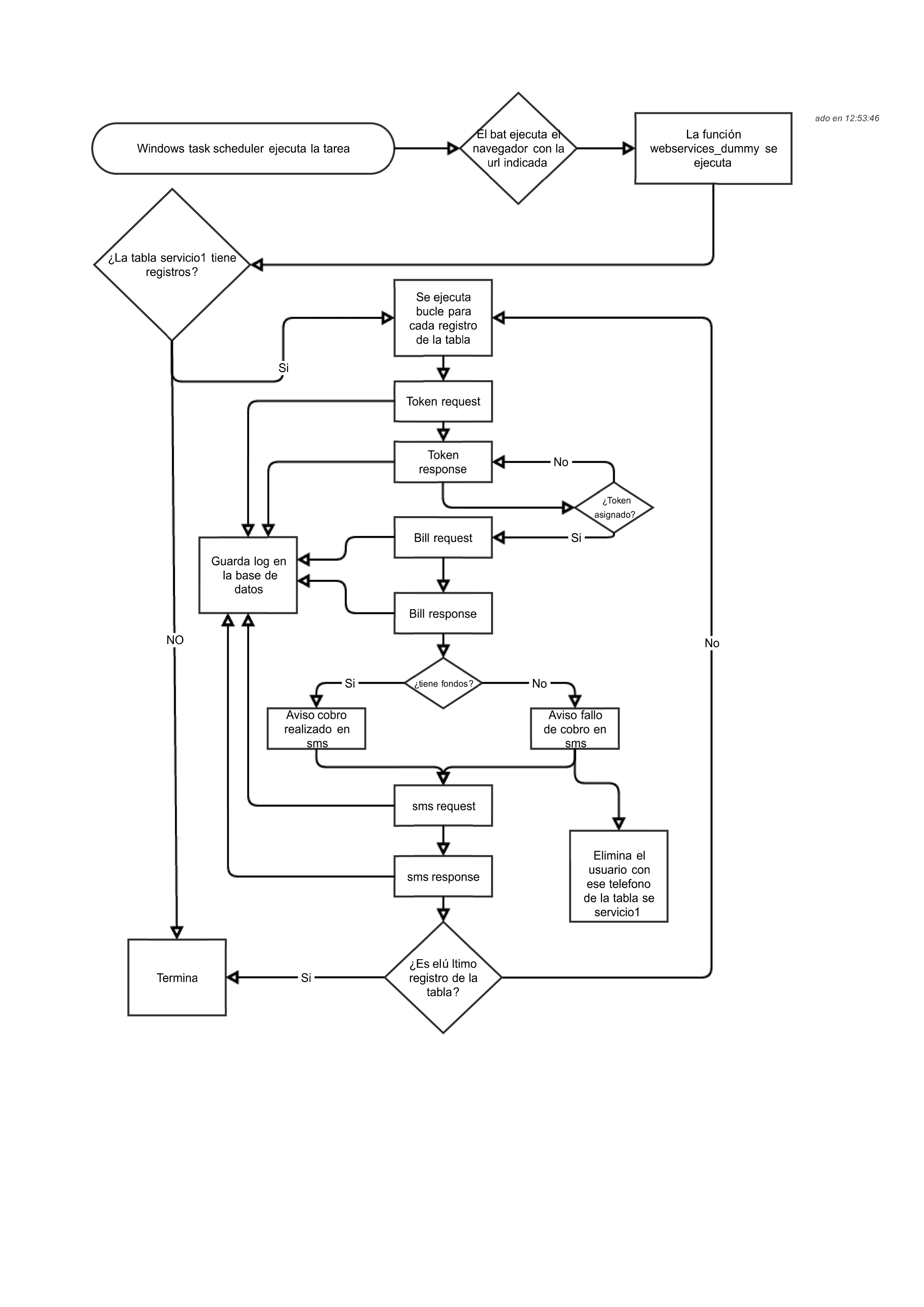
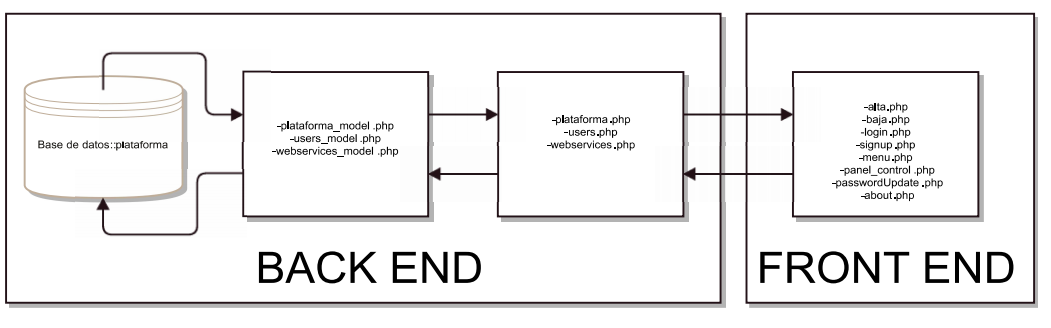


Diagrama de flujo del sistema de cobros:



# Diagrama de arquitectura.



# Requisitos.

-Framework Codeigniter.

-XAMPP o similar que incluya base de datos mysql y php.

# Setup del código.

-Servidor XAMPP o similar:

Instalar xampp y copiar en htdocs el proyecto obtenido de GitHub.

-Base de datos:

Importar el archivo .sql localizado en recursos el cual creara las tablas y unos registros para realizar pruebas.

# Credenciales para pruebas.

|  |  |
| --- | --- |
| Teléfono | Contraseña |
| 616616616 | qwerty |
| 606606606 | qwerty |