**UNIVERSIDAD CATÓLICA BOLIVIANA SAN PABLO**

**UNIDAD ACADÉMICA REGIONAL COCHABAMBA**

**Departamento de Ingeniería y Ciencias Exactas**

**Carrera de Ingeniería de Sistemas**

****

**Implementación de una aplicación móvil para el registro de pedidos a través de Webservices para la ciudad de Cochabamba**

*Proyecto Final del Diplomado en Programación Móvil*

**Ing. Daniel Cosio Burgoa**

Cochabamba – Bolivia

Octubre de 2016

**RESUMEN**

La presente propuesta busca proporcionar a la ciudadanía Cochabambina de una aplicación para dispositivos móviles especializada en la realización de pedidos a diferentes restaurantes y proveedores de alimentos de una forma rápida y sencilla.

El objetivo de este proyecto es usar el IDE de desarrollo ANDROID para implementar una herramienta que permita al usuario utilizar un Smartphone para comunicarse con un servidor de pedidos y, mediante el uso de webservices, realizar las siguientes funciones:

* Obtener la lista de restaurantes registrados en un servidor.
* Conocer el detalle del menú correspondiente a un restaurante específico.
* Registrar en el servidor un pedido de comida a un restaurante específico.

La metodología en el prototipo estará basada en SCRUM como proceso ágil de desarrollo.

**Conclusiones:**

* Aplicación Android para el registro de pedidos en un servidor particular.
* Bajos costos de mantenimiento y desarrollo.

***Palabras clave:***Aplicación, Pedidos, Restaurantes, *Smartphone*, Android.

|  |  |
| --- | --- |
| **PROBLEMA**   1. **OBJETIVOS**    1. **Objetivo General**    2. **Objetivos Específicos** 2. **ALCANCES** 3. **LIMITACIONES** 4. **TECNOLOGÍAS Y HERRAMIENTAS.**    1. **ARQUITECTURA DE SOFTWARE**       1. **Modelo Vista Controlador (MVC)**       2. **Arquitectura general de la aplicación**    2. **HERRAMIENTAS**       1. **Android Studio 1.4** 5. **DISEÑO DE LA APLICACIÓN** 6. **APENDICES** | 4  **5**  **5**  **5**  **5**  **8**  **9**  **9**  **9**  **12**  **12**  **12**  **13** |
|  |  |

PROBLEMA

Desde hace décadas muchas personas realizan pedidos y reserva de comida a través de llamadas telefónicas, pero Internet y los móviles están cambiando muchas cosas, incluida la forma en la que se realiza esta actividad.

Actualmente no existe una aplicación móvil para Android que permita a un usuario conocer diversos restaurantes de la ciudad de Cochabamba, acceder a su información en detalle como menús y precios y poder realizar un pedido específico.

La presente propuesta busca proporcionar a la ciudadanía Cochabambina de una aplicación móvil que permitirá al usuario acceder a un listado de restaurantes registrados en un servidor, ver sus menús y realizar pedidos mediante tecnología móvil de forma simple, ágil, escalable, con mínima inversión pero con resultados inmediatos.

Se propone una arquitectura simple, funcionamiento de la aplicación en entorno online, una interfaz amigable y un proceso de acceso a la información concreto y fácil, con el objetivo de proveer información a los usuarios finales.

1. **OBJETIVOS**
   1. **Objetivo General**

Diseñar e implementar una Aplicación Móvil que permita registrar en un servidor un pedido a un restaurante a través de webservices utilizando el IDE de desarrollo ANDROID.

* 1. **Objetivos Específicos**
* Elaborar el diseño de la aplicación para dispositivos Android.
* Utilizar el IDE de desarrollo ANDROID STUDIO para la implementación y desarrollo de la aplicación.
* Realizar el diseño e implementación de los webservices en formato JSON.
* Hacer uso de la información obtenida de los webservices en la aplicación.
* Implementar la interfaz de usuario y los requerimientos funcionales de cada pantalla.

1. **ALCANCES**

El proyecto desarrollado e implementado brindará la siguiente información:

* **Restaurantes:**
  + Navegación:
    - La navegación se realizará mediante el control ListView.
  + Datos:
    - Al momento de ingresar a la vista de restaurantes se llamará al servicio web que devolverá la lista de restaurantes registrados en el servidor.
* Online-Offline
  + - Si no existe conexión a internet se emitirá una alerta que indique al usuario que no se encuentra conectado a ninguna red.
  + Internacionalización
    - El título de la vista contará con internacionalización.
* Acceso a nuevas pantallas
  + - Cuando el usuario presione una fila del ListView accederá a la vista de platillos del restaurante seleccionado.
* **Platillos:**
  + Navegación:
    - La navegación se realizará será mediante el control ListView.
  + Datos:
    - Al momento de ingresar a la vista de platillos se llamará al servicio web que devolverá la lista de platillos del restaurante seleccionado en la vista de restaurantes.
* Online-Offline
  + - Si no existe conexión a internet se emitirá una alerta que indique al usuario que no se encuentra conectado a ninguna red.
  + Internacionalización
    - El título de la vista y los controles contarán con internacionalización.
  + Validación
    - Cuando el usuario presione el botón Aceptar se verificará que haya incrementado el campo Cantidad de al menos una fila.
* Acceso a nuevas pantallas
  + - Cuando el usuario presione el botón Aceptar accederá a la vista de pedido si pasó satisfactoriamente la validación.
* **Pedido:**
  + Navegación:
    - El botón Regresar implementará el evento back.
  + Datos:
    - Al momento de ingresar a la vista de Detalle se llamará al servicio web que devolverá el detalle de los platillos seleccionados en la vista de platillos, se procesarán los datos y se mostrará el resultado en pantalla.
  + Geo localización
    - La aplicación proporcionará las coordenadas actuales del usuario y las mostrará en un mapa.
* Online-Offline
  + - Si no existe conexión a internet se emitirá una alerta que indique al usuario que no se encuentra conectado a ninguna red.
  + Internacionalización
    - El título de la vista y los controles contarán con internacionalización.
* Acceso a nuevas pantallas
  + - Cuando el usuario presione el botón Regresar volverá a la vista de selección de platillos.
    - Cuando el usuario presione el botón Aceptar se validaran los datos ingresados y se dirigirá a la vista de pedidos realizados.
* **Pedidos realizados:**
  + Navegación:
    - La navegación se realizará será mediante el control ListView.
  + Datos:
    - Al momento de ingresar a la vista de pedidos realizados se llamará al servicio web que devolverá la lista de pedidos registrados con el nombre de usuario y nit ingresados en la vista de pedido.
* Online-Offline
  + - Si no existe conexión a internet se emitirá una alerta que indique al usuario que no se encuentra conectado a ninguna red.
  + Internacionalización
    - El título de la vista y los controles contarán con internacionalización.
* Acceso a nuevas pantallas
  + - Cuando el usuario presione una fila del ListView accederá a la vista detalle del pedido seleccionado.
    - Cuando el usuario presione el botón Aceptar retornará a la vista de restaurantes para poder realizar un nuevo pedido.

1. **LIMITACIONES**
   1. **Aplicación móvil**

* Sólo se realizará el prototipo para Android debido a las limitaciones del hardware disponible.
* La aplicación móvil Android funcionará en el entorno de desarrollo a través de un simulador.
* El proyecto no incluye la publicación y distribución de la aplicación móvil en la tienda de Google Play.
* La aplicación solo contempla el diseño y la implementación para dispositivos Android de gamma xxhdpi. No Tablet Android.
* La aplicación requiere de conexión a internet continua para el funcionamiento de la misma. El modo offline no guarda localmente imágenes ni datos.
  1. **Servidor web**
* El registro de restaurantes y platillos se realizará únicamente mediante la página web alojada en el servidor. El usuario solo podrá registrar pedidos en el servidor a través de la aplicación móvil.

1. **TECNOLOGÍAS Y HERRAMIENTAS.**
   1. **ARQUITECTURA DE SOFTWARE**
      1. **Modelo Vista Controlador (MVC)**

El modelo–vista–controlador (MVC) es un patrón de [arquitectura de software](https://es.wikipedia.org/wiki/Arquitectura_de_software) que separa los [datos](https://es.wikipedia.org/wiki/Datos) y la [lógica de negocio](https://es.wikipedia.org/wiki/L%C3%B3gica_de_negocio) de una aplicación de la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y el módulo encargado de gestionar los eventos y las comunicaciones.

Con este modelo de Arquitectura se pretende estructurar los proyectos de forma ordenada, eficiente y siguiendo las buenas prácticas de programación.

* + 1. **Arquitectura general de la aplicación e interacción con el servidor**

La aplicación se comunicará con el servidor a través de servicios web.

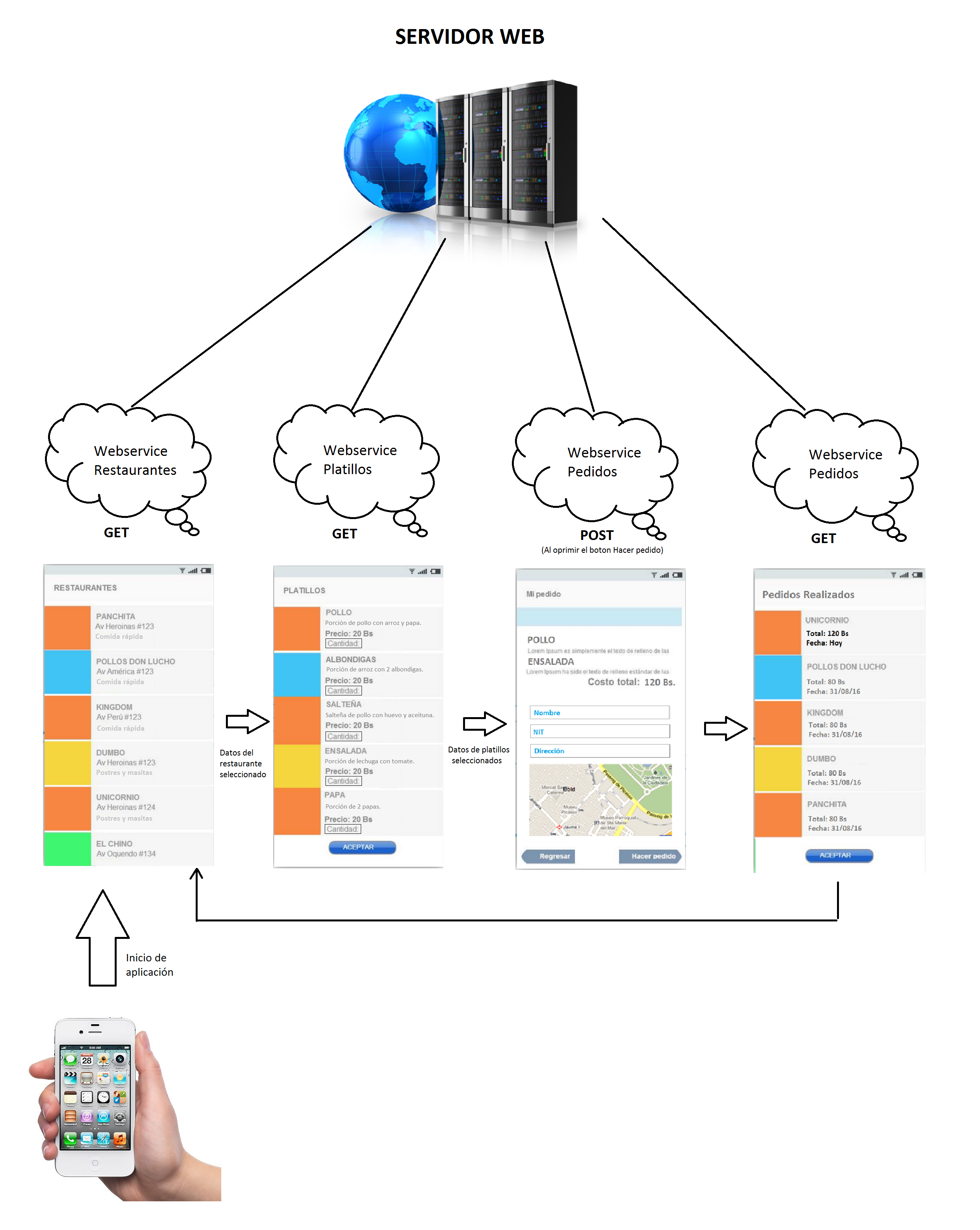
Un servicio web es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red. Generalmente, la interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor alojado en la web. El cliente (La aplicación móvil) solicita información, enviando datos al servidor para que pueda procesar su solicitud. El servidor genera una respuesta que envía de vuelta al cliente, adjuntando otra serie de datos que forman parte de esa respuesta. Por tanto, podemos entender un servicio web como un tráfico de mensajes entre dos máquinas.

Para los sistemas de comunicación entre máquinas se requiere una serie de características, básicamente marcadas porque las máquinas implicadas en la comunicación pueden tener sistemas muy diferentes. Pueden usar lenguajes de programación o bases de datos diferentes y hasta los sistemas operativos suelen ser distintos entre clientes y servidores. Por ello, para la comunicación de los datos se usa básicamente lenguajes escritos en archivos de texto plano.

Entre los lenguajes más usados para la comunicación de los datos en los servicios web, encontramos XML y JSON.

JSON es un formato de intercambio ligero que al igual que el XML puede ser  leído fácilmente  y es independiente de la plataforma, puede ser convertido a javascript rápidamente y por lo tanto es ideal para aplicaciones Web. La ventaja de JSON es que es capaz de representar la información en un formato más ligero que el XML y por lo tanto es más rápido de transportar y consume menos ancho de banda.

Con la utilización de JSON podemos disminuir de manera considerable la cantidad de datos que enviamos y recibimos al servidor lo cual convertirá a nuestra aplicación en un sistema mucho más escalable y que responderá mucho mejor ante una sobre carga de usuarios.

**Diagrama de interacción con el servidor**

* 1. **HERRAMIENTAS**
     1. **Android Studio**

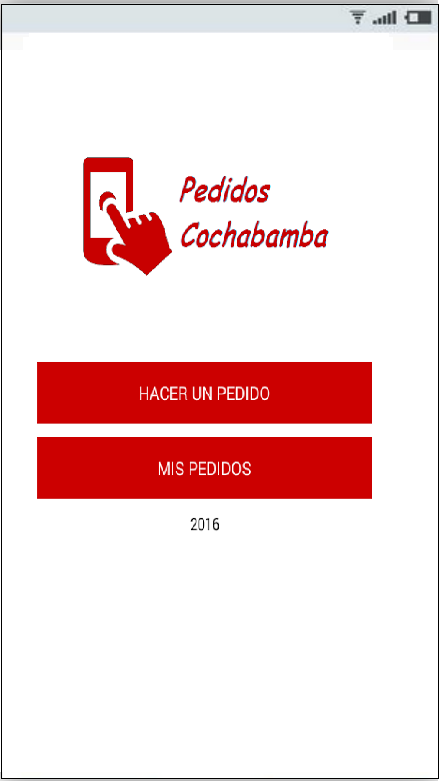
Android Studio es un entorno de desarrollo integrado ([IDE](http://academiaandroid.com/ide-entornos-integrados-de-desarrollo-para-android/)), basado en [IntelliJ IDEA](http://academiaandroid.com/ide-android-intellij-android-studio-aide/" \t "_blank" \o "IDE para Android: IntelliJ IDEA, Android Studio y AIDE) de la compañía JetBrains,  que proporciona varias **mejoras con respecto al plugin ADT** (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre [Apache 2.0](http://es.wikipedia.org/wiki/Apache_License), está programado en Java y es multiplataforma.

**Características de Android Studio**

* Renderización en tiempo real.
* Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
* Soporte para construcción basada en [Gradle](https://es.wikipedia.org/w/index.php?title=Gradle&action=edit&redlink=1" \o "Gradle (aún no redactado)).
* [Refactorización](https://es.wikipedia.org/wiki/Refactorizaci%C3%B3n) especifica de Android y arreglos rápidos.
* Herramientas [Lint](https://es.wikipedia.org/wiki/Lint" \o "Lint) para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
* Plantillas para crear diseños comunes de Android y otros componentes.
* Soporte para programar aplicaciones para [Android Wear](https://es.wikipedia.org/wiki/Android_Wear" \o "Android Wear).

1. **DISEÑO DE LA APLICACIÓN**

**Pantalla inicial**

****

**Proceso de registro de pedido**

**Paso 1.** Selección de Restaurante

****

**Ejemplo webservice\_restaurante?op=GET**

[

{"id":"6","nombre":"PANCHITA","direccion":"Av. Heroinas #123","tipo":"Comida Rapida","logo":"Pollos-Panchita-Review-Cochabamba-Logo.jpg"},

{"id":"1","nombre":"POLLOS DON LUCHO","direccion":"Av. America # 123","tipo":"Comida Rapida","logo":"polllos don lucho.jpg"},

{"id":"3","nombre":"KINGDOM","direccion":"Av. Santa Cruz # 987","tipo":"Comida Rapida","logo":"Logo.png"},

…

]

**Paso 2.** Selección de Platillos



**Ejemplo webservice\_platillo?nid\_restaurante=1 (GET)**

[

{"id":"15","nombre":"Pollo","descripcion":"Porcion de pollo con arroz y papa",""precio":"20.00","foto":"pollo broaster.jpg"},

{"id":"16","nombre":"Albondigas","descripcion":"Porcion de arroz con 2 albondigas","""precio":"20.00","foto":"albondigas.jpg"},

{"id":"14","nombre":"Salteña","descripcion":"Salteña de pollo con huevo y aceituna""precio":"20.00","foto":"chicharon de pollo.jpg"}

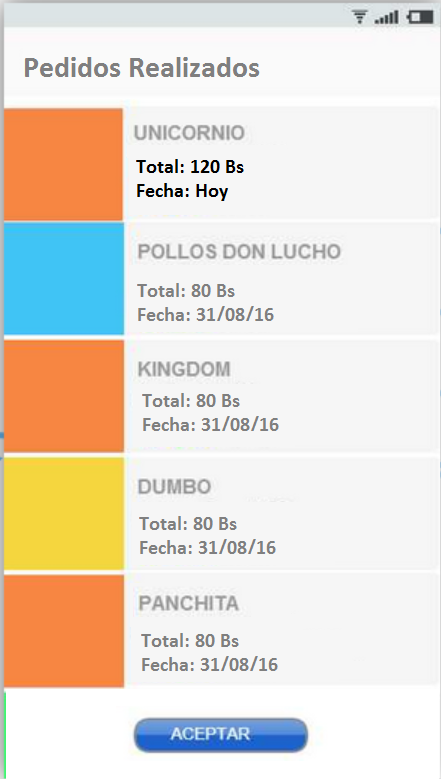
…

]

**Paso 3. Registro de** Pedido



**Paso 4.** Pedidos Realizados

****

**Ejemplo webservice\_pedido?usuario=DanielCosio&nit=123456789&imei=12345678(GET)**

**[**

**{"id":"4","restaurante":"UNICORNIO","destino":"Av. Heroinas #456","total":"80","fecha":"20/08/2016"},**

**{"id":"1","restaurante":"POLLOS DON LUCHO","destino":"Av. America # 123","total":"80","fecha":"20/08/2016"},**

**{"id":"3","restaurante":"KINGDOM","destino":"Av. Santa Cruz # 987","total":"80","fecha":"20/08/2016"}**

**…**

**]**

1. **APENDICES**

**Link a repositorio:**

<https://github.com/DanielCosio/Proyecto-Diplomado-Moviles>