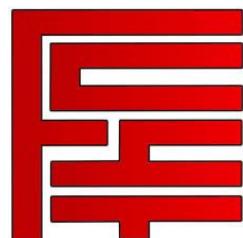




UNIVERSIDAD MAYOR DE SAN SIMON  
FACULTAD DE CIENCIAS Y TECNOLOGIA  
INGENIERIA INFORMATICA

---



# **APLICACIÓN MÓVIL EN ANDROID QUE EXTIENDE SERVICIOS DE LA APLICACIÓN SAGAA PARA LA DISPONIBILIDAD DE INFORMACIÓN**

Proyecto de grado presentado para optar al título de Licenciatura en Ingeniería  
Informática

---

**ELABORADO POR:** GEOVANNA LIZETTE GIL TERCEROS

**TUTOR:** MSC. ING. JORGE WALTER ORELLANA ARAOZ

COCHABAMBA - BOLIVIA

AGOSTO 2017

## Dedicatoria

*Dedicado a  
mi familia*



## Agradecimientos

A Dios primeramente, por brindarme la vida y a mi familia por brindarme su amor incondicional y por apoyarme en los momentos mas difíciles y por sabios consejos.



# Tabla de Contenido

<b>Dedicatoria</b>	I
<b>Agradecimientos</b>	III
<b>Lista de abreviaturas</b>	IX
<b>Lista de figuras</b>	XIII
<b>1. Introducción</b>	1
1.1. Antecedentes . . . . .	1
1.2. Definición del problema . . . . .	1
1.3. Descripción del problema . . . . .	1
1.4. Objetivo general . . . . .	2
1.5. Objetivos específicos . . . . .	2
1.6. Justificación . . . . .	2
1.7. Alcance . . . . .	2
<b>2. Fundamento teórico</b>	3
2.1. Sistemas Distribuidos . . . . .	3
2.1.1. Computación cliente servidor . . . . .	3
2.2. La arquitectura orientada a servicio web . . . . .	4
2.2.1. Protocolo de servicio web . . . . .	5
2.3. Protocolo REST . . . . .	5
2.3.1. Protocolo HTTP . . . . .	5
2.4. Unidad de datos del servicio . . . . .	5
2.4.1. Json . . . . .	6
2.4.2. Json web token . . . . .	6
2.5. Las aplicaciones móviles . . . . .	7
2.5.1. La aplicación nativa . . . . .	7
2.5.2. La aplicación web . . . . .	7
2.5.3. Aplicación Híbrida . . . . .	7
2.6. Los dispositivos móviles . . . . .	7
2.6.1. El sistema operativo android . . . . .	8
2.7. El diseño adaptativo . . . . .	8
2.8. Las plataformas de desarrollo . . . . .	8
2.8.1. El entorno de desarrollo VIM . . . . .	8

2.8.2. El navegador web - Chrome . . . . .	9
2.9. Herramientas de Desarrollo . . . . .	9
2.9.1. Framework ionic . . . . .	9
2.9.2. Framework angularJS . . . . .	9
2.9.3. Framework node.js . . . . .	9
<b>3. Análisis de la planilla de notas</b>	<b>11</b>
3.1. Entrevista . . . . .	11
3.2. Diagrama de proceso de llenar la planilla de notas . . . . .	12
3.3. Sistema de apoyo a la gestión académica y administrativa . . . . .	12
3.3.1. Funcionalidad de modificar la planilla de notas . . . . .	13
3.3.2. Los pasos para descargar la planilla de notas . . . . .	13
3.3.3. Los pasos para publicar la planilla de notas . . . . .	15
3.3.4. Las restricciones de la página SAGAA . . . . .	16
3.4. El transcriptor . . . . .	17
3.4.1. La funcionalidad de la aplicación del transcriptor . . . . .	17
3.4.2. Detalle del Transcriptor . . . . .	18
3.4.3. Las restricciones del transcriptor . . . . .	20
3.5. La planilla de notas . . . . .	21
3.5.1. La información de la planilla de notas . . . . .	21
3.5.2. La estructura de la planilla de notas . . . . .	23
3.5.3. Las restricciones de la planilla de notas . . . . .	24
<b>4. Proceso de desarrollo de servicio</b>	<b>25</b>
4.1. El servicio web como componente de reutilización . . . . .	25
4.2. La ingeniería de servicio . . . . .	25
4.3. Servicios de sistemas heredados . . . . .	27
4.4. Desarrollo de software con servicios . . . . .	27
4.4.1. Diseño e implementación del flujo de trabajo . . . . .	28
4.4.2. Pruebas del servicio . . . . .	28
4.5. El diseño metodológico . . . . .	29
4.6. Conclusiones . . . . .	30
<b>5. Diseño e implementación del servicio web</b>	<b>31</b>
5.1. Identificar el candidato a servicio . . . . .	31
5.1.1. Requerimientos del servicio web . . . . .	31
5.2. Diseño del servicio . . . . .	32
5.2.1. La etapa de diseño de interfaz lógica . . . . .	32
5.2.2. Diseño de mensaje . . . . .	33
5.2.3. Desarrollo de WSDL . . . . .	34
5.2.4. Especificación del servicio . . . . .	36
5.2.5. Implementación y despliegue del servicio . . . . .	37
5.3. Desarrollo de la implementación del servicio . . . . .	39
5.3.1. Herramientas y configuración para la implementación del servicio . . . . .	40
5.3.2. Estructura del servicio web . . . . .	41
5.3.3. Configuración del servicio web . . . . .	41

5.3.4. Aplicar el servicio web en Express . . . . .	41
<b>6. Análisis y Diseño de la aplicación móvil</b>	<b>45</b>
6.1. Requerimientos . . . . .	45
6.2. Desarrollo de módulos del proyecto . . . . .	45
6.2.1. Lista de módulos del proyecto . . . . .	46
6.2.2. Lista de submodulo del proyecto . . . . .	46
6.3. Diseño de interfaz . . . . .	46
6.4. Diseño de la comunicación entre cliente y servicio web . . . . .	48
<b>7. Implementación de la aplicación móvil</b>	<b>49</b>
7.1. Herramienta y configuración de la aplicación móvil . . . . .	49
7.1.1. Estructura del Proyecto . . . . .	50
7.1.2. Configuración del proyecto . . . . .	51
7.2. Herramientas Extras . . . . .	51
7.2.1. Framework cordova . . . . .	52
7.2.2. PouchDB . . . . .	52
7.2.3. Base de datos websql . . . . .	53
7.2.4. Almacenamiento local . . . . .	53
7.2.5. Interceptor . . . . .	53
7.2.6. Json web token . . . . .	53
7.3. Implementación del Proyecto . . . . .	54
7.3.1. El primer intento o iteración la configuración para conectarse al servicio web . . . . .	54
7.3.2. El segundo intento o iteración la sesión . . . . .	55
7.3.3. El tercer intento o iteración trabajar sin el servicio web . . . . .	56
<b>8. Implementacion del diseño adaptativo para la aplicación móvil</b>	<b>59</b>
8.1. Diseño adaptativo o responsive . . . . .	59
8.2. Componentes de Ionic . . . . .	59
8.3. Implementación de flexbox . . . . .	60
8.4. Visualización del contenido con flexbox . . . . .	60
<b>9. Pruebas del servicio web a través de la aplicación móvil</b>	<b>63</b>
9.1. Descargar la planilla de notas . . . . .	63
9.1.1. La sesión . . . . .	63
9.1.2. Seleccionar la carrera para descargar la planilla de notas retorna la planilla de notas . . . . .	64
9.2. Modificar la planilla de notas . . . . .	65
9.2.1. Filtrar y convierte la planilla de notas . . . . .	65
9.2.2. Enviar la planilla de notas . . . . .	65
9.2.3. Modificar la planilla de notas . . . . .	66
9.3. Adjuntar la planilla de notas . . . . .	66
9.3.1. Seleccionar la gestion adjuntar la planilla de notas . . . . .	66
9.3.2. Seleccionar el grupo para habilitar estudiante . . . . .	67
<b>10. Conclusiones</b>	<b>69</b>

<b>Bibliografía</b>	<b>69</b>
<b>A. Cuestionario de identificacion del servicio web</b>	<b>73</b>

# **Lista de abreviaturas**

UMSS	Universidad Mayor de San Simón
SAGAA	Sistema de Apoyo a la Gestión Académica y Administrativa
WEBSISS	Web Sistema de Información San Simón
UPSI	Unidad de Provisión de Servicios de Información
ASO	Arquitectura Orientado a Servicios
BPMN	Proceso de Negocio de Modelo de Notación
URL	Localizador Uniforme de Recursos
ODBC	Abrir la Conecctividad de la Base de Datos
BD	Base de Datos
HTTP	Hipertexto de Transferencia de Protocolo
REST	Representación Estándar de Transferencia
SaaS	Software como un Servicio
MEMI	Mejoramiento de Enseñanza de Matemática Informática
MVC	Modelo Vista y Controlador
FCYT	Facultad de Ciencias y Tecnología
UML	Unidad de Lenguaje Modificado
XML	Lenguaje de Marcado Extensible
WSDL	Lenguaje de descripción de servicios web

X

# **Lista de figuras**

2.1. Computación cliente-servidor, adoptado para la realización del proyecto. Arquitectura cliente servidor de [1] . . . . .	4
2.2. Arquitectura orientado a servicio, adoptado para la realización del proyecto. El objetivo del SOA de [1] . . . . .	4
2.3. Representación de json por objeto, adoptado para la realización del proyecto. Estructura de json por objeto, de Figura: Elaboración propia . . . . .	6
2.4. Representación del json por arreglo, adoptado para la realización del proyecto. Estructura de json por arreglo de Fuente: Elaboración propia . . . . .	6
2.5. Representación del json web token Fuente: Elaboración propia . . . . .	6
2.6. Capas del sistema operativo de android, adoptado para la realización del proyecto. Herramienta de android de [2] . . . . .	8
3.1. Diagrama de la funcionalidad de llenar y mostrar las notas, en la página del SAGAA Fuente: Elaboración propia . . . . .	12
3.2. Iniciar sesión en la página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT . . . . .	13
3.3. Elegir menú Acádémico Pre-Grado Fuente: Proporcionado por la UPSI de la FCYT . . . . .	13
3.4. Elegir la opción descargar la planilla de notas Fuente: Proporcionado por la UPSI de la FCYT . . . . .	14
3.5. Opción de gestión, Fuente: Proporcionado por la UPSI de la FCYT . . . . .	14
3.6. Elegir la descarga de una planilla de notas, Fuente: Proporcionado por la UPSI de la FCYT . . . . .	15
3.7. Paso 1: Opción de habilitar estudiantes Fuente: Proporcionado por la UPSI de la FCYT . . . . .	15
3.8. Paso 2: Opción de adjuntar el archivo seleccionado Fuente: Proporcionado por la UPSI de la FCYT . . . . .	16
3.9. Paso 3: Subir planilla de notas por grupo Fuente: Proporcionado por la UPSI de la FCYT . . . . .	16
3.10. Buscar planilla de notas del Transcriptor Fuente: Proporcionado por el MEMI . . . . .	17
3.11. Buscar planilla de notas Fuente: Proporcionado por el MEMI . . . . .	18
3.12. Lista de estudiantes, Fuente: Proporcionado por el MEMI . . . . .	18
3.13. Detalle de la lista de grupo Fuente: Proporcionado por el MEMI . . . . .	19
3.14. Detalle de la lista de estudiantes Fuente: Proporcionado por el MEMI . . . . .	20
4.1. El proceso de ingeniería de servicio, adoptado para la realización del proyecto. El proceso de ingeniería de servicio [1] . . . . .	26
4.2. Etapas de la Construcción de servicio, adoptado para la realización del proyecto. Construcción de servicio [1] . . . . .	27

4.3. Secuencia de pasos para el proceso de ingeniera. Fuente: En base a la construcción de servicio de Somerville [1] . . . . .	29
5.1. Diseño de interfaz lógico para descargar la planilla de nota, Fuente: Elaboración propia	32
5.2. Diseño de interfaz lógico para adjuntar la planilla de notas, Fuente: Elaboración propia	33
5.3. Diseño de mensaje de entradas, salidas y excepciones, Fuente: Elaboración propia .	33
5.4. Proceso de descargar Planilla Notas. Figura: Elaboración propia . . . . .	34
5.5. Proceso de modificar la Planilla de Notas, Fuente: Elaboración propia . . . . .	35
5.6. Proceso de subir a la página del SAGAA, Fuente: Elaboración propia . . . . .	35
5.7. Protocolos de servicio, Fuente: Elaboración propia . . . . .	36
5.8. Diagrama de flujo de trabajo, Fuente: Elaboración propia . . . . .	36
5.9. Diseño de flujo de trabajo de descarga de la planilla de notas, Fuente: Elaboración propia . . . . .	38
5.10. Diseño de flujo de trabajo de la descarga de la planilla de notas, Fuente: Elaboración propia . . . . .	38
5.11. Diseño de flujo de trabajo de subir la planilla de notas, Fuente: Elaboración propia . .	38
5.12. Diseño de flujo de trabajo planilla de notas, Fuente: Elaboración propia . . . . .	39
5.13. Diseño de flujo de trabajo planilla de notas para la implementación, Fuente: Elaboración propia . . . . .	39
5.14. Arquitectura de Express, adoptado para la realización del proyecto. Arquitectura de Express de [3]	40
5.15. Arquitectura de NodeJs, adoptado para la realización del proyecto. Arquitectura de NodeJs de [4] . . . . .	40
6.1. Interfaz de Usuario primer intento, Fuente: Elaboración Propia . . . . .	47
6.2. Interfaz de Usuario representado en mockups, Fuente: Elaboración Propia . . . . .	47
6.3. Diseño de comunicación entre el cliente y servicio web Fuente: Elaboración Propia . .	48
7.1. Arquitectura de Ionic, adoptado para la realización del proyecto Fuente: [5] . . . . .	50
7.2. Arquitectura de Cordova, adoptado para la realización del proyecto Fuente: [6] . . . . .	52
7.3. Adaptadores de PouchDB, adoptado para la realización del proyecto Fuente: [7] . . . .	52
7.4. Base de datos websql, adoptado para la realización del proyecto Fuente: [8] . . . . .	53
7.5. Json web token, adoptado para la realización del proyecto Fuente: Elaboración propia	54
8.1. Componente de CSS de Ionic, Fuente: Página oficial . . . . .	60
8.2. Visualizacion de contenido vertical en la tablet, Fuente: Elaboración propia . . . . .	61
8.3. Visualización de contenido horizontal en la tablet, Fuente: Elaboración propia . . . . .	61
9.1. La aplicación móvil solicita la sesión al servicio web, Fuente: Elaboración propia . . . . .	64
9.2. Respuesta de la página del SAGAA al servicio web, Fuente: Elaboración propia . . . . .	64
9.3. Aplicación móvil selecciona la carrera para descargar la planilla de notas, Fuente: Elaboración propia . . . . .	64
9.4. Peticion descarga de la planilla de notas, Fuente: Elaboración propia . . . . .	64
9.5. Aplicación móvil lista los datos de la planilla de notas, Fuente: Elaboración propia . .	65
9.6. Envia los datos de la planilla de notas en Json, Fuente: Elaboración propia . . . . .	65
9.7. Información general, Fuente: Elaboración propia . . . . .	65
9.8. La información de los grupos, Fuente: Elaboración propia . . . . .	65

9.9. La lista de estudiantes, Fuente: Elaboración propia . . . . .	65
9.10. El servicio web, recibe la planilla de notas, Fuente: Elaboración propia . . . . .	66
9.11. El servicio web, busca y reemplaza los datos modificado en la planilla de notas, Fuente: Elaboración propia . . . . .	66
9.12. El servicio web, envia la gestion y adjunta la planilla de notas, Fuente: Elaboración propia . . . . .	66
9.13. El servicio web, énvia el grupo que debe ser modificado, Fuente: Elaboración propia .	66
9.14. En el servicio web, es la respuesta de la página del SAGAA, Fuente: Elaboración propia	67



# **Capítulo 1**

## **Introducción**

En el transcurso del tiempo el manejo de información de las organizaciones ha mejorado de forma exponencial, a través del avance en el área de la informática con los servicios web. El cual ofrecen una disposición de información accesible a otros clientes. Es el caso en que las páginas web, tienen la necesidad de mejorar la disposición y generalizar el formato de información con respecto a las nuevas tecnologías. Para el presente proyecto se ha utilizado la página del seguimiento y control de los estudiantes, docentes y postulantes, denominada SAGAA<sup>1</sup>, de la FCYT<sup>2</sup> de la UMSS<sup>3</sup>.

### **1.1. Antecedentes**

Los servicios web surgen finalmente para estandarizar la información y la comunicación entre diferentes plataformas, con la capacidad de interoperar en la web. Estas tecnologías intercambian datos entre ellas con el fin de ofrecer servicios. Es por lo tanto en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML<sup>4</sup>, SOAP<sup>5</sup> y REST<sup>6</sup>. [1](Sommerville, 2011)

En la actualidad muchos sistemas están pasando a ser servicios web. Un buen ejemplo es el Twitter, donde gracias a un servicio web particular, cualquier aplicación puede leer o incluso escribir tuits en nombre de los usuarios. [9]

### **1.2. Definición del problema**

La ausencia de accesibilidad de información, de la página del SAGAA, en dispositivos actuales.

### **1.3. Descripción del problema**

Para el presente proyecto se ha utilizado, como análisis de prueba la página del SAGAA, tiene la funcionalidad de descargar, modificar y publicar la planilla de notas de los estudiantes, el cual es

---

<sup>1</sup>Sistema de Apoyo a la Gestión Académica y Administrativa

<sup>2</sup>Facultad de Ciencias y Tecnología

<sup>3</sup>Universidad Mayor de San Simón

<sup>4</sup>XML- Lenguaje de marcas extensibles

<sup>5</sup>SOAP- Acceso de protocolos de objetos simples

<sup>6</sup>REST - Transferencia de estado representacional

utilizado por el plantel docente de la FCYT <sup>7</sup>.

Las funcionalidades producen dificultades en los procesos, las cuales son:

- Disponibilidad de información de la planilla de notas no es adecuada
- Páginas no actualizadas.
- Solo solo para sistema operativo windows.
- los procesos se repiten y la visibilidad no es adecuada para dispositivos actuales.

## **1.4. Objetivo general**

Proveer los servicios de la aplicación SAGAA a través de una aplicación móvil para lograr el mejoramiento de la disponibilidad de la información.

## **1.5. Objetivos específicos**

1. Construir un mecanismo de recolección de datos del Sistema SAGAA para la interacción con dispositivos móviles.
2. Proveer un diseño adaptativo para la aplicación móvil.
3. Implementar la estructura de dato distribuido para la información local del dispositivo móvil.
4. Implementar un proceso de sincronización para la transferencia de información para la aplicación móvil.

## **1.6. Justificación**

Debido a que la tecnología esta avanzando, en gran manera en el área de las páginas web y en frente a la dificultad del proceso de registro de planilla de notas. Nace el incentivo para llevar a cabo el presente proyecto que pretende crear el servicio web de la página del SAGAA de la Facultad de Ciencias y Tecnologías de la Universidad Mayor de San Simón, para lograr la disponibilidad de información de una manera más clara y efectiva a través de una aplicación móvil.

## **1.7. Alcance**

Se pretende realizar el proyecto para mejorar la disponibilidad de información de la página del SAGAA, a través de una aplicación móvil, el cuál tiene las siguientes restricciones:

1. La aplicación móvil, solo contemplara la descarga y el subir de la planilla de notas.
2. El llenado de planilla de notas, es para la Facultad de Ciencias y Tecnología.
3. Se delimitara la version de android.

---

<sup>7</sup>Facultad de ciencia y tecnología

## Capítulo 2

# Fundamento teórico

Las páginas web han mejorado continuamente en la disponibilidad de la información a través de los servicios web denominado SOA<sup>1</sup>, los cuales utilizan el protocolo de comunicación para que la aplicación móvil puede realizar una solicitud al servicio web.

### 2.1. Sistemas Distribuidos

Los sistemas distribuidos son un conjunto de computadoras independientes para mostrar a los usuarios como un solo sistema en diferentes dispositivos tanto en hardware como en software y se comunican en red para coordinar sus acciones mediante el envío de mensaje(Colourios, 2012). [10]

Según Colourios las ventajas, para utilizar un sistema distribuido son:

- a) **El intercambio de recursos:** permite compartir hardware y software de sus recursos a través de la red.
- b) **Las aperturas distribuidas:** se utilizan los diseños estándares de protocolos.
- c) **La concurrencia:** son los procesos que podrían operar al mismo tiempo en computadoras separadas sobre la red. Estos procesos podrían comunicarse con otros, durante su operación normal.
- d) **La escalabilidad:** es aumentar nuevos recursos al sistema.

El sistema distribuido se organiza como un sistema de cliente y servidor.

#### 2.1.1. Computación cliente servidor

La computación de cliente servidor se refiere al usuario, interactúa con un programa y se ejecuta en la computadora local. Este interactúa con un programa que se ejecuta en otra computadora remota, el cual proporciona servicios como acceso a páginas web [1]. Como se muestra en la figura 2.1.

---

<sup>1</sup>SOA- Arquitectura orientado a servicios

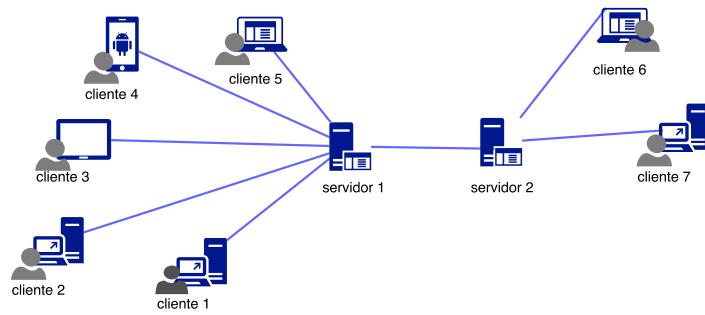


Figura 2.1: Computación cliente-servidor, adoptado para la realización del proyecto. Arquitectura cliente servidor de [1]

Para implementar un sistema cliente servidor se tiene que instalar un programa en la computadora cliente para que se comunique con el servidor. Este software se denomina un servicio remoto, conocido como SOA que se explica a continuación.

## 2.2. La arquitectura orientada a servicio web

El arquitectura orientada a servicios es un componente de software de reutilización, debidamente ajustado, se accede de manera programática. [1]. Como se muestra en la figura 2.2.

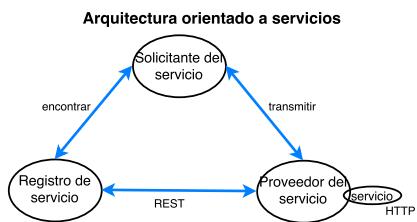


Figura 2.2: Arquitectura orientado a servicio, adoptado para la realización del proyecto. El objetivo del SOA de [1]

Las características del servicio web son las siguientes:

- Compartir información entre empresas.
- La información debe tener una representación estándar.
- Es independiente de la aplicación o sistema que usa el servicio web.
- El servicio web es independiente de la plataforma y del lenguaje de implementación.
- Desde el principio, tiene un proceso estándar para el servicio web en donde las compañías se han comprometido a dicho estándar.

Para el presente proyecto la implementación y la interface del servicio, son definidos como un sistema heredado.

### **2.2.1. Protocolo de servicio web**

El protocolo de servicio web es un conjunto de reglas, que utilizan para la comunicación e intercambio de información entre el protocolo mencionado anteriormente, a través de una red (Navarro, 2006). [11]. Según Navarro los protocolos se dividen en dos tipos:

- Protocolos estándares los cuales son : SOAP, WSDL<sup>2</sup>, WS-BPEL<sup>3</sup>.
- Protocolo no estándar es el metodo REST.

Para el presente proyecto se ha utilizado el protocolo REST, se desarrolla en el siguiente párrafo.

## **2.3. Protocolo REST**

Es un estilo de arquitectura de software, se refiere a una colección de principios para el diseño de arquitecturas en la red. El término frecuentemente es utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP<sup>4</sup>. Se basa en estándares de protocolo HTTP(Navarro, 2006). [11].

### **2.3.1. Protocolo HTTP**

El protocolo HTTP es un conjunto de métodos de petición para indicar la acción que se desea realizar, en el recurso determinado cada uno de ellos implementan una semántica diferente, pero tienen alguna característica similar, las cuales son: Get, Head, Post, Put, Delete, Connect, Options, Trace y Patch.(Navarro, 2006) [11].

Según Navarro, las funciones sobre la web que maneja HTTP son: el cache, los requisitos de origen,el proxie, el túnel y sesión. Para este proyecto se han utilizado algunas funciones, como:

- a) **El cache:** es la función para almacenar los documentos en la memoria del navegador.
- b) **Requisitos de origen:** son funciones para compartir la información de datos y puede flexibilizar la información entre cliente y servidor.
- c) **La autenticación:** es la función que establece la sesión y almacena información en el cokie para guardar en el navegador.

## **2.4. Unidad de datos del servicio**

La unidad de datos del servicio es un termino genérico, para compartir información entre el servicio y el cliente. Estas unidades de datos pueden ser: Json<sup>5</sup> , Xml y Yaml<sup>6</sup>. Json es el que se emplea en este proyecto, por lo tanto explicaremos mas sobre este a continuación.

---

<sup>2</sup>WSDL - Descripción de lenguajes de servicios web

<sup>3</sup>WS-BPEL - Lenguaje de Ejecución de procesos de negocio con servicios web

<sup>4</sup> HTTP- Hipertexto de Transferencia de Protocolo

<sup>5</sup>Json- Notación de objetos de javascript

<sup>6</sup>Otro lenguaje de marcado más

### 2.4.1. Json

El Json es un formato ligero y abierto de intercambio de datos. Utiliza el texto plano para compartir información el cual es independiente del lenguaje del lenguaje de programación. Las estructura para compartir información pueden ser objetos o arreglo y también se puede combinar ambos, como se muestra en la figura 2.3 y en la figura 2.3 que se muestran a continuación.

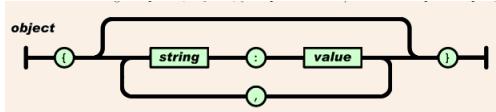


Figura 2.3:  
Representación de json  
por objeto, adoptado  
para la realización del  
proyecto. Estructura de  
json por objeto, de  
Figura: Elaboración  
propia

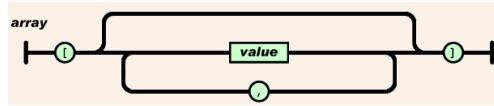


Figura 2.4:  
Representación del json  
por arreglo, adoptado  
para la realización del  
proyecto. Estructura de  
json por arreglo de  
Fuente: Elaboración  
propia

### 2.4.2. Json web token

El json web token conocido como JWT<sup>7</sup> es un método abierto y estándar para representar las solicitudes de forma segura entre dos partes. Este se divide 3 partes: la cabecera, la carga útil y una firma.

Figura 2.5: Representación del json web token Fuente: Elaboración propia

En este proyecto el tipo de servicio web es identificado como un servicio heredado porque realiza solicitud de datos para la aplicación móvil.

<sup>7</sup>JWT - Json web token

## **2.5. Las aplicaciones móviles**

Según la corporación de IBM<sup>8</sup> define a las aplicaciones móviles en tres partes, las cuales son: nativa, web y híbrido. [12]

### **2.5.1. La aplicación nativa**

La aplicación nativa se conecta directamente con el sistema operativo móvil, sin ningún intermediario ni contenedor. La aplicación nativa puede acceder libremente a todas las Apis que el proveedor del sistema operativo ponga a su disposición y en muchos casos, tiene características y funciones únicas que son típicas del sistema operativo móvil en particular.

### **2.5.2. La aplicación web**

La aplicación web utiliza únicamente tecnologías basadas en la web como ser la quinta versión de HTML<sup>9</sup>, el mismo tiene componente de interfaz de usuario avanzado, acceso a múltiples tipos de medios, servicios de geoposicionamiento y disponibilidad sin Internet.

Una de ventaja de esta aplicación es el soporte para múltiples plataformas y se ejecuta dentro del navegador.

### **2.5.3. Aplicación Híbrida**

La aplicación híbrida tiene un enfoque híbrido que combina desarrollo nativo con tecnología web. La aplicación híbrida utiliza múltiples plataformas y tiene acceso a los dispositivos del celular tales son: la cámara, acceso de dato, almacenamiento de información y otros.

En este proyecto se ha utilizado la aplicación híbrida porque permite acceder a la parte nativa del móvil y almacena información en la aplicación.

## **2.6. Los dispositivos móviles**

Actualmente, el dispositivo móvil es pequeño para ser transportado y utilizado durante su transporte. Se caracteriza por el mejoramiento que va adquiriendo su tamaño reducido, la telecomunicación de software, la gran interacción entre las personas. Convirtiéndolos en una necesidad primaria para la sociedad [13].

Desde su creación los dispositivos móviles han evolucionado en gran magnitud, es por eso que muchas empresas han ofrecido diferentes sistemas operativos como ser: android, iOS, Windows, etc. El sistema operativo de android se utiliza para el presente proyecto.

---

<sup>8</sup>Máquinas de negocios internacionales

<sup>9</sup>HTML - Lenguaje de Marcado para Hipertexto

### 2.6.1. El sistema operativo android

El sistema operativo android basada en la plataforma de software Linux para dispositivos móviles. El sistema operativo android es un sistema operativo libre y tiene acceso a sus recursos como la pantalla, camara y otros [2]. A continuación consta con las siguientes capas, se muestra en la siguiente figura 2.6.

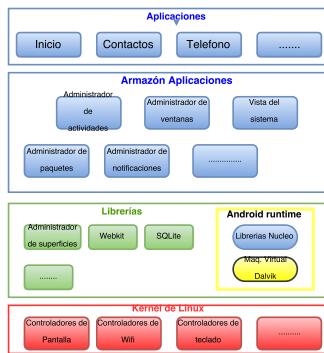


Figura 2.6: Capas del sistema operativo de android, adoptado para la realización del proyecto. Herramienta de android de [2]

Con el avance de la tecnología han aumentado el incremento, en diferentes dispositivos como ser tablet PC, tablet, smartphone y otros. El cual utilizan el sistema operativo android. Debido a la diversidad de dispositivos móviles, para el presente proyecto se utiliza el diseño adaptativo que se explica a continuación.

### 2.7. El diseño adaptativo

El diseño adaptativo se refiere a la satisfacción de visitar una página web a través de diferentes dispositivos, ha generado la filosofía de Responsive Web Design, este pensamiento fue establecida por Steven Champeon (2003), quién propone realizar un único diseño web para todos los dispositivos y accesible para todos los usuarios [14].

### 2.8. Las plataformas de desarrollo

Se han utilizado dos herramientas para el desarrollo, las cuales son: el entorno de desarrollo vim y el navegador de chrome que desarrollaremos a continuación.

#### 2.8.1. El entorno de desarrollo VIM

Según la página oficial de Vim, el Vim se define como un editor de texto altamente configurable construido para creaciones y cambios de cualquier tipo de texto. Tiene las siguientes ventan-

jas: persistente multinivel, extensivos plugin para el sistema, soportan diferentes lenguajes de programación y diferentes formatos de archivos y otros.

### **2.8.2. El navegador web - Chrome**

Es un navegador web rápido, seguro y gratuito, diseñado para la web actual, así lo define la página oficial del mismo.

También es un navegador para dispositivos móviles. La ventaja que brinda es verificar los errores para celulares, minimizar los gastos de datos, almacenar la sesión y permitir crear base datos en el navegador.

## **2.9. Herramientas de Desarrollo**

Para el presente proyecto se han utilizado tres framework, los cuales son ionic, angularjs y nodejs. Estos se basan en el lenguaje de javaScript, los cuales se ha utilizado para la implementación del servicio web y la aplicación móvil. A continuación se explican los framework:

### **2.9.1. Framework ionic**

Según la página oficial de ionic este framework es un SDK para HTML5 que nos ayuda a construir una aplicación móvil híbrida usando las tecnologías como HTLM, CSS y Javascript. Se caracteriza por ser multiplataforma y utiliza el framework de angularJS.

### **2.9.2. Framework angularJS**

Según la página oficial de angularJS este es un conjunto de herramientas para construir el framework mas adecuado para desarrollo de aplicaciones. Es totalmente extensible y trabaja con otras librerías, cada característica puede ser modificado o remplazado, para adaptarse a su flujo de trabajo.

### **2.9.3. Framework node.js**

El framework de nodeJS es un entorno de ejecución para javaScript construido con el motor JavaScript V8 de Chrome. Nodejs utiliza un modelo de operación de E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema mas grande de librerías de código abierto en el mundo. Según la página oficial de nodejs.



## **Capítulo 3**

# **Análisis de la planilla de notas**

El plantel docente de la Facultad de Ciencias y Tecnología de la Universidad Mayor de San Simón, utiliza la página del Sistema de Apoyo a la Gestión Académica y Administrativa, el cual tiene una secuencia de pasos para descargar y publicar la planilla de notas. Para reconocer y modificar la planilla de notas se utiliza, la aplicación del transcriptor.exe la cual tiene diferentes casos, dependiendo de la estructura de la planilla de notas.

### **3.1. Entrevista**

La siguiente entrevista se realiza al *Ing. Cristian Lazarte*, responsable del proyecto Transcriptor.exe en la UPSI<sup>1</sup> de la UMSS<sup>2</sup>. A continuación se muestra los resultados de la entrevista en los siguientes puntos.

- Explicación de la estructura de planilla de notas, el archivo sis:
  - El código de la UPSI.
  - El código de template define la estructura que utiliza el Transcriptor.exe.
  - En el template el ARB son valores permitidos, para A de aprobado, R reprobado y B abandonado.
  - Los datos del grupo están definidos en el template y las restricciones de las calificaciones.
- Mencionamos algunos de los aspectos importantes y/o límite que considera a la hora de modificar la planilla de notas en el transcriptor.exe.
  - Modificar las notas se debe mostrar en grillas.
  - Los estados para volver atrás.
  - Cuando modifica la lista de estudiantes, no debe permitir cerrar sin guardar.
  - Habilitar la 2da opción y tomar en cuenta el mayor entre la 1ra y 2da opción.
  - En el enter, ir al siguiente dato para introducir.

---

<sup>1</sup>Unidad de Provisión de Servicios de Información

<sup>2</sup>Universidad Mayor de San Simón

- Analizar la visualización en el dispositivo móvil.

La entrevista ha proporcionado la comprensión de la estructura y la funcionalidad al llenar la planilla de notas.

### 3.2. Diagrama de proceso de llenar la planilla de notas

Los procesos de llenar la planilla de notas tiene los siguientes pasos los cuales: descargar la planilla de notas, modificar la planilla de notas y subir la planilla de notas. Se explica con mas detalle en el capítulo 5 en la sección de ??.

La unión de los proceso se representa en un diagrama general como se muestra en la figura 3.1

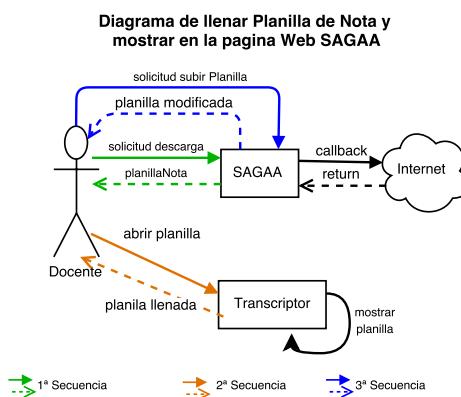


Figura 3.1: Diagrama de la funcionalidad de llenar y mostrar las notas, en la página del SAGAA Fuente: Elaboración propia

A continuación se explica la funcionalidad del llenado de planilla de notas de la página SAGAA<sup>3</sup> y la herramienta del transcriptor.exe.

### 3.3. Sistema de apoyo a la gestión académica y administrativa

El sistema de apoyo a la gestión académica y administrativa tiene alrededor de 10 años proporcionando el servicio de gestionar la información académica de pre-grado de los estudiantes.

La funcionalidad de la información académico de pre-grado tiene los siguientes procesos, las cuales son: llenar currículum, publicar y modificar avisos materias, descargar y habilitar la planilla de notas, ver estudiantes habilitados, administrador de archivos y manual.

Para el presente proyecto se utiliza la funcionalidad de modificar la planilla de notas que tiene el proceso de descargar y habilitar la planilla de notas como una herramienta de prueba. A continuación se explica la funcionalidad mencionada anteriormente.

<sup>3</sup>Sistema de Apoyo a la Gestión Académica y Administrativa

### 3.3.1. Funcionalidad de modificar la planilla de notas

La funcionalidad de modificar la planilla de notas tiene los siguientes procesos descargar, modificar y adjuntar la planilla de notas. A continuación se explican.

### 3.3.2. Los pasos para descargar la planilla de notas

Para realizar la descarga de la planilla de notas tiene 5 pasos de manera secuencial como se explican a continuación:

**Paso 1:** Primeramente, se debe escribir la dirección de la página en el navegador: `http://pruebas.fcyt.umss.edu.bo/sagaa`. Despues se debe acceder al link *Ingresar al sistema*, debe ingresar su cuenta de usuario y contraseña así como se muestra en la figura 3.2.



Figura 3.2: Iniciar sesión en la página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT

**Paso 2:** Elegir el menú de usuario *Académico Pre-Grado* como se muestra en la figura 3.3.



Figura 3.3: Elegir menú Académico Pre-Grado Fuente: Proporcionado por la UPSI de la FCYT

**Paso 3:** Elegir el menú la opción *Descargar Planilla de Notas*, como se muestra en la figura 3.4.

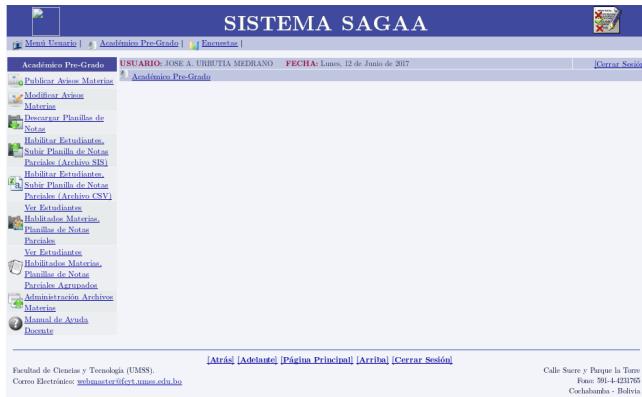


Figura 3.4: Elegir la opción descargar la planilla de notas Fuente:  
Proporcionado por la UPSI de la FCYT

**Paso 4:** Seleccionar la gestión de la lista de opciones *Seleccionar Gestión*, como se muestra en la figura 3.5.



Figura 3.5: Opción de gestión, Fuente: Proporcionado por la UPSI de la FCYT

**Paso 5:** Seleccionar el ícono de descarga, de la lista de carrera para *Descargar la planilla de notas*, como se muestra en la figura 3.6.

NRO	PLAN	ARCHIVO	NÚMERO DE DESCARGAS	ULTIMA DESCARGA	DESCARGAR ZIP
1	ELECTRICA(209701)	URRUTIA_MEDRANO_JOSE_A_19880008_-2016-06-26	181	12-20-2016	
2	ELECTROMECANICA(60001)	URRUTIA_MEDRANO_JOSE_A_19880008_-2016-06-26	2	08-11-2016	
3	ELECTRONICA(429701)	URRUTIA_MEDRANO_JOSE_A_19880008_-2016-06-26	0	-	

Figura 3.6: Elegir la descarga de una planilla de notas, Fuente:  
Proporcionado por la UPSI de la FCYT

Finalmente, se tiene el documento de planilla de notas, en donde se debe llenar las notas y después se debe realizar el proceso de publicación de la planilla de notas.

### 3.3.3. Los pasos para publicar la planilla de notas

Para publicar la planilla de notas, se realiza el paso 1 de realizar la sesión y el paso 2 elegir el menú *Académico Pre-Grado* de descargar la planilla de notas 3.3.2, después se debe realizar los siguientes pasos:

**Paso 1:** Elegir el menú la opción, *Habilitar Estudiantes*, *Subir Planilla de Notas Parciales (Archivo SIS)*, como se muestra en la figura 3.7.

Figura 3.7: Paso 1: Opción de habilitar estudiantes Fuente: Proporcionado por la UPSI de la FCYT

**Paso 2:** Adjuntar archivo y seleccionar la gestión de la lista de opciones, después de hacer click en *Subir Archivo*, como se muestra a continuación en la figura 3.8.

Universidad Mayor de San Simón

SISTEMA SAGAA

Facultad de Ciencias y Tecnología

USUARIO: JOSE A. URUTIA MEDRANO FECHA: Lunes, 12 de Junio de 2017 [Cerrar Sesión]

Habilitar Estudiantes, Subir Planilla de Notas Parciales (Archivo SIS)

Habilitar Estudiantes

Notas: La información de notas en esta página es informativa, la única página WEB de publicación de notas oficiales es el WEBSITE. Siendo responsable del usuario la correcta publicación.

Habilitar ESTUDIANTES, SUBIR NOTAS PARCIALES

Gestión: 2 - 2016 SEGUNDO SEMESTRE

Archivo SIS: Choose File URRUTIA\_...2016.xls

Subir Archivo

Atrás | Adelante | Página Principal | Arriba | Cerrar Sesión

Facultad de Ciencias y Tecnología (UMSS)  
Correo Electrónico: [uhmuser@fcyt.umss.edu.bo](mailto:uhmuser@fcyt.umss.edu.bo)

Calle Surie y Parque la Torre  
Fono: 591-4-421265  
Cochabamba - Bolivia

Figura 3.8: Paso 2: Opción de adjuntar el archivo seleccionado Fuente:  
Proporcionado por la UPSI de la FCYT

**Paso 3:** Seleccionar el grupo para adjuntar la planilla de notas y por ultimo, click en *Finalizar Operación* como se representa en la siguiente figura3.9.

Universidad Mayor de San Simón

SISTEMA SAGAA

Facultad de Ciencias y Tecnología

USUARIO: JOSE A. URUTIA MEDRANO FECHA: Lunes, 12 de Junio de 2017 [Cerrar Sesión]

Habilitar Estudiantes, Subir Planilla de Notas Parciales (Archivo SIS)

Habilitar Estudiantes

Notas: La información de notas en esta página es informativa, la única página WEB de publicación de notas oficiales es el WEBSITE. Siendo responsable del usuario la correcta publicación.

Habilitar ESTUDIANTES, SUBIR PLANILLAS DE NOTAS PARCIALES

Docente: JOSE A. URUTIA MEDRANO

Gestión: 2-2016 SEGUNDO SEMESTRE

Plan: 209701 LICENCIATURA EN INGENIERIA ELECTRICA

NRO	CODIGO MATERIA	NOMBRE MATERIA	GRUPO	SUBIR NOTAS
1	201404	ELECTRONICA DIGITAL I	I	<input type="checkbox"/>
2	201404	ELECTRONICA DIGITAL I	ME	<input type="checkbox"/>

Finalizar Operación | Cancelar Trabajo

Atrás | Adelante | Página Principal | Arriba | Cerrar Sesión

Facultad de Ciencias y Tecnología (UMSS)  
Correo Electrónico: [uhmuser@fcyt.umss.edu.bo](mailto:uhmuser@fcyt.umss.edu.bo)

Calle Surie y Parque la Torre  
Fono: 591-4-421265  
Cochabamba - Bolivia

Figura 3.9: Paso 3: Subir planilla de notas por grupo Fuente: Proporcionado por la UPSI de la FCYT

Para finalizar, se muestra el mensaje *Finalizo la habilitación de estudiantes y la planilla de notas* y accede nuevamente al paso 1.

### 3.3.4. Las restricciones de la página SAGAA

Para concluir con el análisis de la página del SAGAA, debemos de tomar en cuenta las siguientes restricciones:

- Solo permite subir archivo de la planilla de notas con extensión sis.
- Se tiene que trabajar de modo conectado a internet para realizar el subir o bajar la planilla de notas.

## 3.4. El transcriptor

El transcriptor es un programa ejecutable tiene la funcionalidad de modificar la planilla de notas. Previamente, se descarga el documento de planilla de notas de la página del SAGAA<sup>4</sup>. El creador el Lic. Cristian Lazarte, quién actualmente trabaja en la UPSI<sup>5</sup> realizando mantenimiento a dicha aplicación.

### 3.4.1. La funcionalidad de la aplicación del transcriptor

La funcionalidad del transcriptor es llenar la planilla de notas las cuales se describen en los siguientes pasos.

**Paso 1:** Primeramente seleccionar el botón *Abrir* como se observa en el círculo azul. En segundo lugar se busca la ubicación de la planilla de notas, como se observa en la figura 3.10.



Figura 3.10: Buscar planilla de notas del Transcriptor Fuente: Proporcionado por el MEMI

**Paso 2:** Seleccionar un grupo y hacer click en el círculo azul, en el botón *Editar*, se representa en la figura 3.11.

<sup>4</sup>Sistema de Apoyo a la Gestión Académica y Administrativa

<sup>5</sup>Unidad de Provisión de Servicios de Información

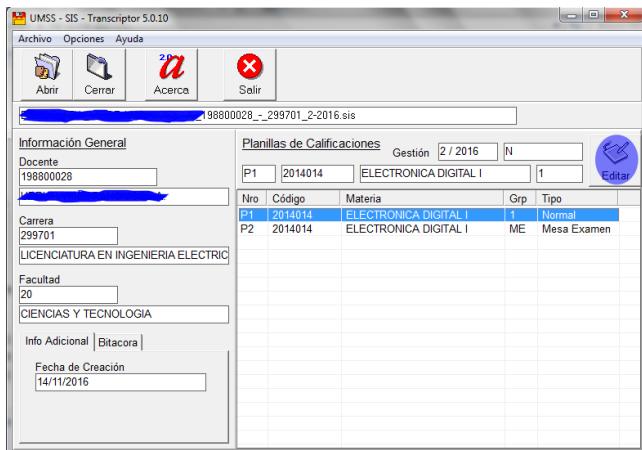


Figura 3.11: Buscar planilla de notas Fuente: Proporcionado por el MEMI

**Paso 3:** En el círculo rosado se debe llenar las notas de forma ordenada, se representa en la figura 3.12.

NRO	CODEST	NOMBRE	1ERPAR	2DOPAR	POME	EXAFIN	2da	NOTFIN	NOTCO
1	200206684	AGUILAR TORRICO RAMIRO	0	0	0	0	0	B	
2	201400485	AGUIRRE QUISPE LITZY DALMIRA	0	0	0	0	0	B	
3	200708877	CAYO ROSAS SCARLEY JIMENA	0	0	0	0	0	B	
4	201206854	CHILENO MAMANI EMERSON GROVE	0	0	0	0	0	B	
5	201302515	CHUCA CONDO EYNAR	0	0	0	0	0	B	
6	201101123	CHUGAR MANCILLA GONZALO	0	0	0	0	0	B	
7	201107013	FLORES VILLARROEL WILSON SERG	0	0	0	0	0	B	
8	201300942	JIMENEZ TOLA ADELaida ABIGAIL	0	0	0	0	0	B	
9	201308678	LUNA ALVAREZ SAMUEL BRANDON	0	0	0	0	0	B	
10	201001346	NINA CADIMA JOEL	0	0	0	0	0	B	
11	199909022	PERREDO VALENCIA JUAN MARCELO	0	0	0	0	0	B	

Figura 3.12: Lista de estudiantes, Fuente: Proporcionado por el MEMI

**Paso 4:** Guardar la planilla de notas y hacer click en el botón *Guardar* para cerrar la ventana y guardar los datos en el archivo de planilla de notas como se muestra en la figura 3.12.

Después de concluir con descargar y adjuntar la planilla de notas, se modifica la planilla de notas con el transcriptor.exe.

### 3.4.2. Detalle del Transcriptor

El transcriptor interpreta los datos de la planilla de notas que se encuentra con la extensión sis, como se muestra en los siguientes casos a continuación en la figura 3.13.

**Paso 1:** Mostrar los datos de la información general y la lista de materias, como se muestra en la figura 3.13

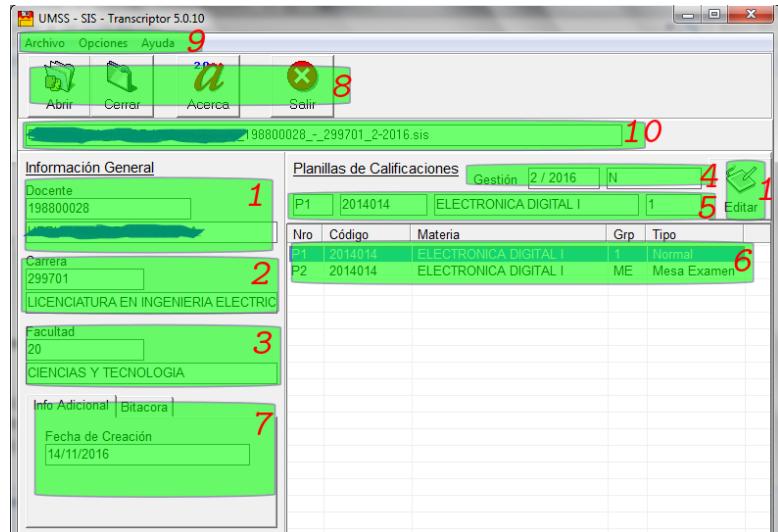


Figura 3.13: Detalle de la lista de grupo Fuente: Proporcionado por el MEMI

Se divide en las siguientes partes, las cuales están enumeradas según el detalle:

- 1) Los datos del docente: el código, el nombre y apellido.
- 2) Los datos de la carrera: el código y nombre.
- 3) Los datos de la facultad: el código y nombre.
- 4) El dato de la gestión.
- 5) Los datos del grupo: el código, nombre, código de reconocimiento y número de grupo.
- 6) La lista de grupo: el número, código, nombre, identificador y tipo.
- 7) La información adicional de una bitácora de la fecha de creación y bitácora del archivo.
- 8) En el menú del número 8 tiene los eventos los cuales son: abrir, cerrar, acerca y salir.
- 9) En el menú del número 9 consta de los eventos: archivo, opciones y ayuda.

**Paso 2:** Mostrar la lista de estudiantes, como se muestra en la figura 3.14.

Nro	CODEST	NOMBRE	1ERPAR	2DOPAR	PROMEDIO	EXAFIN	2da	NOTFIN	NOTCO
1	200206684	AGUILAR TORRICO RAMIRO		0			0	B	1
2	201400485	AGUIRRE QUISPE LITZY DALMIRA		0		0	0	B	
3	200708877	CAYO ROSAS SCARLEY JIMENA		0		0	0	B	
4	201206854	CHILENO MAMANI EMERSON GROVE		0		0	0	B	
5	201302515	CHUCA CONDO EYNAR		0		0	0	B	
6	201101123	CHUGAR MANCILLA GONZALO		0		0	0	B	
7	201107013	FLORES VILLAROEL WILSON SERG		0		0	0	B	
8	201300942	JIMENEZ TOLA ADELaida ABIGAIL		0		0	0	B	
9	201308678	LUNA ALVAREZ SAMUEL BRANDON		0		0	0	B	
10	201001346	NINA CADIMA JOEL		0		0	0	B	
11	199909022	PEREIDO VALENCIA JUAN MARCELO		0		0	0	B	

Figura 3.14: Detalle de la lista de estudiantes Fuente: Proporcionado por el MEMI

La lista de estudiantes se divide en las siguientes partes, las cuales están enumeradas detalladamente en la siguiente lista:

- 1) La lista de estudiantes consta con los siguientes datos: número, código, nombre, 1er parcial, 2do parcial, promedio, examen final, 2da instancia, nota final y nota contado.
- 2) Los datos del grupo son: código de reconocimiento, tipo y número de grupo.
- 3) Los datos de la materia son: el código y nombre.
- 4) El dato de gestión.
- 5) En los datos extra de los estudiantes estan : la cantidad, porcentaje de reprobado, aprobado y abandonado.
- 6) La lista de menú son guardar, imprimir, imprimir 2da instancia, pesos, copia todo, importar csv, exportar csv y salir.

### 3.4.3. Las restricciones del transcriptor

La aplicación del transcriptor, según el detalle y la funcionalidad, tiene las siguientes restricciones:

- a) Se puede usar únicamente en el sistema operativo windows.
- b) Solo permite abrir archivos con la extensión sis.
- c) No permite abrir un documento de planilla de notas, si estas dañado.
- d) Controla los datos a ser insertados.
- e) Protege la información del archivo y solo permite modificar la nota del estudiante.
- f) Solo permite modificar la planilla de nota por grupo.
- g) No permite cerrar la lista de grupos, sin previamente haber guardado.

Después de concluir con el análisis del transcriptor, se empieza a analizar el contenido de la planilla de notas que tiene la extensión sis.

### **3.5. La planilla de notas**

El documento de la planilla de notas es un estándar que utiliza la UMSS<sup>6</sup> para la modalidad de calificación de las diferentes carreras. Para explicar la utilidad de la planilla de notas se divide en la información y la estructura de la planilla de notas.

#### **3.5.1. La información de la planilla de notas**

La información se divide en datos academicos, la estructura del formato de los grupos, la estructura de los grupos y estructura de la información del estudiante. En el siguiente párrafo se explicara.

- a) Los datos academicos se explica en cada línea del documento de la planilla de notas, la cual tiene la siguiente estructura.

```
PCD5.0 //codigo inicio
2719 //codigo fin
09/08/2016 //fecha de creacion
09/08/2016 //fecha de creacion
198800028 //codigo docente
URRUTIA MEDRANO JOSE A.//nombre y apellido docente
299701//codigo carrera
LICENCIATURA EN INGENIERIA ELECTRICA //nombre carrera
2016// gestion
1 //numero gestion
Primer Semestre //literal del semestre
20 //codigo facultad
CIENCIAS Y TECNOLOGIA //nombre facultad
11 //codigo UPSI
UPSI nombre Upsi
263 //tipo de template
tecnico //carrera de template
26 //nota maxima para el template
```

- b) La estructura del formato de grupos se explica en forma horizontal del documento de planilla de notas.

```
\\"a,b ,c ,d,e ,f,g ,h ,i,j,k
1,NUMERO,RD,0,100,,NRO,38,N,N,NONE
.....
10,NOTCON,RD,0,100,ARB,NOTCON,50,N,N,ERES
```

---

<sup>6</sup>Universidad Mayor de San Simón

- 1) 1 es el orden de la fila.
- 2) NUMERO es identificador de la etiqueta ordenado por fila.
- 3) RD o WR permiso de lectura o escritura.
- 4) 0 es el inicio del rango de datos.
- 5) 100 o 51 es el final del rango de datos.
- 6) A o R o B es el valor permitido.
- 7) NRO nombre de la etiqueta ordenado por fila.
- 8) 38 o 72 o 230 o 52 o 50 ancho de la columna.
- 9) N para habilitar si tiene nota final.
- 10) N para habilitar si tiene segunda instancia.
- 11) NONE palabra reservado.

La estructura del formato de los grupos depende del formato de calificación es por lo cual el grupo normal tiene 10 columnas y el grupo mesa tiene 7 columnas.

- c) La estructura del formato de los grupos se divide por la coma se explican a continuación de manera secuencial.

Detalle grupo normal:

```
\\"a, b, ,c, d ,e , f ,g,h ,i ,j,k,l,m
P1,2014014,1,ELECTRONICA DIGITAL I,263,100,0,50,50,0,0,0,0
P2,2014014,ME,ELECTRONICA DIGITAL I,263,100,0,50,50,0,0,0,0
```

- 1) P1 o P2 es el código grupo.
- 2) 2014014 es el código de la materia.
- 3) 1 o ME es el tipo de grupo.
- 4) ELECTRONICA DIGITAL I es el nombre de la materia.
- 5) 263 es el tipo de template.
- 6) 100 es el peso global de T1, T2 y T3.
- 7) 0 es el peso del examen final.
- 8) 50 es el T1.
- 9) 50 es el T2.
- 10) 0 es el T3.
- 11) 0 es el P1
- 12) 0 es el P2.
- 13) 0 es el P3.

Los datos que empieza con la letra P son los parciales y los datos que empieza con la letra T son los trabajos.

- d) La estructura de la información del estudiante se divide en grupo normal y mesa como se muestra a continuación.

- 1) Lista de estudiantes de grupo normal tiene 10 columnas como se muestra en la siguiente linea del documento:

```
1,200206684,AGUILAR TORRICO RAMIRO,,,,,,
```

La estructura de la información del estudiante se explica de manera secuencial los cuales son: 1 es NRO, 200206684 es COD, AGUILAR TORRICO RAMIRO es NOMEST, vacio es 1ERPAR , vacio es 2DOPAR, vacio es PROMED, vacio es EXAFIN, vacio es 2da, vacio es NOTFIN y vacio es NOTCON.

- 2) Lista de estudiantes del grupo mesa tiene 7 columnas, según la siguiente linea del documento:

1,201209060,CALIZAYA MAMANI GAITH EFRAIN,20,,20,R

La lista de estudiantes se desarrollan de forma ordenada estas son: 1 es NRO, 200206684 es COD, CALIZAYA MAMANI GAITH EFRAIN es NOMEST, 20 es 1RAOPC, vacio es 2DAOPC, 20 es NOTFIN y vacio es NOTCON.

### 3.5.2. La estructura de la planilla de notas

La estructura de la planilla de notas tiene una estructura de etiqueta de inicio y final el cual divide a la información en una manera organizada para el reconocimiento del transcriptor.exe. A continuación se explica el orden de la estructura de la planilla de notas.

```
<pcd>(inicio documento)
<head>(inicio cabecera)
<info>(inicio Informacion)
</info> (fin Informacion)
<columna> (inicio columna)
<template> (inicio template)
<normal> (inicio normal)
</normal>(fin normal)
<me>(inicio mesa)
</me>(fin mesa)
</template>(fin template)
</columna>(fin columna)
<group>(inicio de grupo)
  <normal>(inicio de grupo normal)
  </normal>(fin de grupo normal)
  <me>(inicio de grupo mesa)
  </me>(fin de grupo mesa)
</group>(fin de grupo mesa)
</head>(fin de cabecera)
<body>(inicio cuerpo)
<gradelist>(inicio de lista)
<P1>(inicio materia tipo)
</P1>(fin materia tipo)
<P2>(inicio materia tipo)
</P2>(fin materia tipo)
</gradelist>(fin de lista)
</body>(fin cuerpo)
</pcd>(fin documento)
```

### **3.5.3. Las restricciones de la planilla de notas**

La planilla de notas es un archivo con extensión sis que tiene las siguientes restricción:

- a) No se debe cambiar la extensión.
- b) No se debe alterar la estructura del archivo en caso contrario no sera reconocido por la aplicación del transcriptor.exe.

En conclusión se puede notar que la funcionalidad de modificar la planilla de notas tiene 3 procesos en los cuales se utilizan las siguientes aplicaciones: la página del SAGAA<sup>7</sup> y el transcriptor.exe. Tomando en cuenta que cada uno tiene su respectiva restricción es por este motivo que se elige esta misma como una herramienta de prueba.

---

<sup>7</sup>SAGAA - Sistema de Apoyo a la Gestión Académica y Administrativa

## **Capítulo 4**

# **Proceso de desarrollo de servicio**

El desarrollo de software y la web en los años de 1990 mejoró el intercambio de información entre las organizaciones. A través del proceso de desarrollo de servicios web con el fin de apoyar en el intercambio de información entre las organizaciones.

### **4.1. El servicio web como componente de reutilización**

Según Somerville "los servicios web son un desarrollo natural de los componentes de software. Una diferencia fundamental entre un servicio y un componente de software, es que el servicio debe ser independiente y opera en la misma forma, sin importar su entorno de ejecución." Este servicio web define lo que necesita de otro servicio al establecer sus requerimientos en un mensaje y al enviar el mensaje con la especificación y los detalles de su interfaz. Los cuales se explican a continuación en los documentos de lenguaje de descripción de servicios web como WSDL<sup>1</sup> [1].

- En el documento de interface se define las operaciones que soportan el servicio y el formato de los mensajes que envían y reciben por parte del mismo servicio.
- En el documento de enlace mapea la interfaz abstracta a través de un conjunto concreto de protocolos y especifica los detalles técnicos de cómo comunicarse con el servicio.
- En el documento de ubicación se describe la ubicación de la implementación del servicio web en su punto final.

El desarrollo de WSDL expresa el servicio web en el formato XML.

### **4.2. La ingeniería de servicio**

La ingeniería de servicio web es el proceso de desarrollo de servicios para la reutilización en aplicaciones orientadas a servicios. La ingeniería de servicio garantiza que el servicio represente la abstracción y la reutilización, los cuales son útiles para otros sistemas. [1].

---

<sup>1</sup>WSDL-Lenguaje de descripción de servicios web

Según Somerville existen tres etapas lógicas en el proceso de ingeniería de servicio, el cual se representa a continuación en la figura 4.1 y la descripción de cada etapa.

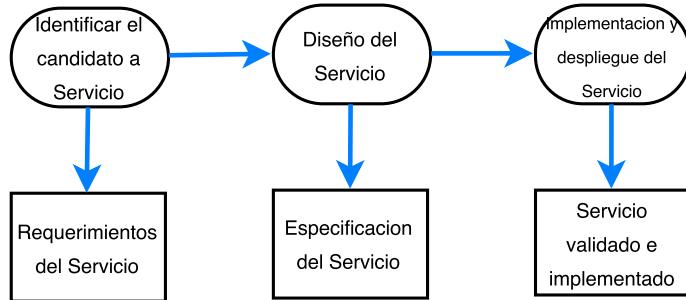


Figura 4.1: El proceso de ingeniería de servicio, adoptado para la realización del proyecto. El proceso de ingeniería de servicio [1]

La descripción de cada etapa lógica se explica en las siguientes etapas:

- **Identificación de candidatos a servicios:** la identificación orientada a servicio implica comprender y analizar los procesos empresariales de la organización para decidir cuáles servicios de reutilización podrían implementarse. Existen tres tipos de servicios para identificar el servicio los cuales son: servicio utilitario, servicio empresarial y servicio de coordinación o proceso. El presente proyecto es tipo empresarial porque se trata de servicios asociados con una función empresarial específica. Para verificar si el servicio es útil, se plantean las siguientes preguntas las cuales son:

**El cuestionario:** ayuda a reconocer el servicio y verificar el servicio.

- a) El servicio de entidades esta asociado a una entidad lógica que se utiliza en diferentes procesos empresariales?
- b) Que operaciones soporta la entidad ?
- c) Se trata de tareas que realizan diferentes personas en la organización?
- d) El servicio es independiente?
- e) El servicio tiene estado?
- f) El servicio pueden usar el cliente fuera de la organización?
- g) Diferentes usuarios tienen diferentes requerimientos?

Las respuestas ayudan a seleccionar y refinar abstracciones para implementar el servicio. El resultado del proceso de selección de servicios es un conjunto de servicios identificados y requerimientos funcionales del servicio, el cual define que debe hacer el servicio.

- **Diseño de interfaces del servicio:** el diseño de interface define las operaciones asociadas con el servicio y sus parámetros. Existen tres etapas en el diseño de la interfaz del servicio:

- En el diseño de interfaz lógica comienza con los requerimientos del servicio y define los nombres, parámetros de entrada salida, de operación y excepciones.

- En el diseño de mensajes se define la estructura de los mensajes que envían y reciben el servicio.
- Desarrollo de WSDL es el diseño lógico y de mensajes se traducen a una descripción de interfaz abstracta como se explica anteriormente el documento de interface, el documento de enlace y el documento de enlace.
- **Implementación y despliegue del servicio:** una vez concluido los candidatos y los diseños de interfaces, la etapa final del proceso de ingeniería de servicios es la implementación del servicio.

### 4.3. Servicios de sistemas heredados

Según Somerville "los sistemas heredados son sistemas de software antiguos que emplea una organización, en lo general dependen de tecnología obsoleta, pero son esenciales para la empresa." Los servicios que se desarrollan para acceder al sistema heredado son coherentes y soportan una sola área de su funcionalidad, los cuales se pueden representar graficamente en un diagrama de UML<sup>2</sup>. [1]

### 4.4. Desarrollo de software con servicios

Según Somerville el "desarrollo de software utiliza el servicio, el cual se basan en la idea de que usted combina y configura servicios para crear nuevos servicios compuestos. Estos pueden integrarse con una interfaz de usuario implementada en un navegador para crear una aplicación web".

El desarrollo de software con servicios es el proceso de diseñar nuevos servicios reutilizando los servicios existentes.

Según Somerville en la siguiente figura se muestra las etapas de construcción de servicio mediante composición:

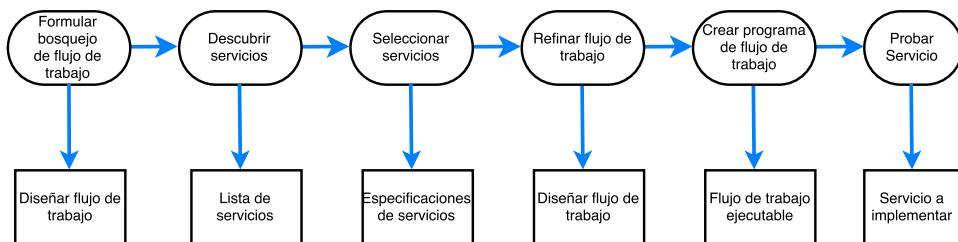


Figura 4.2: Etapas de la Construcción de servicio, adoptado para la realización del proyecto. Construcción de servicio [1]

- *Formular un bosquejo de flujo de trabajo* comienza con los requerimientos para crear un servicio compuesto, el cual es la base para un nuevo diseño.

<sup>2</sup>UML- Unidad de Lenguaje Modificado

- *El descubrimiento del servicio* se buscan los registros o catálogos de los servicios para descubrir cuales servicio existen, quién los proporciona y los detalles de la provisión del servicio.
- *Seleccionar posibles servicios* a partir del conjunto de posibles candidatos a servicio que se haya descubierto, se seleccionan los posibles servicios que puedan implementar actividades del flujo de trabajo.
- *Refinar el flujo de trabajo* sobre la base de la información acerca de los servicios que se selecciona y se refine el flujo de trabajo.
- *Crear un programa de flujo de trabajo* durante la etapa el diseño de flujo de trabajo abstracto se transforma en un programa ejecutable.
- *Prueba de servicio o aplicación terminada* es el proceso de probar el servicio terminado para realizar la prueba se utilizan los servicios externos.

Para concluir se explica el diseño y las pruebas del flujo de trabajo. Uno de los problemas es la reutilización dentro de la organización.

#### **4.4.1. Diseño e implementación del flujo de trabajo**

Según Somerville "diseño de flujo de trabajo implica analizar los procesos empresariales existentes o planeados para comprender las diferentes actividades que se realizan, estas intercambian información, luego se define el nuevo proceso empresarial en la notación de flujo de trabajo". Si los procesos empresariales tienen diferentes partes de una organización estas se pueden separar mediante carriles. Una vez concluido con el diseño final, el cual se debe convertirse en un programa ejecutable. [1]

- *Implementar los servicios* se debe tener en cuenta que los servicios son independientes del lenguaje de implementación.
- *Generar una versión ejecutable* del modelo de flujo de trabajo traducirlo a un código de manera manual o automática.

#### **4.4.2. Pruebas del servicio**

Las pruebas son importantes para los procesos de desarrollo de sistemas, estos muestran que un sistema cumple con los requerimientos funcionales y detectan defectos introducidos durante el proceso de desarrollo. [1]

Según Somerville "para comprender la implementación del servicio se debe presentar dificultades cuando se prueba el servicio y la combinación de los servicios".

Según Somerville a continuación los tipos de prueba del servicio.

- a) *Prueba de servicio o aplicación terminada* es el proceso de probar el servicio terminado, es complejo cuando se usa servicios externos.
- b) *Los servicios externos* están bajo el control de proveedor de servicio y no del usuario del servicio. El proveedor puede retirar los servicios, dicho problema se debe manejar como componentes para mantener las versiones.

- c) Los servicios a largo plazo la aplicación puede usar otro servicio.
- d) El comportamiento no funcional no depende solo de como se usa, por parte de la aplicación que se pone a prueba.
- e) El modelo de pago para los servicios se puede hacer que los servicios sean muy costosos. Se debe utilizar servicios que son gratuitos.

## 4.5. El diseño metodológico

El diseño metodológico para el presente proyecto se realizo en base a la puntos anteriores teniendo como resultado la siguiente figura 4.3 que representa la secuencia de procesos de desarrollo para este proyecto.

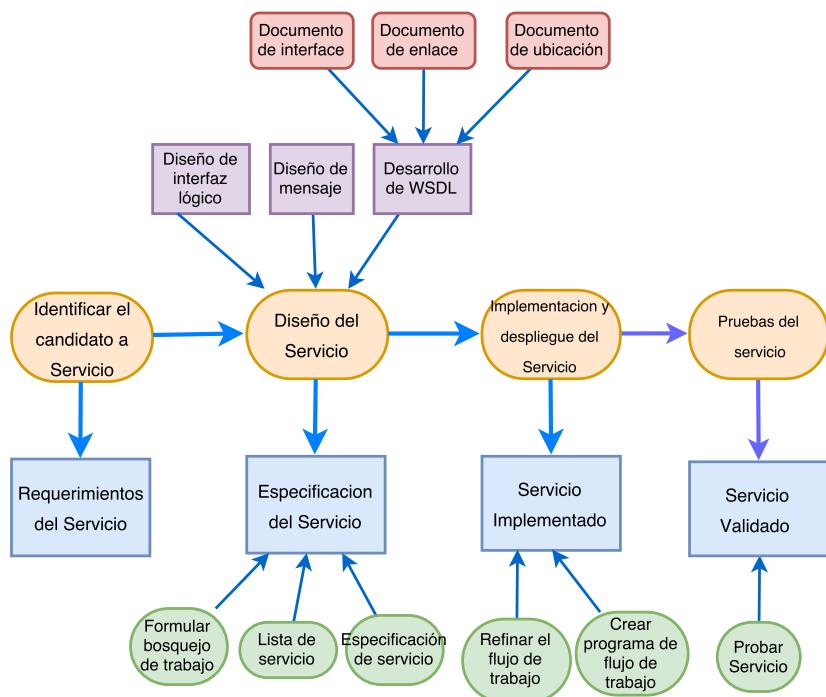


Figura 4.3: Secuencia de pasos para el proceso de ingeniería. Fuente: En base a la construcción de servicio de Somerville [1]

El presente proyecto es un sistemas heredado porque utiliza como antecedente a la página del SAGAA para el desarrollo del servicio web. Es por lo cual se ha incorporado las etapas de construcción de servicio mediante composición en el modelo metodológico.

## **4.6. Conclusiones**

La arquitectura de servicios web utiliza el desarrollo de componentes como base para la etapa de construcción del servicio web.

Los sistemas heredados utilizan un servicio web existente de una organización para crear un nuevo servicio web con la herramienta de diagrama de flujos de trabajo.

El diseño metodológico utiliza el diseño de servicio como base y complementa con la herramienta del desarrollo de software de servicio por componentes para la etapa de diseño del servicio con los diagrama de flujos.

## **Capítulo 5**

# **Diseño e implementación del servicio web**

El desarrollo del servicio web proporciona el análisis de las actividades y la comunicación de la página del SAGAA, a través de los diseños para ofrecer una explicación lógicamente coherencia e independiente.

### **5.1. Identificar el candidato a servicio**

Para identificar el candidato a servicio sin embargo la experiencia y la habilidad en el desarrollo de servicios web.

El presente proyecto tiene el objetivo de optimizar los procesos de llenar la planilla de notas en la página del SAGAA como se ha mencionado anteriormente en el capítulo 3. Para verificar si el servicio es útil se realiza el cuestionario, el cual se desarrolla en el anexo A. Despues de lo mencionado anteriormente se realiza la lista de requerimientos para el servicio web estos se mencionan a continuación.

#### **5.1.1. Requerimientos del servicio web**

Los requerimientos que se muestran a continuación son un conjunto de servicios los cuales se obtienen a partir de los procesos de llenar la planilla de notas como se ha explicado anteriormente en el capítulo 3.

- a) Descargar la planilla de notas según la gestión y la carrera correspondiente.
- b) Obtener datos de la página del SAGAA para la unidad de datos para compartir la información de la planilla de notas.
- c) Adjuntar la planilla de nota modificada en la página del SAGAA.

Una vez concluido con los requerimientos se avanza a la etapa de diseño del servicio web.

## 5.2. Diseño del servicio

El diseño del servicio describe la información al enviar los mensajes de entrada y salida de la página del SAGAA al realizar el proceso de llenar la planilla de notas. Las etapas del diseño de servicio son la: interfaz lógica, mensajes, el desarrollo de WSDL y la especificación del servicio. A continuación se explican cada etapa.

### 5.2.1. La etapa de diseño de interfaz lógica

El diseño de interfaz lógica es la comunicación entre el cliente y el servidor. El servidor almacena información de la página del SAGAA. Para explicar los procesos para llenar la planilla de notas.

- *El Diseño de interfaz lógica del proceso de descargar la planilla de notas* se representa en la siguiente figura 5.1 el cual es la secuencia de peticiones ordenadas, los parámetros de entrada y salida o mensajes y las direcciones de url.

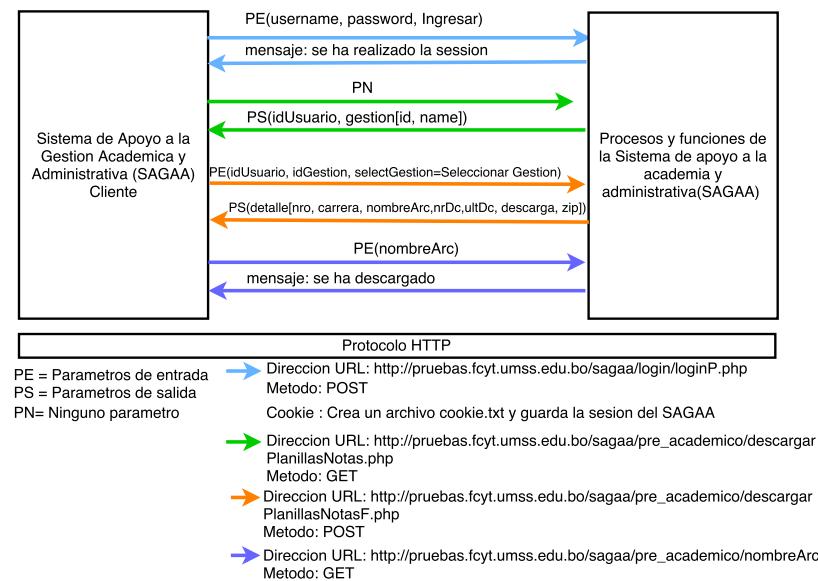


Figura 5.1: Diseño de interfaz lógico para descargar la planilla de nota,  
Fuente: Elaboración propia

- **Diseño de interfaz lógica de adjuntar la planilla de notas**, describe la secuencia de peticiones ordenadas y los parámetros de entrada y salida o mensajes y las direcciones de url. Como se muestra en la figura 5.2.

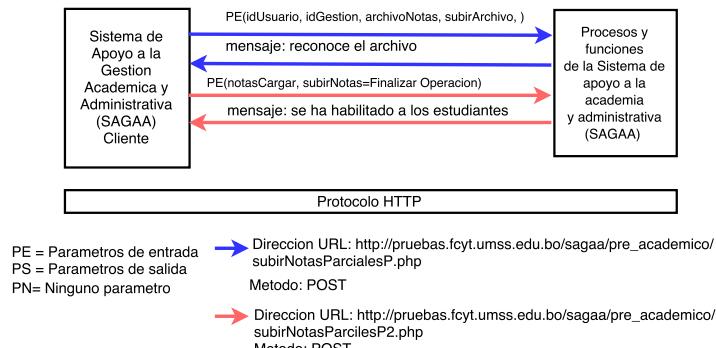


Figura 5.2: Diseño de interfaz lógico para adjuntar la planilla de notas,  
Fuente: Elaboración propia

### 5.2.2. Diseño de mensaje

El diseño de mensaje define la estructura de los mensajes que transcurren en la comunicación del cliente y servidor cuando se realiza una petición. Los mensajes ayudan a mostrar los estados de la petición si es correcta, incorrecta y excepciones. En la siguiente figura 5.3 se muestra los mensajes del proceso de llenar la planilla de notas.

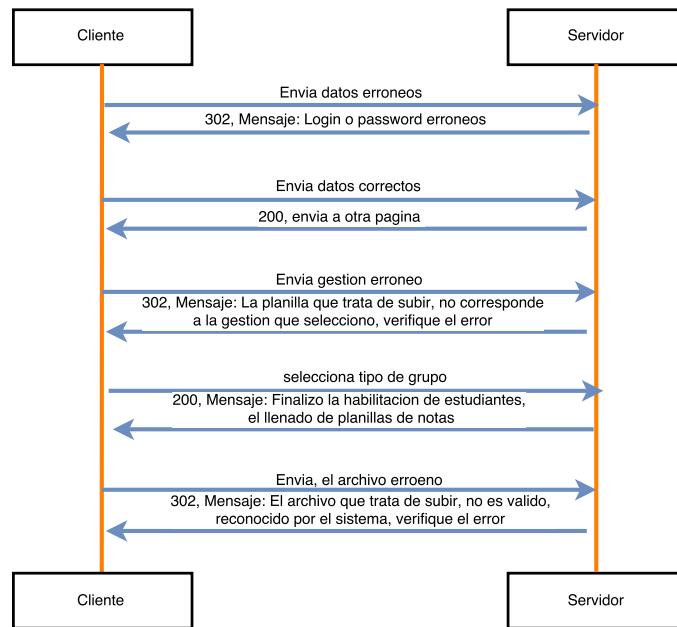


Figura 5.3: Diseño de mensaje de entradas, salidas y excepciones, Fuente:  
Elaboración propia

### 5.2.3. Desarrollo de WSDL

El desarrollo de WSDL no ha utilizado la estructura de desarrollo XML porque utiliza el protocolo REST para la implementación del servicio. El desarrollo de WSDL describe de la interfaz abstracta para llenar la planilla de notas como se explican a continuación en el documento de interface y el documento de enlace.

#### ■ Documento de interface

El documento de interface define las operaciones del proceso de llenar la planilla de notas lo cual se dividen en tres procesos los cuales son:

**Proceso 1:** Secuencia de procesos para descargar la planilla de notas en el formato .sis de la página del SAGAA. En la figura 5.4.

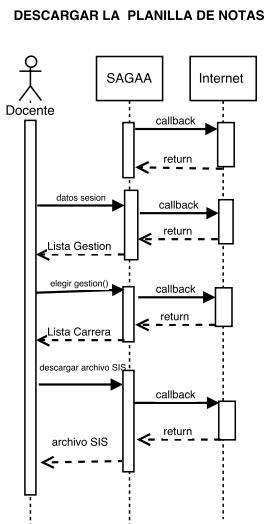


Figura 5.4: Proceso de descargar Planilla Notas. Figura: Elaboración propia

**Proceso 2:** En este proceso se muestran los pasos para modificar la planilla de notas. Se ha utilizado la aplicación del Transcriptor.exe para modificar la planilla de notas. Como se muestra en la siguiente figura 5.5.

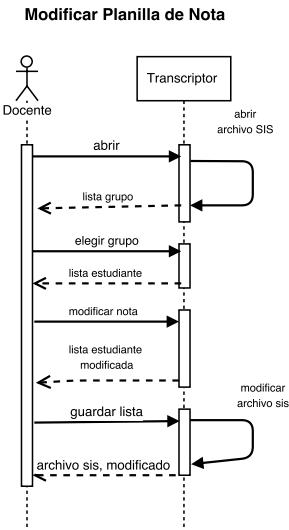


Figura 5.5: Proceso de modificar la Planilla de Notas, Fuente: Elaboración propia

**Proceso 3:** La ultima secuencia de pasos para adjuntar el archivo de planilla de notas en la página del SAGAA se muestra en la figura 5.6.

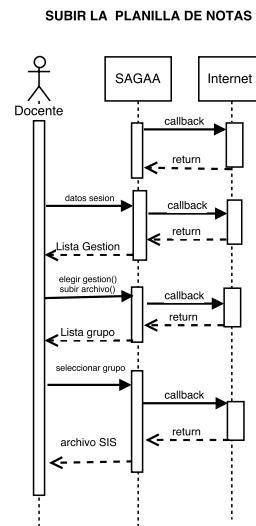


Figura 5.6: Proceso de subir a la página del SAGAA, Fuente: Elaboración propia

#### ■ Documento enlace

El documento de enlace es la interfaz abstracta de un conjunto concreto de protocolos los

cuales especifican los detalles técnicos de la comunicación entre la página del SAGAA y el servidor. El servidor es donde se almacena la información solicitada. La interfaz abstracta se representa en la siguiente figura 5.7.

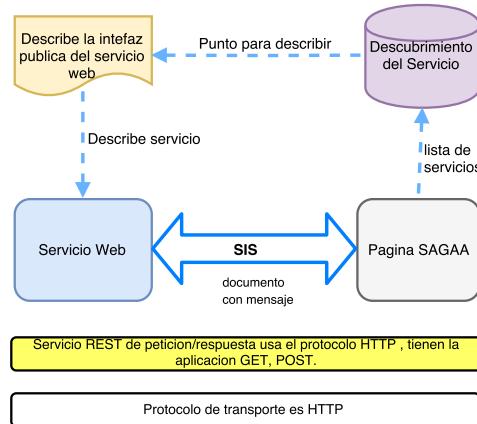


Figura 5.7: Protocolos de servicio, Fuente: Elaboración propia

#### 5.2.4. Especificación del servicio

En la etapa de especificación del servicio describe las construcciones de servicios las cuales son: formular el bosquejo de trabajo, lista de servicio y especificación de servicio.

- **Formular el bosquejo de flujo de trabajo**

A partir de los requerimientos se ha desarrollado un servicio compuesto el cual es la base para el nuevo diagrama de flujo de trabajo. Este servicio compuesto describe la secuencia de pasos de llenar la planilla de notas como se observa en la siguiente figura 5.8.

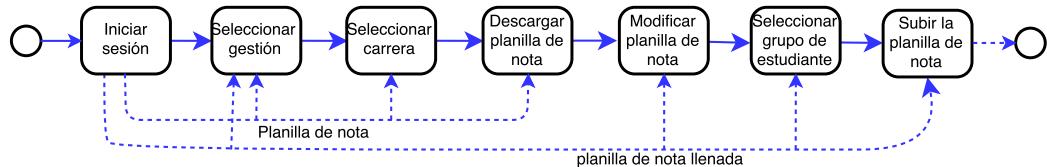


Figura 5.8: Diagrama de flujo de trabajo, Fuente: Elaboración propia

- **Descubrir el servicio**

Durante la etapa de descubrir el servicio se ha buscado el catálogo o registros para realizar el proceso de llenar la planilla de notas en la página del SAGAA el cual se ha mencionado anteriormente en el capítulo 3. A continuación la siguiente lista de servicio

son:

### **Lista de servicio**

- a) Realizar la sesión.
- b) Listar la gestión.
- c) Seleccionar la gestión
- d) Listar las carreras.
- e) Seleccionar carrera, para descargar la planilla de notas.
- f) Subir la planilla de nota modificado.

#### **■ Seleccionar el servicio**

A partir de la lista de servicios se han seleccionado los candidatos a servicios. Los cuales son los siguientes indices.

- a) Para descargar la planilla de notas se realizan la sesión del docente, seleccionar la gestión, seleccionar la carrera y por ultimo se descarga la planilla de nota.
- b) Modificar la planilla de notas es donde se guarda la planilla de datos localmente, reconoce la planilla de notas, y con la ayuda de la aplicación móvil mostrar y modificar la planilla de notas y enviar la planilla de notas modificada a la página del SAGAA.
- c) Para subir la planilla de notas primeramente se elige el grupo de materia, luego se adjunta la planilla y por ultimo se selecciona el tipo de grupo para publicar la planilla de notas en la página del SAGAA.

### **5.2.5. Implementación y despliegue del servicio**

Al concluir con los diseños de interface abstracta se realizan las etapas de refinar el flujo de trabajo y crear el programa de flujo de trabajo.

#### **■ Refinar el flujo de trabajo**

A partir de seleccionar el servicio se realiza el diseño del servicio. Cada diagrama mejora el detalle y la descripción abstracta de cada servicio seleccionado. Cada diagramas mejoran el detalle y la descripción abstracta de cada especificación del servicio seleccionado. A continuación se explican cada diagrama de flujo de trabajo.

- *Descarga la planilla de notas* es la secuencia de procesos para descargar la planilla de notas el cual se representa a través del diagrama de flujos de trabajo como se muestra en la siguiente figura 5.9.

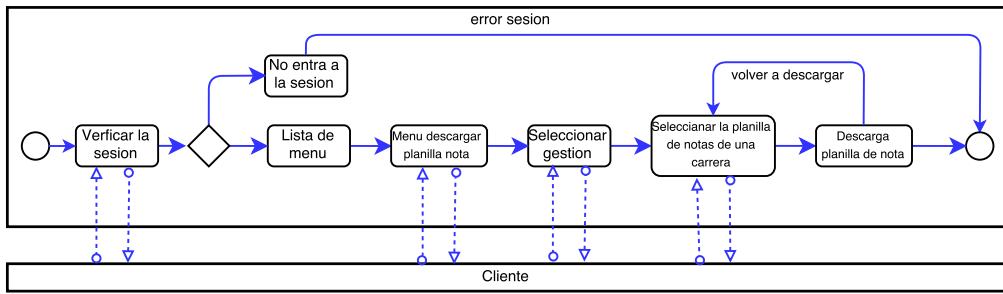


Figura 5.9: Diseño de flujo de trabajo de descarga de la planilla de notas,  
Fuente: Elaboración propia

- *Diseño de modificar la planilla de notas* utiliza la aplicación del Transcriptor.exe el cual tiene una secuencia de procesos. En la siguiente figura 5.10 el diagrama de flujo de trabajo. .

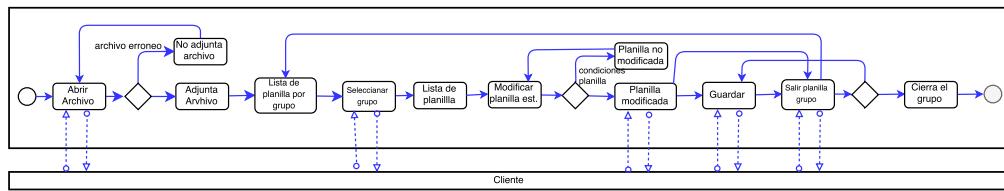


Figura 5.10: Diseño de flujo de trabajo de la descarga de la planilla de notas,  
Fuente: Elaboración propia

- *Diseño para adjuntar la planilla de notas* son pasos secuenciales para publicar la planilla de notas el cual se representa a través de la figura 5.11.

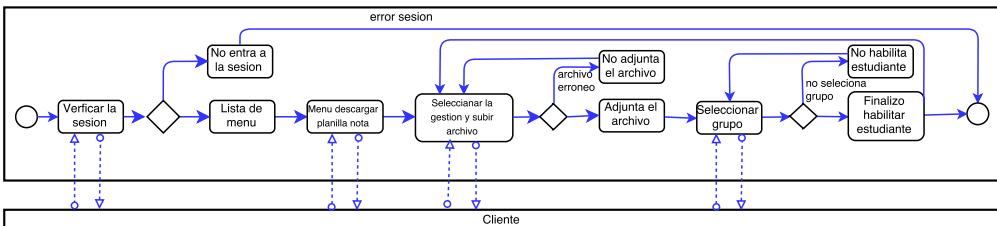


Figura 5.11: Diseño de flujo de trabajo de subir la planilla de notas, Fuente:  
Elaboración propia

- **Crear un programa de flujo de trabajo** el diagrama para crear un programa de flujo de trabajo es la unión de los servicios seleccionados anteriormente el diseño de flujo de trabajo de descargar, modificar y adjuntar. Se representa en la siguientes figuras 5.12 el crear un programa de flujo de trabajo.

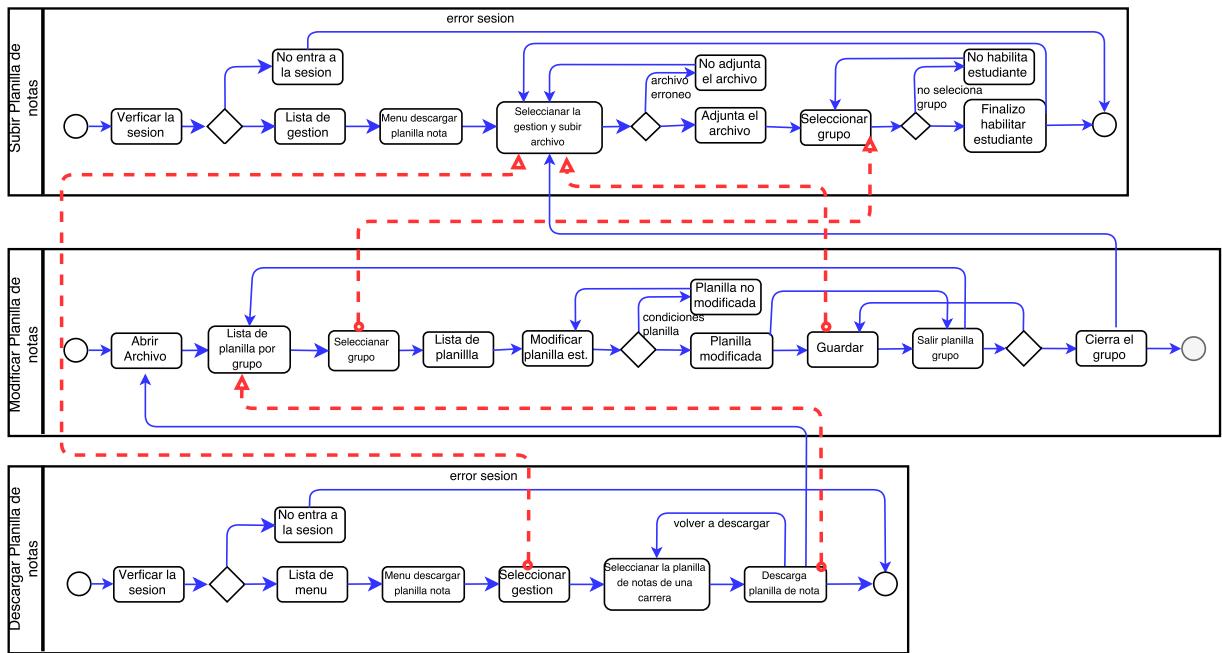


Figura 5.12: Diseño de flujo de trabajo planilla de notas, Fuente: Elaboración propia

En la figura 5.12 se representan los tres procesos, las flechas de punteada, es la comunicación entre procesos y parámetros que comparten entre ellos, para el desarrollo del servicio web.

### 5.3. Desarrollo de la implementación del servicio

Para el desarrollo de la implementación del servicio se realiza a partir de la figura 5.12, el cual describe la selección del servicio. La figura 5.13 representa el diagrama de flujo para la implementación del servicio.

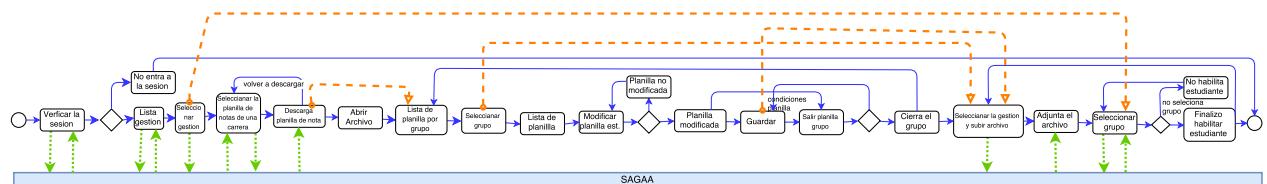


Figura 5.13: Diseño de flujo de trabajo planilla de notas para la implementación, Fuente: Elaboración propia

La figura 5.13 es la secuencia de proceso para el desarrollo del servicio web. Las líneas naranjas son parámetros que comparten entre actividades.

### 5.3.1. Herramientas y configuración para la implementación del servicio

Para el presente proyecto se ha utilizado el lenguaje de programación de javascript y express es un framework de Nodejs. El express es una extensión de conexión con muchos métodos de la arquitectura de REST y intercambio de información a su disposición.

- **Arquitectura del framework Express** express utiliza la arquitectura de modelo, vista y controlador definida como MVC.

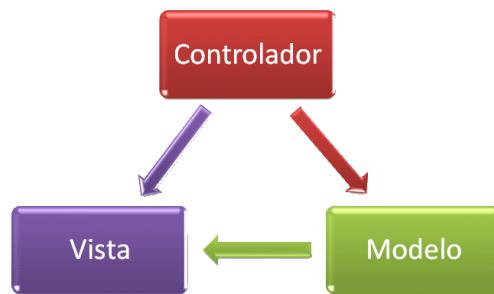


Figura 5.14: Arquitectura de Express, adoptado para la realización del proyecto. Arquitectura de Express de [3]

Para el presente proyecto se ha implementado la parte del modelo para explicar la comunicación del servicio web como se muestra en la arquitectura de Nodejs en la figura 5.15..

- **Arquitectura de Nodejs** utiliza la arquitectura orientado cliente/servidor.

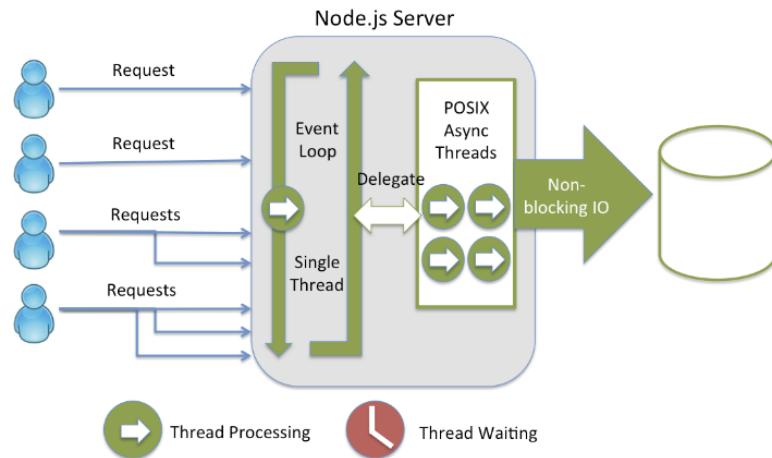


Figura 5.15: Arquitectura de NodeJs, adoptado para la realización del proyecto. Arquitectura de NodeJs de [4]

En la arquitectura de Nodejs, analiza las peticiones de entrada y salida.

### **5.3.2. Estructura del servicio web**

La estructura de servicio web utiliza la estructura de Ionic tal como se ha mencionado anteriormente en el presente proyecto y se aumenta el archivo server.js.

```
proyecto
|---package.json (paquetes de plugin de NodeJS)
|---plugins/ (Se guardan los plugins para el server)
|---www/ (Código fuente principal)
|---archivo/ (guarda los archivos)
    |---js/ ( El código, Javascript de la aplicación)
        |--server.js
```

### **5.3.3. Configuración del servicio web**

Previamente se debe instalar node y npm para comenzar a realizar los siguientes comandos para configurar el servidor:

```
$ npm install -g express (instalar globalmente express)
$ touch server.js (crear el archivo server.js)
$ node server.js (ejecuta el server.js)
$ npm install nombre_plugins -g (instalar plugins necesarios para el proyecto)
```

### **5.3.4. Aplicar el servicio web en Express**

Para la implementación el servicio web se ha utilizado la herramienta de Curl, Cherrie y la cuenta de docente el cual ha sido proporcionada por la UPSI de la FCYT. Se ha utilizado el diseño de flujo de trabajo de implementación de planilla de notas del diagrama 5.13 el cual tiene los siguientes servicios web:

- Verificar la sesión.
- Listar la gestión.
- Seleccionar la gestión y devuelve la lista de las carreras.
- Seleccionar la carrera de la planilla de notas para descargar planilla de notas.
- Filtrar y convertir la planilla de notas a unidades de datos de servicio json.
- La planilla de notas con la unidad de datos modificados.
- Crear una firma de clave para el servicio web.
- Seleccionar la gestión y adjuntar planilla de notas modificada.
- Crear unidad de datos json.
- Modificar los datos de la planilla de notas.
- Configurar la sesión de token.

- Depende del anterior item seleccionar grupo para habilitar estudiante.

Cada servicio web es muy importante para lograr el objetivo general y se utilizan los datos del diseño de interfaz lógico de la figura 5.1 y el diseño de interfaz lógica para adjuntar la planilla de notas de la figura 5.2 para realizar cada servicio web. Para mostar la implementación del servicio web se explican las partes importantes a continuación:

- *Servicio web verificar la sesión* para verificar la sesión tenemos los siguientes datos: la dirección de url, el método que utiliza, los parámetros y agentes de autorización. Finalmente envía los datos mencionados a otra página.

```
var curl = new Curl(),
url = 'http://pruebas.fcyt.umss.edu.bo/sagaa/login/loginP.php',
data = {
  'loginUsuario' : req.body.username,
  'claveUsuario' : req.body.password,
  'botonFormulario' : 'Ingresar'
};
curl.setOpt( Curl.option.URL, url );//direccion
curl.setOpt( Curl.option.FOLLOWLOCATION, true );//permite enviar a otra pagina
curl.setOpt( Curl.option.POST, 1 );//metodo que utiliza
curl.setOpt( Curl.option.POSTFIELDS, data );//parametros
curl.setOpt( Curl.option.HTTPHEADER, ['User-Agent: Mozilla/5.0',
  'Content-Type: application/x-www-form-urlencoded' ] );
// agentes de autorizacion
curl.setOpt( Curl.option.COOKIEFILE, 'pcb5r5tg8l1lbcjbqffr6hfm56' );
// codigo de cookie de otra sesion, para obtener la sesion
curl.setOpt( Curl.option.COOKIEJAR, 'cookie.txt' );
//guarda la cookie en el archivo txt
curl.on('end', function( statusCode, body, headers ) {
  .....
  //respuesta de la comunicacion con el servidor del SAGAA
});
.....
```

- *Servicio web seleccionar la gestión y adjuntar planilla de notas* para este servicio se conoce: la dirección de url, el método que utiliza, parámetros encriptados, el httpheader para los agentes de autorización y la ubicación de la planilla de notas para adjuntar.

```
.....
dirA = '/home/geovanna/appSAGAA/appSAGAA/www/archivo/',
//ubicacion de la planilla de notas modificada
nameD = nombreAr,
idUsuario = usuarioId;
idGestion = gestionId;
url = 'http://pruebas.fcyt.umss.edu.bo/sagaa/pre_academico/
subirNotasParcialesP.php';
curlS.setOpt( Curl.option.URL, url ); //direccion
```

```

curlS.setOpt( Curl.option.POST, 1); //metodo
curlS.setOpt( Curl.option.HTTPHEADER,['Connection: keep-alive',
'User-Agent: Mozilla/5.0','Content-Type: multipart/form-data;',
'Cache-Control: max-age=0', 'Accept-Encoding: gzip, deflate']);
\\agentes de autorizacion
curlS.setOpt( Curl.option.HTTPPOST, [{name: 'idUsuario',
contents: idUsuario }, {name: 'idGestion', contents: idGestion },
{ name: 'archivoNotas', file: dirA + nameD,
type: 'application/vnd.symbian.install'},
{ name: 'subirArchivo', contents: 'Subir Archivo'}]);
//estructura de parametros encriptados
curlS.setOpt( Curl.option.COOKIEFILE, 'cookie.txt');
//utiliza el cookie .txt para utilizar la sesion
.....

```

- *Servicio web la unidad de datos json* la planilla de notas se convierte en datos json.

```

var parseString = require( 'xml2js' ).parseString;
.....
app.post('/detalle', function(req, res){
.....
parseString( fileS( body ), function( error, result ){
    sisJson = result;
    ....});
.....
})

```

- **Modificar los datos de la planilla de notas** para modificar el archivo de planilla de notas primeramente buscar los dato modificado reemplazar el el archivo original.

```

\\ se busca del dato antiguo a partir del dato actual
function searchdato (pathlocal, arrayM){
    fs.readFileSync(pathlocal).toString().split('\n').forEach(recorriendo);
    function recorriendo(line, index, arr) {
        ....//verifica si son diferentes lo almacena en datoNuevo y datoAntiguo
        if(aux != arrayM[index]){
            .....
            if(arrayM[index] != '<normal>'){
                if(arrayM[index] != '<me>'){
                    datoNuevo.push(arrayM[index]);
                    datoAntiguo.push(line);
                }
            }
        .....
    }
}

```

- *Actualizar la planilla de datos*, se utiliza el plugin *glob* el cual busca linea por linea el dato antiguo en la planilla de nota y lo remplaza por un dato modificado.

```

function addFile(file, antiguo, nuevo){
.....

```

```

glob(file, function(err, files) {
  if (err) { throw err; }
  files.forEach(function(item, index, array) {
    replace({
      regex : antiguo,
      replacement : nuevo + '\r',
      paths : [item],
      recursive : true,
      silent : true
    });
    console.log('Remplaza por linea o  complete');
  });
  .....
}

```

- La configuración la sesión de token a partir de la sesión y el token, se obtine la autentificación para el servicio web.

```

var Base64 = require( 'js-base64' ).Base64;
var mySecretKey = "1234asdfLKKJH";
app.post('/datosU', function(req, res){
  .....
  var cred64 = Base64.encode( data );
  var token = jwt.sign( data, mySecretKey );
  .....
  return res.status( statusCode ).json( token );
  .....
}

```

Después de concluir, con la implementación del servicio web, se debe realizar las pruebas del servicio el cual se explica en el siguiente capítulo 6.

## **Capítulo 6**

# **Análisis y Diseño de la aplicación móvil**

En este capítulo se realiza la descripción para el implementación de la aplicación móvil desde los requerimientos, la creación de modulos del proyecto, el desarrollo diseño de interfaz y el diseño de la comunicación entre cliente y el servicio web.

### **6.1. Requerimientos**

La especificación, de los requerimientos o necesidades que se plantearon para realizar la aplicación móvil, se tiene a partir del requerimiento del servicio web en el capítulo 5 en la sección 5.1.1 y la recolección de datos en el capítulo 3, para cumplir los objetivos específicos en la sección 1.5.

- Realizar Diseño adaptativo.
- Realizar la sesión.
- Listar la gestión.
- Listar carreras.
- Mostrar los datos de la planilla de notas
- Modificar la planilla de notas.
- Almacenar la planilla de notas en el celular.
- Modificar planilla de notas sin internet.

Los requerimientos, se divide por modulos para el desarrollo del presente proyecto.

### **6.2. Desarrollo de módulos del proyecto**

Los requerimientos y procesos del proyecto, se divide en 4 módulos, los cuales son:

### **6.2.1. Lista de módulos del proyecto**

- a) Mostrar la planilla de notas.
- b) Modificar la planilla de notas.
- c) Obtener datos del servicio web.
- d) Guardar la planilla de notas en el servicio web.

### **6.2.2. Lista de submodulo del proyecto**

- a) Mostrar la planilla de notas.
  - Crear la sesión para mostrar la planilla de notas.
  - Seleccionar la planilla de notas, según la gestión.
  - Seleccionar la planilla de notas, según la carrera.
- b) Modificar la planilla de notas.
  - Mostrar la planilla de notas ordenado por grupo.
  - Modificar la planilla de notas por grupo,
- c) Obtener datos del servicio web.
  - Verificar la sincronización, con el servicio web.
  - Si no tiene sincronización con el servicio web, generar un repositorio de la planilla de notas local.
- d) Guardar la planilla de notas en el servicio web.
  - Enviar la planilla de notas, al servicio web.
  - Verificar la sincronización con el servicio web, para guardar la planilla de notas.

Una vez concluido con la lista de módulos, se determina el diseño de interfaz para la aplicación móvil.

## **6.3. Diseño de interfaz**

El diseño es la etapa de proceso, donde los modulos y submodulos se funcionan, y se interlanzan con la implementación.

Para realizar el diseño de interfaz, se utiliza los submódulos 6.2, donde se desarrolla el diseño de la interfaz para el cliente.

Para el presente proyecto, se realizaron 2 intentos de diseños, están representadas de forma secuencial para cumplir con los objetivos específicos 1.5 y se utiliza la herramienta de wireframe, las cuales son:

- a) La primera interfaz, esta orientado a la aplicación del Transcriptor.exe, el cual tiene algunos datos repetidos como ser el buscar red, no se toma en cuenta al servicio web y solo representa la etapa de modificar planilla de notas, se muestra en la figura 6.1.



Figura 6.1: Interfaz de Usuario primer intento, Fuente: Elaboración Propia

Es la primera presentación, se presento con datos estáticos.

- b) El segundo diseño, se ha modificado con las observaciones del primer intento, se ha tomado en cuenta el servicio web y no se repiten las plantillas, se muestra en la figura 6.2.

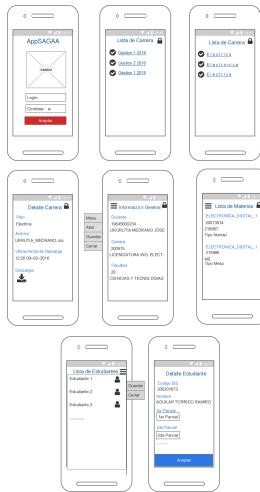


Figura 6.2: Interfaz de Usuario representado en mockups, Fuente: Elaboración Propia

Una vez concluido con el diseño de interfaz, conocido como el esquema de pantallas, se realiza el diseño de la comunicación entre cliente y el servicio web.

## 6.4. Diseño de la comunicación entre cliente y servicio web

El diseño de la comunicación es una representación abstracta de las solicitudes o peticiones que realiza la aplicación móvil(cliente) y las respuestas que el servicio web ofrece, como se muestra en la figura 6.3.

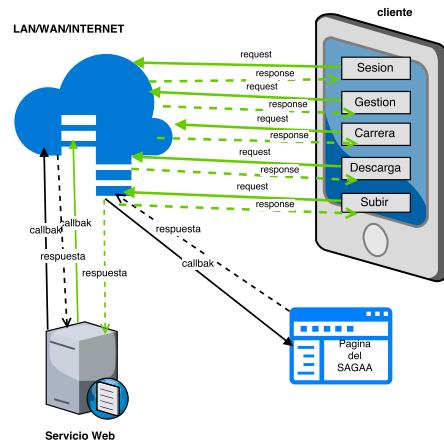


Figura 6.3: Diseño de comunicación entre el cliente y servicio web  
Fuente:  
Elaboración Propia

La implementación es el proceso de realizar el diseño como una aplicación móvil, que se explica en el siguiente capítulo 7.

## **Capítulo 7**

# **Implementación de la aplicación móvil**

En este capítulo se utilizan los datos obtenidos, del capítulo 6 y se explican las herramientas que se utilizan para la implementación de la aplicación móvil y el desarrollo del mismo.

### **7.1. Herramienta y configuración de la aplicación móvil**

Para el desarrollo de la aplicación móvil, se ha utilizado el framework ionic. Ionic es de código libre y se basa en librerías orientadas únicamente a aplicaciones de dispositivos móviles. También se enfocan a desarrollo de aplicaciones híbridas, construida con HTML3, CSS3 y Javascript se construye páginas web, se ejecutan dentro de un navegador, el cual aportan para ejecutar, en diferentes plataformas: android, iOS, windowsPhone, etc. Utiliza el framework de angular y es integrado con cordova, el cual permite el acceso a las características nativas del dispositivo. El framework de angular, utiliza la arquitectura de patrones de modelo, vistas y controladores, es la base para la arquitectura de ionic, se muestra en la figura 7.1.

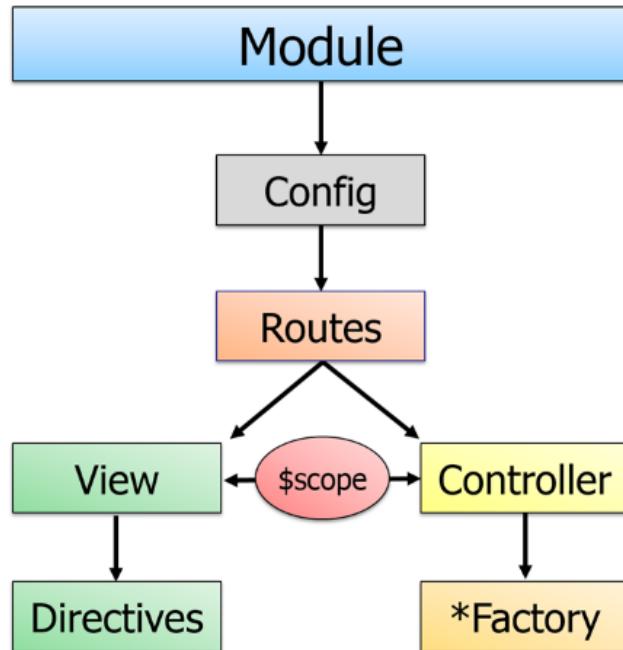


Figura 7.1: Arquitectura de Ionic, adoptado para la realización del proyecto  
Fuente: [5]

Para explicar la arquitectura de ionic se tiene diferentes tipos de componentes, que se explican a continuación:

- Config y routes:** es el archivo app.js, donde se realiza la configuración y las rutas, que permiten enlazar el controlador con la la interfaz de usuario correspondiente.
- Controller:** es el archivo controller.js, realiza la comunicación a través de la variable scope entre la interfaz de usuario y los servicios de los archivos services.js o factory.js.
- Directives:** es el archivo directive.js permite crear y usar componentes con aspectos y comportamiento.
- Factory:** son los archivo service.js y factory.js, son modelo de datos que ayudan a obtener los datos, del servicio web.
- View:** son los archivos html que contienen la descripción visual y obtiene los datos a mostrar el scope.

### 7.1.1. Estructura del Proyecto

El framework ionic, utiliza la estructura de modelo, vista y controlador del proyecto, el cual se genera automáticamente al momento de crear el proyecto en carpetas y archivos, para organizar el código en los siguientes archivos:

```

proyecto
|---bower.json (Lista de dependencias y paquetes de Bower)

```

```

|---.bowerrc
|---config.xml (Contiene la configuracion de la plataforma)
|---.editorconfig
|---.gitignore
|---gulpfile.js (Lista de tareas de Gulp)
|---hooks/ (Anade scripts que producen eventos)
|---ionic.project (Configuracion de Ionic)
|---package.json (Dependencias y paquetes de NodeJS)
|---platforms/ (Codigo plataformas para compilar)
|---plugins/ (Plugins o modulos de aplicacion)
|---resources/ (Recursos plataforma concreta)
|---scss/ ( Codigo SCSS compilado en www/css)
|---www/ (Codigo fuente principal)
    |--css/ (Estilos que se usa en la aplicacion)
    |--img/ (Imagenes de nuestro proyecto)
    |--index.html (Fichero principal, cargamos necesario)
    |--js/ (El codigo, Javascript de la aplicacion)
        |--app.js
        |--controllers.js
        |--directive.js
        |--factory.js
        |--filter.js
        |--service.js
    |--lib (Librerias, del codigo)
    |--templates/(Vistas de la aplicacion)

```

### 7.1.2. Configuración del proyecto

Para, el presente proyecto se utiliza, el interprete de linea de comando denominado (cli) de ionic, el cual, tiene los siguientes comandos:

```

$ ionic start appSAGAA sidemenu (crea proyecto)
$ ionic platform add android (adir la plataforma)
$ ionic build android (compilar el proyecto)
$ ionic run android (ejecutar el proyecto)
$ ionic serve (ejecutar, compila y muestra en el navegador )

```

Se crea automáticamente, la carpeta de appSAGAA con la estructura de la figura 7.1.1 y el menú sidemenu por defecto.

## 7.2. Herramientas Extras

Para el presente proyecto se ha utilizado algunas herramientas extras como ser: cordova, pouchDB, json web token y localstorage.

### 7.2.1. Framework cordova

Cordova es un framework de código libre, para desarrollo móvil, que nos permite usar estándares de tecnologías web como HTML5,CSS3 y Javascript. Se basa en los enlaces de API's compatibles para acceder a las capacidades del dispositivos como sensores, red, etc. En la figura 7.2 se muestra arquitectura de Cordova.

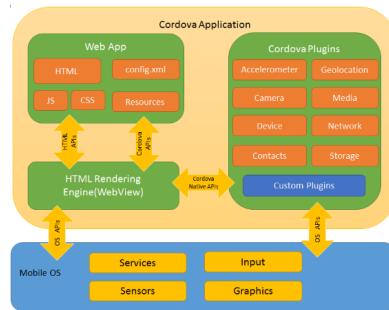


Figura 7.2: Arquitectura de Cordova, adoptado para la realización del proyecto Fuente: [6]

Para el presente proyecto se utiliza cordova, se agrega su librería al proyecto y se instala el plugin necesario.

### 7.2.2. PouchDB

Es una capa de otras base de datos, se almacenan en el navegador, permite guardar los datos a lado del cliente, es desarrollado en JavaScript. La estructura de pouchdb, se representa en la figura 7.3.

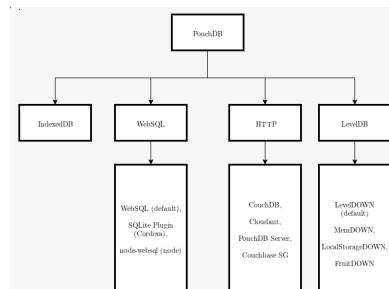


Figura 7.3: Adaptadores de PouchDB, adoptado para la realización del proyecto Fuente: [7]

Para el presente proyecto se ha utilizado la capa del adaptador de websql, se explica a continuación.

### 7.2.3. Base de datos websql

Es una herramienta para internet e intranets, facilita el acceso a base de datos relacionada con la web. Integra la tecnología del cliente y abre la sybase, el cual permite que los datos de la fuente, se han incorporadas dinámicamente en la página web, se representa en la figura 7.4.

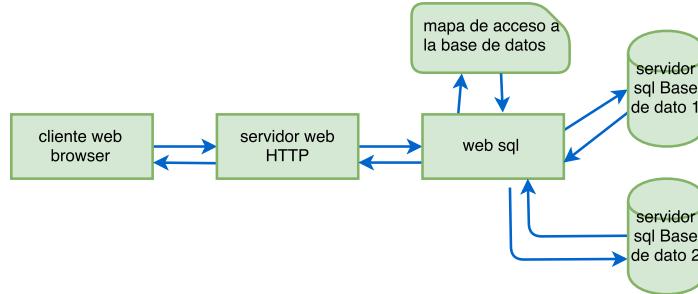


Figura 7.4: Base de datos websql, adoptado para la realización del proyecto

Fuente: [8]

### 7.2.4. Almacenamiento local

Es una propiedad de HTML5 web de almacenamiento, que permiten almacenar datos en nuestro navegador web denominada localstorage. Guarda la información que permanece almacenada por tiempo indefinido, sin importar que el navegador se cierre. Tiene las siguientes características, almacenar entre 5MB y 10MB de información, está almacenada en la computadora del cliente y no es enviado en cada petición del servidor, previene perdidas de información cuando se desconecta de la red y la información es guardada por dominio web [15].

### 7.2.5. Interceptor

El interceptor utiliza el servicio de *http*, el cual permite la comunicación con el servidor y captura cada petición y lo manipula a través del *httpProvider*, es el que registra el contenedor del arreglos y ofrece un servicio regulador.

### 7.2.6. Json web token

Json web token es un método abierto y estándar para representar las reclamaciones de forma segura, entre dos partes que comparten información y autentificación moderna, de móvil. El cual tiene una estructura representada en la figura 7.5.



Figura 7.5: Json web token, adoptado para la realización del proyecto  
Fuente: Elaboración propia

Estos son las herramientas extras, se utilizan para el presente proyecto, los cuales son compatibles con el framework ionic.

### 7.3. Implementación del Proyecto

Para la implementación del presente proyecto, se ha utilizado las herramientas 7.1 y las herramientas extras 7.2, se ha realizado según los modulos 6.2, el diseño de interfaz 6.3 que se han realizado en el capítulo 6 dividido en los siguientes intento.

#### 7.3.1. El primer intento o iteración la configuración para conectarse al servicio web

Configurar la aplicación móvil para solicitar la planilla de notas al servicio web, se explican los puntos importantes:

- La configuración para conectar la aplicación móvil al servicio web, se realizo en el archivo `www/js/fileFactory.js`:

```
//IP de la maquina donde se encuentra el server.js
var urlBase = 'http://172.20.10.3:8080';
var conf = {
headers : {
'Access-Control-Allow-Origin' : '*',
'Access-Control-Allow-Methods' : 'POST, GET, OPTIONS, PUT',
'Content-Type': 'application/jsonr',
'Accept': 'application/json'
}
};

};
```

- Solicitud de la planilla de notas al servicio web, se realizo en el archivo `www/js/fileFactory.js`

```
sisFactory.posDataDetalle = function(carrera){
.......
```

```

$http.post(urlBase+'/detalle', descargarD, {skipAuthorization : false}, conf).
success(function(data) {
.....
});

```

- c) Ordenar los datos del servicio web, se realizo en el archivo www/js/fileService.js.

```

divFile : function(data , template){
if(template == 'informacion'){
    return (((((data.pcd).head)[0]).info))[0];
.....
},
crearBDInf : function(array){
return newBD = {
    'fechaC' : array[0],
    'fecgaE' : array[1],
    'codDoc' : array[2],
    'nomApeDoc' : array [3],
.....
};
},

```

### 7.3.2. El segundo intento o iteración la sesión

Para asegurar la sesión el servicio web, envía en la cabecera la llave denominado jwt, el cual se explica la implementación en las partes importante a continuación:

- a) La configuración para enviar el jwt en la cabecera de la petición, se crea el interceptor, el cual se guarda localmente, se realizo en el archivo www/app.js

```

...
.config(function($stateProvider, $urlRouterProvider, $authProvider,
$httpProvider, jwtInterceptorProvider, jwtOptionsProvider) {
.....
jwtOptionsProvider.config({
    //IP de la maquina actual
    whiteListedDomains: ['localhost', '172.20.10.3']
    tokenGetter: function(options, jwtHelper){
        var token = localStorage.getItem('id_token');
    }});
    //metodo para enviar un json web token
    $httpProvider.interceptors.push('jwtInterceptor');
.....

```

### 7.3.3. El tercer intento o iteración trabajar sin el servicio web

Para modificar la planilla de notas, sin conexión a internet, anteriormente ha debido crear su sesión y descargar la planilla que utilizan localmente, se muestra los puntos mas importantes:

- a) Se utiliza el pouchdb y websql para crear, actualizar y modificar la base de datos, se realizo en el archivo www/js/fileService.js.

```
.service('SagaaService', function($q) {  
    ....  
    _db = new PouchDB('sagaa', { adapter: 'websql' },  
    { skip_setup: true });//crear base de datos  
    ....  
    return $q.when( _db.post( sagaa));//adir  
    ....  
    return $q.when(_db.put(sagaa));//actualizar  
    ....
```

- b) Se guardan las solicitudes al servicio web, que tienen algún problema con la conexión, se realizo en el archivo app.js.

```
....  
//crea un interceptor  
$httpProvider.interceptors.push('myInterceptor');  
....  
//crear un interceptor  
//crear metodo, para conocer la respuesta o la peticion  
var interceptor = function ($q, logHttp) {  
    return {  
        responseError: function(rejection) {  
            logHttp.push(rejection.config);  
            ....  
            return $q.reject(rejection);  
            .....}  
    }  
}
```

- c) Guardar y verificar las peticiones, se realizo en el fileService.js

```
.service('myInterceptor', function($q, $timeout, logHttp){  
    return {  
        'request': function(config){  
            ....//guardamos la peticion sin error  
            logHttp.push(config);  
            ....  
        'requestError': function(rejection){  
            //guarda la peticion con error,  
            window.localStorage.setItem('id_request', data);  
            ....// lo mismo realiza en la respuesta con error  
            .....  
    }}
```

```
.service('logHttp', function($q) {  
push: function(config) {  
requestsConfig = config;  
.....
```



## Capítulo 8

# Implementacion del diseño adaptativo para la aplicación móvil

Como se ha mencionado en el capítulo 1 en la sección del objetivo específico 1.5, se tiene en el presente capítulo como objetivo presentar el responsive de la aplicación móvil, a través de los componentes de CSS del framework ionic, que utiliza cuadrillas para proporcionar una solución a la variedad de resolución de diferentes dispositivos móviles inteligentes y tablets.

### 8.1. Diseño adaptativo o responsive

El diseño adaptativo es definido como el **Responsive Web Design (RWD)**, es una filosofía establecida por Steven Champeon (2003) que propone solucionar los problemas de diseño para los dispositivos o orientación. El contenido se debe adaptar a los dispositivos, creando una solución única. [14].

Para el presente proyecto se desarrolla una aplicación híbrida móvil para el sistema operativo de android, el cual tiene dispositivos en diferentes tamaño de resolución de pantalla, para la implementación se ha utilizado ionic.

### 8.2. Componentes de Ionic

Para el presente proyecto se ha utilizado el componente de Ionic, el cual tiene un sistema cuadrículada, con una mayoría de estándar de CSS y modulo flexible box de layout. Los dispositivos que son compatibles con el framework ionic soportan al **flexbox** que utiliza la herramienta grid. Las características de componentes de css se muestran en la figura 8.1.

Espacios iguales de columnas	En cada fila, agregar columnas reciben el mismo tamaño <div class="row"> <div class="col"></div> <div class="col"></div> </div>
Especifica el tamaño de columnas	En cada fila, agregar columnas reciben el mismo tamaño <div class="row"> <div class="col-50">El 50% de la fila</div> <div class="col-33">El 33.3% de la fila</div> </div>
Especifica el espacio de columnas	Se puede definir el tamaño de la columna <div class="row"> <div class="col col-offset-33">Espacio de 33</div> <div class="col"></div> </div>
Columnas alineada verticalmente	Cuando las filas se apilen cuando es pequeño la pantalla <div class="row"> <div class="col-top"></div> <div class="col-center"></div> <div class="col-bottom"></div> </div>
Responsive cuadrillas	Cuando las filas se apilen cuando es pequeño la pantalla <div class="row responsive-sm responsive-md responsive-lg"> <div class="col"></div> <div class="col"></div> <div class="col"></div> </div>

Figura 8.1: Componente de CSS de Ionic, Fuente: Página oficial

### 8.3. Implementación de flexbox

Para el presente proyecto se utiliza el flexbox, el cual se encarga de realizar los cortes de las cuadrillas es la siguiente linea de código.

```
<div class="row responsive-sm responsive-md">
//El responsive-sm reduce la fila para visualizar de forma horizontal para móvil
//El responsive-md se amplia la columna de forma vertical para la tablets
<div class="col col-33">
\\ El col-33 define el 33% de la resolucion de la pantalla
```

### 8.4. Visualización del contenido con flexbox

Para el presente proyecto, se aplica el componente grid responsive de ionic, el cual nos ayuda en definir las filas, columnas responsive grid donde la columna se adapta al área. En la siguiente figura se muestra la diferencia.

A screenshot of a mobile application interface on a tablet in portrait orientation. The screen displays several input fields and labels:

- Código SIS:** A text input field containing "200206684".
- Nombre:** A text input field containing "AGUILAR TORRICO RAMIRO".
- 1er Parcial:** A text input field with placeholder text "Primer Parcial" and a red error message "Introducir solo numero".
- 2do Parcial:** A text input field with placeholder text "Segundo Parcial" and a red error message "Introducir solo numero".
- PROMEDIO:** A text input field.
- Examen Final:** A text input field with placeholder text "Examen Final" and a red error message "Introducir solo numero".

Figura 8.2: Visualización de contenido vertical en la tablet, Fuente:  
Elaboración propia

En la figura 8.2 es una tablet con orientación vertical donde las imágenes se acomodan en forma vertical de manera secuencial.

A screenshot of a mobile application interface on a tablet in landscape orientation. The screen displays data in three columns:

Código SIS	Nombre	1er Parcial
200206684	AGUILAR TORRICO RAMIRO	Primer Parcial Introducir solo numero

2do Parcial	PROMEDIO	Examen Final
Segundo Parcial Introducir solo numero		Examen Final Introducir solo numero

2da Instancia	Nota Final	Nota Cont
2da Instancia Introducir solo numero		

**GUARDAR**

Figura 8.3: Visualización de contenido horizontal en la tablet, Fuente:  
Elaboración propia

En la figura 8.3 es la tablet en orientación horizontal y los datos se acomodan de 3 columnas debido a la resolución de la pantalla.



## **Capítulo 9**

# **Pruebas del servicio web a través de la aplicación móvil**

En esta capítulo 9, se trata de la etapa de prueba de servicio web, el cual es una combinación entre servicios. Es la etapa de verificar que nuestro servicio web, brinde servicio a la aplicación móvil. En el capítulo 6 la aplicación móvil realiza peticiones al servicio web y en el capítulo 5 el servicio web ofrece servicios de la página del SAGAA. Uniéndolos se empieza a validar el servicios web, se dividen a continuación en las siguientes partes:

### **9.1. Descargar la planilla de notas**

Para descargar la planilla de notas, primeramente se realizo la sesión del docente, listar la gestión, seleccionar la gestión, listar las carreras y selección de la carrera para descargar la planilla de notas. A continuación se muestra la validación de los servicios para descargar la planilla de notas.

#### **9.1.1. La sesión**

El desarrollo de la sesión, comienza con la aplicación móvil envía los datos al servicio web, el cual se encarga de enviar los parámetros, a la página del SAGAA. En la figura 9.1, es la aplicación móvil que enviando los datos al servicio web. En la figura 9.2, es la ejecución del servicio web, el cual muestra la conexión y la respuesta con la página de SAGAA y obtiene la respuesta *OK*.

```

> GET /sagaa/usuarios/ HTTP/1.1
Host: pruebas.fcyt.ums.edu.bo
Accept: */*
Cookie: PHPSESSID=6othb32s2ofgdsudo8ucpdjpi
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 200 OK
< Date: Fri, 18 Aug 2017 16:52:22 GMT
*> Server: Apache/2.2.3 (CentOS) is not blacklisted
*> Server: Apache/2.2.3 (CentOS)
*> X-Powered-By: PHP/5.3.3
*> Expires: Thu, 19 Nov 1981 08:52:00 GMT
*> Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
*> Pragma: no-cache
*> Connection: close
*> Transfer-Encoding: chunked
*> Content-Type: text/html; charset=UTF-8

```

Figura 9.1: La aplicación móvil solicita la sesión al servicio web, Fuente: Elaboración propia

Figura 9.2: Respuesta de la página del SAGAA al servicio web, Fuente: Elaboración propia

### 9.1.2. Seleccionar la carrera para descargar la planilla de notas retorna la planilla de notas

En el desarrollo para descargar la planilla de notas, primeramente se ha debido elegir una gestión, la carrera y la aplicación móvil lista el detalle de la carrera y elige la opción de descargar planilla de notas y guarda el archivo. En la figura 9.3, se muestra la aplicación móvil que solicita la descarga de la planilla de notas al servicio web, a través del ícono descargar. En la figura 9.4, el cual muestra la ejecución del servicio web, el cual muestra una conexión y respuesta con la página del SAGAA; se obtiene la respuesta **OK** y la descarga de la planilla de notas.

```

GET /sagaa/plan/ HTTP/1.1
Host: pruebas.fcyt.ums.edu.bo
Accept: */*
Cookie: PHPSESSID=6othb32s2ofgdsudo8ucpdjpi
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 200 OK
< Date: Fri, 18 Aug 2017 16:52:22 GMT
*> Server: Apache/2.2.3 (CentOS) is not blacklisted
*> Server: Apache/2.2.3 (CentOS)
*> X-Powered-By: PHP/5.3.3
*> Expires: Thu, 19 Nov 1981 08:52:00 GMT
*> Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
*> Pragma: no-cache
*> Connection: close
*> Transfer-Encoding: chunked
*> Content-Type: text/html; charset=UTF-8

```

Figura 9.3: Aplicación móvil selecciona la carrera para descargar la planilla de notas, Fuente: Elaboración propia

Figura 9.4: Petición descarga de la planilla de notas, Fuente: Elaboración propia

## 9.2. Modificar la planilla de notas

Para modificar la planilla de notas, se tiene que realizar la sección 9.1.1, se guarda la planilla de notas en el servidor local del servicio web, se filtran los datos, se convierten la planilla de notas a unidad de datos json y se envía a la aplicación móvil para mostrarlo. A continuación se muestra la validación de algunos servicios:

### **9.2.1. Filtrar y convierte la planilla de notas**

El desarrollo de filtrar, se obtiene los datos desde el archivo de planilla de notas, se elimina el inicio y fin para convertir a unidad de datos json. En la figura 9.5, se muestra y limpia los datos eliminando. En la figura 9.6 crea la unidad de datos json.



Figura 9.5: Aplicación móvil lista los datos de la planilla de notas, Fuente: Elaboración propia

```
-----tam data sis-----  
2370  
.am los divide por salto de linea:89  
.am2 sacando el inicio:88  
.am8 sacando el final87  
| ped: { \r\n, \r\n, \r\n, head: [ [Object] ], body: [ [Object] ] } }  
--Json-----  
| ped: { \r\n, \r\n, \r\n, head: [ [Object] ], body: [ [Object] ] } }  
| el file ha sido creado y llenando los datos
```

Figura 9.6: Envía los datos de la planilla de notas en Json, Fuente: Elaboración propia

### **9.2.2. Enviar la planilla de notas**

Después de descargar la planilla de notas, el servicio web entrega la planilla de notas, en unidad de datos json y la aplicación móvil lo muestra en las siguientes figuras. La figura 9.7 se muestra la información general, en la figura 9.8 se muestra la lista de grupos y en la figura 9.9 se muestran la lista de estudiantes.

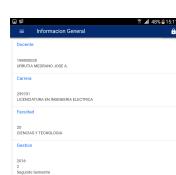


Figura 9.7: Información general,  
Fuente: Elaboración propia



Figura 9.8: La información de los grupos, Fuente: Elaboración propia

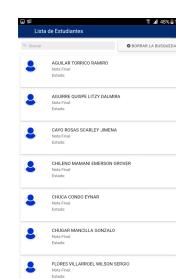


Figura 9.9: La lista de estudiantes, Fuente: Elaboración propia

### **9.2.3. Modificar la planilla de notas**

La aplicación móvil envía el acceso de datos json modificados al servicio web, el cual cambia, busca y reemplaza los datos que son compatibles con el archivo o unidad de datos de la planilla de notas que utiliza la página del SAGAA. En la figura 9.10 se muestra la unidad de datos json que recibe de la aplicación móvil y es modificado a la unidad de datos sis, el cual la página del SAGAA recibe. En la figura 9.11, el servicio web busca y reemplaza linea por linea el dato que ha sido modificada.

Figura 9.10: El servicio web, recibe la planilla de notas, Fuente: Elaboración propia

```
dato a ser analizado: <P1>-----size: 4
aux :<P1>-----size aux: 4
dato a ser analizado: 1, 200206684, AQU LAR TORR CO RAM RQ 60, 60, 60, 60, A-----size: 50
aux : 1, 200206684, AQU LAR TORR CO RAM RQ 60, 60, 60, 60, A-----size aux: 41
busca nuevo: 1, 200206684, AQU LAR TORR CO RAM RQ 60, 60, 60, 60, A
busca antiguo: 1, 200206684, AQU LAR TORR CO RAM RQ , .....
dato a ser analizado: 2, 2014040485, AQU RE QU SE LI IZY DALM RA, , , , , -----size: 47
aux : 2, 2014040485, AQU RE QU SE LI IZY DALM RA, , , , , -----size aux: 47
```

Figura 9.11: El servicio web, busca y reemplaza los datos modificados en la planilla de notas, Fuente: Elaboración propia

### **9.3. Adjuntar la planilla de notas**

Para adjuntar la planilla de notas, se elige la gestión y se adjunta el archivo de planilla de notas. A continuación se validan el servicios web:

### **9.3.1. Seleccionar la gestion adjuntar la planilla de notas**

Después de modificar la planilla de notas, el servicio web recibe los datos, elige la gestión , se adjunta el archivo de la planilla de notas y se elige el grupo que ha sido modificado. En la figura 9.12, se muestra la petición y la respuesta de la selección de gestión y el adjuntar la planilla de notas. En la figura 9.13, se muestra la petición y respuesta de la selección de grupo.

```
> POST /saga/pre_academico/subirNtasParcialesF.php HTTP/1.1
Host: prueba.fcyt.unms.edu.bo
Accept: */*
Cookie: PHPSESSID=D9n4cvuon0s27utfucflnl2qp3
Connection: keep-alive
User-Agent: Mozilla/5.0
Cache-Control: max-age=0
Accept-Encoding: gzip, deflate
Content-Length: 2959
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----09d5a0a5b1dd54e

< HTTP/1.1 100 Continue
< HTTP/1.1 302 Found
< Date: Fri, 18 Aug 2017 19:14:35 GMT
< Server: Apache/2.2.3 (QntOS) is not blacklisted
< Server: Apache/2.2.3 (QntOS)
< X-Powered-By: PHP/5.3.3
< Expires: Thu, 19 Nov 1981 08:52:00 GMT
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
< location: subirNtasParcialesF.php
< Content-Length: 0
< Connection: close
< Content-Type: text/html; charset=ISO-8859-1
<

* Closing connection 9
```

Figura 9.12: El servicio web, envia la gestion y adjunta la planilla de notas, Fuente: Elaboración propia

```
• Hostname was found in DNS cache
•   Trying 167.157.27.7...
Connected to pruebas.fcyt.ums.edu.bo (167.157.27.7) port 80 (HTTP/1.1)
POST /saga/pre_academico/subirNotasParcialesP2.php HTTP/1.1
Host: pruebas.fcyt.ums.edu.bo
Accept: */*
Cookie: PHPSESSID=04cdwvnu0ls27ufufcfl12gp3
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 58

• upload completely sent off: 58 out of 58 bytes
HTTP/1.1 200 OK
Date: Fri, 18 Aug 2017 19:45:35 GMT
Server: Apache/2.2.3 (CentOS)
Server: Apache/2.2.3 (QntrOS)
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 411
Connection: close
Content-Type: text/html; charset=UTF-8
;
• Closing connection 11
```

Figura 9.13: El servicio web, énvia el grupo que debe ser modificado. Fuente: Elaboración propia

### 9.3.2. Seleccionar el grupo para habilitar estudiante

Despues de adjuntar, se selecciona el grupo del archivo que ha sido modificado y se énvia a la página del SAGAA, el cùal responde con un mensaje *Finalizo la habilitaciòn de estudiantes y el cargado de planillas*. En la figura 9.14 se muestra la respuesta de la página del SAGAA.

```
----->
200
----->
<html>
<body>
<font color="#000099" size="5" face="Verdana, Arial, Helvetica, sans-serif">
<script language="JavaScript">
var timerId = setTimout("redireccionarPagina()", 3*1000);
function redireccionarPagina(){
    clearTimeout(timerId);
    window.location="subirNotasParciales.php";
}
</script>
Finalizó la habilitaciòn de estudiantes y el cargado de planillas.
```

Figura 9.14: En el servicio web, es la respuesta de la página del SAGAA,  
Fuente: Elaboraciòn propia



## **Capítulo 10**

# **Conclusiones**

Para desarrollado sincronización de información se puede optimizar con aplicaciones nativas.

Los servicios web es un estudio amplio, el cual se puede identificar diferentes servicios, con un análisis mas profundo, se puede buscar otros casos de estudios a parte de la página del SAGAA.

El proceso de desarrollo del servicio web, se optimiza el trabajo en la identificación de los servicios web.

Para elegir un caso de prueba para crear un servicio, se debe realizar previamente estudios de compartir información y verificar si la información a compartir puede ser pública. Bibliografía



# Bibliografía

- [1] Sommerville I. *Ingeniería de Software*. Pearson Education, Inc., 9a edición, Mexico, 2011.
- [2] Saucedo M. y Torralbo P. Sanz D. *Introducción a Android*. PhD thesis, Universidad Complutense, Madrid, 2014.
- [3] J. Lopez. Ingeniería de software. *Facultad de Informática - Universidad Politécnica de Valencia*, 2017.
- [4] R. Muñoz. Nodejs, javascript en servidor. *Desarrollo web - Cantabria TIC*, 2013.
- [5] J. Gallego. Ionic. *Libros de GitBooks*, 2016.
- [6] Cordova.org. Apache cordova. *La fundación de Software de Apache*, 2013.
- [7] D. Harvey. Pouchdb. *PouchDB.org*, 2014.
- [8] J. Meléndez y M. Tobar C. Cortez. Base de datos en internet. *Centro de Investigación de la Universidad Centroamericana José Simeón Cañas*, 2015.
- [9] Brea O.F. Historia de las web services. url`https://desarrolloweb.com/articulos/1883.php`, 2005.
- [10] Kindberg y Blair Colouris, Dollimore. *Sistemas Distribuidos*. Copyright., 5ta edición, 2012.
- [11] Navarro R. Rest vs web services. *Facultad de Informática - Universidad Politecnica de Valencia*, 2006.
- [12] El desarrollo de aplicaciones móviles nativas, web o híbridas. *Copyright IBM Corporation, Elaborado en los Estados Unidos de América* , 2012.
- [13] J. Morillo. *Introducción a los dispositivos móviles*. PhD thesis, Universidad Oberta de Catalunya, España, 2014.
- [14] A. Vega. *Responsive Web Design: Interfaces Web Adaptables al dispositivo empleado HTML5 Y CSS3*. PhD thesis, Escuela Politécnica Superior de Ingeniería Informática, 2012.
- [15] C. Cardenas. Localstorage. *Blog de Platzi*, 2015.



## **Apndice A**

# **Cuestionario de identificacion del servicio web**

Para seleccionar el servicio, se realizan las siguientes preguntas:

**1.- El servicio esta asociado, con una solo entidad lógica que se usa en diferentes procesos?**

**R.- Si, el servicio, se utiliza un solo archivo que tiene la extensión sis, el cual se encuentra en dos procesos en la página del SAGAA.**

**2.- Qué operaciones, se deben soportar que, se realizan usualmente sobre dicha entidad?**

**R.- La entidad el archivo con la extensión sis, se puede modificar a través de la aplicación del Transcriptor.exe.**

**3.- Se trata de tareas que realizan diferentes personas en la organización?**

**R.- Solo lo realiza un usuario el docente.**

**4.- El servicio es independiente?**

**R.- Si es independiente de las otras funciones.**

**5.- El servicio tiene estado?**

**R.- No tiene estado.**

**6.- El servicio pueden usarlo, clientes fuera de la organización?**

**R.- Si lo pueden usar los estudiante, pero no esta habilitado para ellos.**

**7.- Diferentes usuarios, tienen distintos requerimientos?**

**R.- Los estudiantes, desean ver sus notas, esto se puede ver como requerimientos no funcionales.**