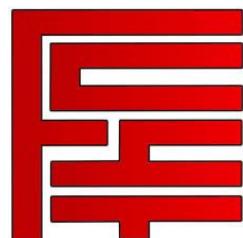




UNIVERSIDAD MAYOR DE SAN SIMON
FACULTAD DE CIENCIAS Y TECNOLOGIA
INGENIERIA INFORMATICA



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 por primeramente
Muchas gracias.

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

Geovanna Lizette Gil Terceros

RESUMEN

Al hablar de evolución normalmente se habla del cambio en un ente a medida que pasa el tiempo, éste proyecto se centra en ofrecer servicio a la aplicación móvil.

Palabras clave: *aplicación, servicio*

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 Provisin 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

Tabla de Contenido

Dedicatoria	I
Agradecimientos	III
Resumen	V
Lista de abreviaturas	VII
Lista de figuras	XVII
1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del Problema	1
1.2.1. Formulación del Problema	2
1.3. Objetivo general	2
1.4. Objetivos específicos	2
1.5. Justificación	2
1.6. Alcance	2
1.7. Diseño Metodológico	3
2. Herramienta para la Implementación	5
2.1. Sistema Distribuido	5
2.1.1. Computación cliente servidor	6
2.2. Servicio web	6
2.2.1. Los sistemas heredados	7
2.2.2. Los protocolos	8
2.3. Transferencia de estado representacional el REST	8
2.3.1. Hipertexto de Transferencia de Protocolo - Hypertext Transfer Protocol - HTTP	8
2.4. Unidades de datos de servicio	8
2.4.1. Notación de objeto de Javascript	9
2.4.2. Json web token	9
2.5. Aplicaciones Móviles	10
2.5.1. Aplicación Nativa	10
2.5.2. Aplicación web	10
2.5.3. Aplicación Híbrida	10
2.6. Dispositivos Móviles	10

2.6.1. Android	11
2.7. Diseño adaptativo	11
2.8. Plataformas de desarrollo	12
2.8.1. Entorno de desarrollo VIM	12
2.8.2. Navegador web - Chrome	12
2.9. Herramientas de Desarrollo	12
2.9.1. Framework ionic	12
2.9.2. Framework angularJS	12
2.9.3. Framework node.js	13
3. Análisis de la planilla de notas	15
3.1. Entrevista	15
3.2. La funcionalidad o operaciones para llenar la planilla de notas	16
3.3. El llenado de planilla de notas de la página del SAGAA	16
3.3.1. Pasos para descargar la planilla de notas	16
3.3.2. Pasos para subir planilla de notas	19
3.3.3. Las restricciones de la página SAGAA	20
3.4. Modificar la planilla de notas con la aplicación del transcriptor.exe	21
3.4.1. La funcionalidad de la aplicación del transcriptor	21
3.4.2. Detalle del Transcriptor	22
3.4.3. Restricciones del transcriptor	24
3.5. Planilla de notas	25
3.5.1. Información de la planilla de notas	25
3.5.2. Estructura de la planilla de notas	26
3.5.3. Restricciones de la planilla de notas	27
4. Proceso de desarrollo de servicio	29
4.1. Análisis del servicio web	29
4.2. Detalles de diseño	29
4.3. Ingeniería de servicio	30
4.3.1. Identificación de candidatos a servicio	31
4.4. Diseño de servicio web	31
4.5. Implementación del servicio web	32
4.6. Pruebas del servicio	32
5. Diseño y Implementación del Servicio Web	33
5.1. Análisis del servicio web	33
5.1.1. Documento de interfaz	33
5.1.2. Documento enlace	35
5.2. Identificar el servicio web	36
5.2.1. Cuestionario de servicio web	36
5.2.2. Requerimientos del servicio web	37
5.2.3. Formular el bosquejo de flujo de trabajo	37
5.2.4. Descubrimiento del servicio	38
5.2.5. Seleccionar los posibles servicios	38
5.2.6. Refinar el flujo de trabajo	39

5.3. Diseño del servicio web	40
5.3.1. Diseño de interfaz lógica	40
5.3.2. Diseño de mensajes	42
5.3.3. Crear un programa de flujo de trabajo	42
5.3.4. Diseño de unidad de datos de servicio	43
5.4. Implementación del servicio web	43
5.4.1. Herramientas y configuración del servicio web	44
5.4.2. Estructura del servicio web	45
5.4.3. Configuración del servicio web	45
5.4.4. Implementación del servicio web en Express	46
6. Análisis y Diseño de la aplicación móvil	51
6.1. Requerimientos	51
6.2. Módulos del proyecto	51
6.2.1. Los módulos para el proyecto	52
6.2.2. Dividiendo por sub modulo	52
6.3. Diseño de interfaz	52
6.4. Diseño de la comunicación entre cliente y servicio web	54
7. Implementación de la aplicación móvil	57
7.1. Herramienta y configuración de la aplicación móvil	57
7.1.1. Estructura del Proyecto	58
7.1.2. Configuración del proyecto	59
7.2. Herramientas Extras	59
7.2.1. Framework cordova	60
7.2.2. PouchDB	60
7.2.3. Base de datos websql	61
7.2.4. Almacenamiento local	62
7.2.5. Interceptor	62
7.2.6. Json web token	62
7.3. Implementación del Proyecto	62
7.3.1. El primer intento o iteración la configuración para conectarse al servicio web	63
7.3.2. El segundo intento o iteración la sesión	64
7.3.3. El tercer intento o iteración trabajar sin el servicio web	64
8. Implementacion del diseño adaptativo para la aplicación móvil	67
8.1. Diseño adaptativo o responsive	67
8.2. Componentes de Ionic	67
8.3. Implementación de flexbox	68
8.4. Visualización del contenido con flexbox	68
9. Pruebas del servicio web a través de la aplicación móvil	71
9.1. Descargar la planilla de notas	71
9.1.1. La sesión	71
9.1.2. Seleccionar la carrera para descargar la planilla de notas retorna la planilla de notas	72

9.2. Modificar la planilla de notas	73
9.2.1. Filtrar y convierte la planilla de notas	73
9.2.2. Enviar la planilla de notas	73
9.2.3. Modificar la planilla de notas	74
9.3. Adjuntar la planilla de notas	75
9.3.1. Seleccionar la gestion adjuntar la planilla de notas	75
9.3.2. Seleccionar el grupo para habilitar estudiante	75
10. Conclusiones	77
Bibliografía	77

Lista de figuras

1.1. Proceso de Ingeniería de software, adoptado para la realización del proyecto. Documentar el proceso del SOA de [1]	3
2.1. Computación cliente-servidor, adoptado para la realización del proyecto. Arquitectura cliente servidor de [1]	6
2.2. Arquitectura orientado a servicio, adoptado para la realización del proyecto. El objetivo del SOA de [1]	7
2.3. Servicio heredado, adoptado para la realización del proyecto. Representación de un servicio heredado de [1]	7
2.4. Representación de json por objeto, adoptado para la realización del proyecto. Estructura de json por objeto, de Figura: Elaboración propia	9
2.5. Representación del json por arreglo, adoptado para la realización del proyecto. Estructura de json por arreglo de Fuente: Elaboración propia	9
2.6. Representación del json web token, adoptada para la realización del proyecto de Fuente: Elaboración propia	9
2.7. Capas del sistema operativo de android, adoptado para la realización del proyecto. Herramienta de android de [2]	11
3.1. Diagrama de la funcionalidad de llenar y mostrar las notas, en la página del SAGAA Fuente: Elaboración propia	16
3.2. Inicio de página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT	17
3.3. Iniciar sesión en la página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT	17
3.4. Elegir menú Acádemico Pre-Grado Fuente: Proporcionado por la UPSI de la FCYT	17
3.5. Elegir opción descargar planilla de notas Fuente: Proporcionado por la UPSI de la FCYT	18
3.6. Opción de gestión, Fuente: Proporcionado por la UPSI de la FCYT	18
3.7. Elegir la descarga de una planilla de notas, Fuente: Proporcionado por la UPSI de la FCYT	19
3.8. Opción de habilitar estudiantes Fuente: Proporcionado por la UPSI de la FCYT	19
3.9. Opción de subir archivo seleccionado Fuente: Proporcionado por la UPSI de la FCYT	20
3.10. Paso 3: Subir planilla de notas por grupo Fuente: Proporcionado por la UPSI de la FCYT	20
3.11. Buscar planilla de notas del Transcriptor Fuente: Proporcionado por el MEMI	21
3.12. Buscar planilla de notas Fuente: Proporcionado por el MEMI	22
3.13. Lista de estudiantes, Fuente: Proporcionado por el MEMI	22
3.14. Detalle de la lista de grupo Fuente: Proporcionado por el MEMI	23
3.15. Detalle de la lista de estudiantes Fuente: Proporcionado por el MEMI	24

4.1. El proceso de ingeniería de servicio, adoptado para la realización del proyecto. El proceso de ingeniería de servicio de [1]	30
5.1. Proceso de descargar Planilla Notas. Figura: Elaboración propia	34
5.2. Proceso de modificar la Planilla de Notas, Fuente: Elaboración propia	34
5.3. Proceso de subir a la página del SAGAA, Fuente: Elaboración propia	35
5.4. Protocolos de servicio, Fuente: Elaboración propia	36
5.5. Diagrama de flujo de trabajo, Fuente: Elaboración propia	38
5.6. Diseño de flujo de trabajo de la descarga de la planilla de notas, Fuente: Elaboración propia	39
5.7. Diseño de flujo de trabajo de la descarga de la planilla de notas, Fuente: Elaboración propia	39
5.8. Diseño de flujo de trabajo de subir la planilla de notas, Fuente: Elaboración propia . .	40
5.9. Diseño de interfaz lógico para descargar la planilla de nota, Fuente: Elaboración propia . .	41
5.10. Diseño de interfaz lógico para subir la planilla de nota, Fuente: Elaboración propia . .	41
5.11. Diseño de mensaje de entradas, salidas y excepciones, Fuente: Elaboración propia . .	42
5.12. Diseño de flujo de trabajo planilla de notas, Fuente: Elaboración propia	43
5.13. Diseño de flujo de trabajo planilla de notas para la implementación, Fuente: Elaboración propia	44
5.14. Arquitectura de Express, adoptado para la realización del proyecto. Arquitectura de Express de [3]	44
5.15. Arquitectura de NodeJs, adoptado para la realización del proyecto. Arquitectura de NodeJs de [4]	45
6.1. Interfaz de Usuario primer intento, Fuente: Elaboración Propia	53
6.2. Interfaz de Usuario representado en mockups, Fuente: Elaboración Propia	54
6.3. Diseño de comunicación entre el cliente y servicio web Fuente: Elaboración Propia . .	55
7.1. Arquitectura de Ionic, adoptado para la realización del proyecto Fuente: [5]	58
7.2. Arquitectura de Cordova, adoptado para la realización del proyecto Fuente: [6]	60
7.3. Adaptadores de PouchDB, adoptado para la realización del proyecto Fuente: [7]	61
7.4. Base de datos websql, adoptado para la realización del proyecto Fuente: [8]	61
7.5. Json web token, adoptado para la realización del proyecto Fuente: Elaboración propia	62
8.1. Componente de CSS de Ionic, Fuente: Página oficial	68
8.2. Visualizacion de contenido vertical en la tablet, Fuente: Elaboración propia	69
8.3. Visualización de contenido horizontal en la tablet, Fuente: Elaboración propia	70
9.1. La aplicación móvil solicita la sesión al servicio web, Fuente: Elaboración propia	72
9.2. Respuesta de la página del SAGAA al servicio web, Fuente: Elaboración propia	72
9.3. Aplicación móvil selecciona la carrera para descargar la planilla de notas, Fuente: Elaboración propia	72
9.4. Peticion descarga de la planilla de notas, Fuente: Elaboración propia	72
9.5. Aplicación móvil lista los datos de la planilla de notas, Fuente: Elaboración propia . .	73
9.6. Envia los datos de la planilla de notas en Json, Fuente: Elaboración propia	73
9.7. Información general, Fuente: Elaboración propia	74
9.8. La información de los grupos, Fuente: Elaboración propia	74

9.9. La lista de estudiantes, Fuente: Elaboración propia	74
9.10. El servicio web, recibe la planilla de notas, Fuente: Elaboración propia	75
9.11. El servicio web, busca y reemplaza los datos modificado en la planilla de notas, Fuente: Elaboración propia	75
9.12. El servicio web, envia la gestion y adjunta la planilla de notas, Fuente: Elaboración propia	75
9.13. El servicio web, énvia el grupo que debe ser modificado, Fuente: Elaboración propia .	75
9.14. En el servicio web, es la respuesta de la página del SAGAA, Fuente: Elaboración propia	76

Capítulo 1

Introducción

En la Facultad de Ciencias y Tecnología de la Universidad Mayor de San Simón tiene como meta fundamental el de contribuir en la formación profesional así también se encarga de la parte administrativa de los estudiantes, docentes y postulantes a través de la página del SAGAA y el sistema WEBSISS. Sin embargo, algunas de estas páginas web no están adecuadas a dispositivos actuales, por ello se genera una insatisfacción en la interacción con la página del SAGAA. Para ello se pretende proveer un servicio de la página del SAGAA a través de una aplicación móvil, para el mejoramiento de la disponibilidad de información. Por lo cual se tendrá una aplicación móvil donde se muestre el servicios de la página del SAGAA para los dispositivos actuales, donde los docentes se beneficiaran a raíz de las necesidades. Como consecuencia a las necesidades debemos mejorar la usabilidad de páginas web existente y adecuar a las nuevas tecnologías de dispositivos móviles actuales.

1.1. Antecedentes

Actualmente el SAGAA es utilizada por los docentes, estudiantes de pregrado y los estudiantes de postulantes para noticias, avisos, subir, publicar y bajar notas. También se tiene soporte técnico para la página del SAGAA. Esto nos ayuda a tener una mejor accesibilidad para estudiar su funcionalidad y crear una aplicación accesible para los usuarios en dispositivos actuales. Difundir los servicios que ofrece la página del SAGAA a los usuarios que les gustan los dispositivos actuales y así proponer una Aplicación Móvil.

1.2. Descripción del Problema

La Facultad de Ciencias y Tecnología apoya en la formación de profesionales el cual cuenta con un plantel docente, personal administrativo, estudiantes de pregrado y estudiantes de postulantes y otros. Los docentes para llenar la calificación de los estudiantes en las planillas tienen una secuencias de pasos que se realizan, primeramente descargan la planilla de los estudiantes en el formato .sis, luego utilizan la aplicación del Transcriptor.exe para abrir el documento, insertan notas de los estudiantes. Para finalizar publican el documento .sis en la página del SAGAA, el cual es difícil de manejar desde un dispositivo móvil y no se pueda utilizar si no se tiene Internet. Así que la mayoría de las páginas web no se encuentran adaptadas para dispositivos móviles. Es por el cual se tiene la

necesidad de crear una aplicación móvil que ofrece servicios a la página del SAGAA accediendo al manejo de datos de manera local y ofrecer una mejora interacción con el usuario. Esto contribuirá con la accesibilidad de la página del SAGAA .Es por eso que actualmente los celulares ya no son simples teléfonos, sino computadoras de bolsillo que hacen multitareas.

1.2.1. Formulación del Problema

Cómo mejorar la página del SAGAA, para que tengan un mejor usabilidad e interacción con los dispositivos actuales?

1.3. 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.

1.4. Objetivos específicos

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

1.5. Justificación

Debido que la página del SAGAA no tiene un adecuado diseño para interactuar con los dispositivos actuales, se contribuirá con el proceso del avance tecnológico para las páginas Web. Nace el incentivo para llevar a cabo el presente proyecto que pretende crear un servicio 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.

1.6. Alcance

Se pretende realizar el proyecto para mejorar la usabilidad 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.

1.7. Diseño Metodológico

El presente proyecto fue realizado, en base al servicio orientado a arquitectura denominado SOA según el libro de [1] que se muestra en la figura 1.1. Normalmente este método se usa para crear servicio web de páginas que tienen tecnología obsoleta o tenga una mayor accesibilidad de la información y se puede integrar con otras aplicaciones como ser aplicación móvil, etc.

La figura 1.1, representa la secuencia de proceso de desarrollo para el presente proyecto.

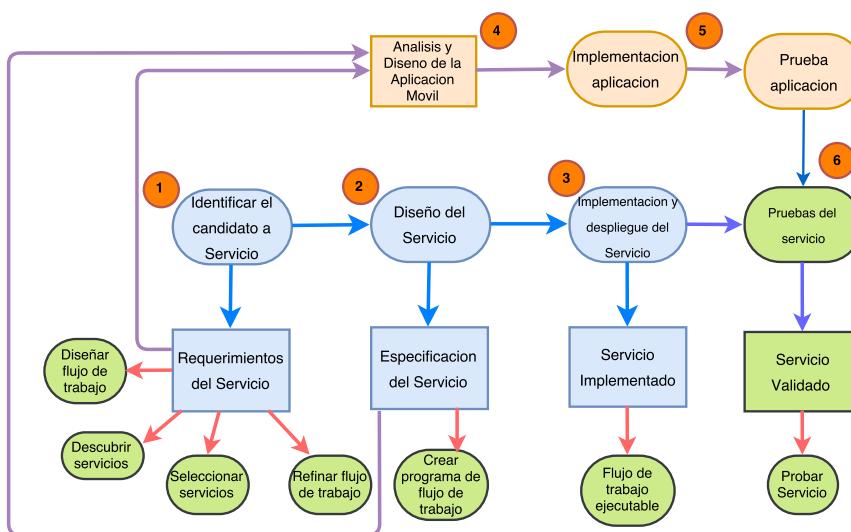


Figura 1.1: Proceso de Ingeniería de software, adoptado para la realización del proyecto. Documentar el proceso del SOA de [1]

En el capítulo 4 se explica el proceso de ingeniería de software del SOA y el capítulo 2, se explican las herramientas para cumplir con los objetivos específicos 1.4 del presente proyecto.

En el capítulo 3 se trata de explicar los detalles de la planilla de notas y procesos para el llenando para cumplir con el objetivo 1, la recolección de datos.

En el capítulo 5 se explica el análisis, diseño e implementación del servicio web, como se muestra en la figura 1.1 en el número 1 - 3 para cumplir con el objetivo 4.

En el capítulo 6, se trata del análisis y diseño de la aplicación móvil, como se muestra en la figura 1.1 en el número 4, para cumplir con el objetivo 4 y en parte al objetivo 1 del cual se tiene a continuación el capítulo 7, en donde se realiza la implementación de la aplicación móvil, como se muestra en la figura 1.1 en el número 5 cumpliendo con el objetivo 1 y 3.

En el capítulo 9 se trata de la prueba de la aplicación móvil y el servicio web, como se muestra en la figura 1.1 en el número 6, para cumplir con el objetivo general 1.3 de proveer el servicio.

del SAGAA, a través de una aplicación móvil.

En el capítulo 8 es el diseño adaptativo de la aplicación móvil que responde al objetivo 2.

Capítulo 2

Herramienta para la Implementación

En este capítulo se muestra la definición de sistema distribuido, donde el componente del sistema, son servicios independientes denominados arquitectura orientada a servicio web abreviado como SOA. En donde se define los tipo de protocolo estándar que se utilizan en el servicio web. Para validar el servicio web, se explica la comunicación entre el cliente servidor, después los tipos de aplicaciones móvil y la usabilidad .Por ultimo explicamos las herramientas para la implementación de la aplicación móvil y el servicio web.

2.1. Sistema Distribuido

Se define al sistema distribuido como un conjunto de computadoras independientes, que se muestran para el usuario como un solo sistema [9], los diferentes dispositivos tanto en hardware, como en software, se comunican en red para coordinar sus acciones mediante el envío de mensaje.

Según [9] identifico las siguientes ventajas para utilizar un sistema distribuido enfocado al desarrollo del sistema :

- a) **Intercambio de recursos:** un sistema distribuido permite compartir hardware y software de recursos, están asociadas a una computadora en red.
- b) **Aperturas distribuidas:** los sistemas son normalmente sistemas abiertos, están diseñado alrededor del estándar de protocolos que permite equipamiento y software desde diferentes vendedores para ser combinado.
- c) **Concurrencia:** en sistemas distribuidos, muchos procesos podrían operar al mismo tiempo en separadas computadoras sobre la red. Estos procesos podrían comunicarse con otros, durante su normal operación.
- d) **Escalabilidad:** en principio los sistemas distribuidos son escalables, la capacidad del sistema puede ser incrementado para añadir nuevos recursos para llevar al frente nuevas demandas en el sistema.
- e) **Tolerancia a fallas:** la capacidad de muchas computadoras y el potencial para replicar información significa que el sistema distribuido puede ser tolerante de algunos hardware y las características de software.

En sistema distribuido, la computación cliente servidor es el responsable de garantizar que los mensajes se transmitan de manera adecuada.

2.1.1. Computación cliente servidor

Los sistemas distribuidos, se conectan a Internet es definido como sistemas cliente servidor, el usuario o cliente interactúa con un programa, que se ejecuta en su computadora local, está interactúa con un programa que se ejecuta en una computadora remota, el cual proporciona servicio [1].

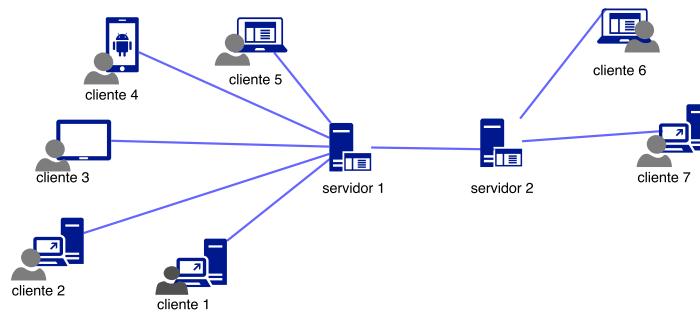


Figura 2.1: Computación cliente-servidor, adoptado para la realización del proyecto. Arquitectura cliente servidor de [1]

La arquitectura de cliente servidor, destaca al software como un servicio,(SaaS, Software as a Service - Software como un Servicio), que toma en cuenta la ampliación, de agregar servidores, para aumentar el numero de transacciones, en donde nace el servicio web.

2.2. Servicio web

El servicio web se define como un acto o función ofrecidos por una parte a otro, aunque el proceso puede asociarse a otro proceso físico, el objetivo del proyecto es hacer una representación estándar, para que la información pueda usar en otros programas. [1]. Resumen de la idea del SOA en la figura 2.2.

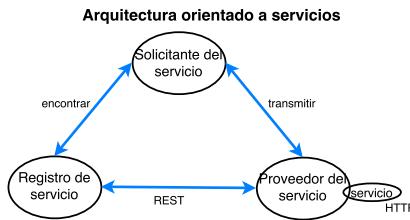


Figura 2.2: Arquitectura orientado a servicio, adoptado para la realización del proyecto. El objetivo del SOA de [1]

Cuando se habla de un servicio web, se debe analizar las características del servicio web según [1] 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 el servicio a implementarse, y las interfaces del servicio, son definidos como sistemas heredados.

2.2.1. Los sistemas heredados

Los sistemas heredados, son sistemas de software antiguos que emplean una organización, esto significa que es posible acceder a la funcionalidad del sistema mediante nuevas aplicaciones, que brinden acceso a las funciones y datos de un sistema y utilizar nuevas tecnologías [1]. Se muestra en la figura 2.3

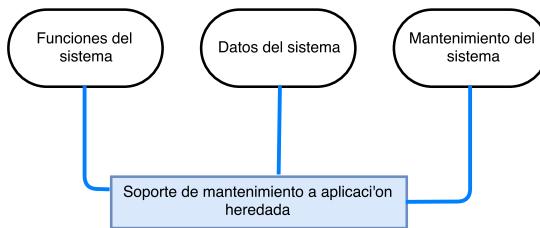


Figura 2.3: Servicio heredado, adoptado para la realización del proyecto. Representación de un servicio heredado de [1]

Los servicios web para la comunicación, utilizan los protocolos que se explican a continuación.

2.2.2. Los protocolos

Los servicios web, nos apoyan a trabajar en la comunicación, para proporcionar servicios integrales de comunicación de red. El cual, cubren todos los aspectos del SOA, desde los mecanismos básicos, para el intercambio de información con los protocolos estándares y no estándar de información de lenguaje, los cuales son: SOAP, WSDL, WS-BPEL, REST. Para el presente proyecto se ha utilizado el protocolo no estandar la arquitectura REST.

2.3. Transferencia de estado representacional el REST

Es un estilo de arquitectura de software, que 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. Aclarando que REST no es estándar, es un estilo de arquitectura. Aunque REST no es estándar, esta en base de estándares de HTTP(Hipertexto de Transferencia de Protocolo) [10].

2.3.1. Hipertexto de Transferencia de Protocolo - Hypertext Transfer Protocol - HTTP

Se ha definido como un conjunto de métodos, de petición para indicar la acción que se desea realizar, para un recurso determinado. En cada uno de ellos implementan una semántica diferente, pero algunas características similares, los cuales son: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH.

4. **Algunas características de HTTP**, se ha permitido que durante el desarrollo, se implemente mas funcionalidades sobre la Web: cache, flexibilidad del requisito de origen, autenticar, proxies, túneles y sesiones. Para realizar este proyecto se ha utilizado algunas funciones:

- a) **Cache**: es la función, para almacenar los documentos en el cache. Para realizar este proyecto, se utiliza en almacenar la base de datos a lado del cliente con PouchDB en el cache.
- b) **Requisitos de origen**: para compartir la información de datos, se puede flexibilizar la división entre cliente y servidor. Para realizar este proyecto se ha utilizado los permisos para que el servidor de la pagina web SAGAA nos proporcione información.
- c) **Autenticar**: establece una sesión específica, con el cookie el cual es el que almacena información, que guardan en tu ordenador. Para el presente proyecto, la sesión se ha realizado mediante el uso de HTTP cookies, para establecer la comunicación entre la página del SAGAA y la aplicación móvil.

2.4. Unidades de datos de servicio

Es un termino genérico, que se refiere a la variedad de formatos de información, que pasa de la capa superior, a la capa de inferior, al realizar una petición. Estas unidades de datos

pueden ser: Json, Xml y Yaml. En este proyecto se ha utilizado Json, el cual se explicara a continuación.

2.4.1. Notación de objeto de Javascript

La notación de objeto de javascript se conoció como Json (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero y abierto de intercambio de datos, utiliza texto plano para codificar y completamente independiente del lenguaje. Se presenta en objeto en la figura 2.4 y arreglo en la figura 2.4.

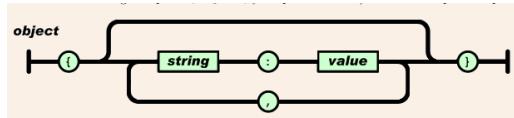


Figura 2.4: Representación de json por objeto, adoptado para la realización del proyecto. Estructura de json por objeto, de Figura: Elaboración propia

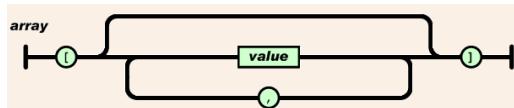


Figura 2.5: Representación del json por arreglo, adoptado para la realización del proyecto. Estructura de json por arreglo de Fuente: Elaboración propia

Las unidades de datos de servicio, tenemos las reclamaciones de seguridad para los datos de servicio, en el intercambio de información.

2.4.2. Json web token

El json web token es un método abierto y estándar para representar las reclamaciones de forma segura, entre dos partes. Se divide 3 partes: la cabecera, carga útil y una firma.

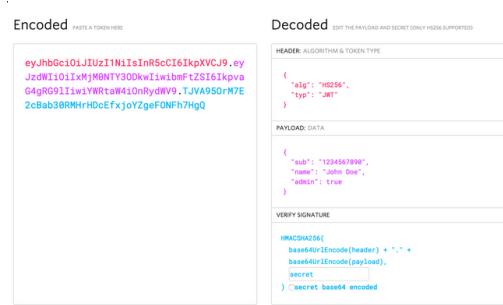


Figura 2.6: Representación del json web token, adoptada para la realización del proyecto de Fuente: Elaboración propia

Para el presente proyecto el tipo de servicio es identificado como servicio heredado, el que solicite los servicios es la aplicación móvil.

2.5. Aplicaciones Móviles

Según Corporación de IBM [11] define a las aplicaciones móviles en tres partes, las cuales son:

2.5.1. 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. Aplicación web

Las aplicaciones web usan únicamente tecnologías basadas en la Web con el potencial de HTML5 son: componentes interfaz de usuario avanzados, acceso a múltiples tipos de medios, servicios de geoposicionamiento y disponibilidad sin Internet. Una de las grandes ventajas es el soporte para múltiple plataformas y se ejecutan dentro del navegador.

2.5.3. Aplicación Híbrida

El enfoque híbrido combina desarrollo nativo con tecnología web, se realiza para múltiples plataformas y mantienen acceso directo a APIs nativas cuando lo necesiten y se puede guardar información. La ventaja que ofrece es que puede aprovechar las características que ofrecen los dispositivos modernos.

Para el presente proyecto, se ha utilizado la aplicación híbrida, porque me permite acceder a la parte nativa del móvil y me permite almacenar información en la aplicación.

2.6. Dispositivos Móviles

Actualmente los dispositivos móviles, son pequeño para ser transportado y utilizado durante su transporte, se caracteriza por el gran mejoramiento que va adquiriendo en su tamaño reducido, la telecomunicación, la gran mayoría de software, la gran interacción con las personas. Convertiéndolo en una necesidad primaria para la sociedad [12]. Desde su creación los dispositivos móviles han evolucionado en gran magnitud, es por eso que muchas empresas han ofrecido diferentes sistemas operativos: android, iOS, Windows, etc. Para el presente proyecto se ha utilizado en sistema operativo de android.

2.6.1. Android

Es un sistema operativo, que se basa en el linux, tiene una plataforma de software basada en Linux para dispositivos móviles, es un sistema operativo libre y tiene acceso a sus recursos como la pantalla, camara y otros [2]. El cual tiene las siguientes capas que muestra en la siguiente figura 2.7.

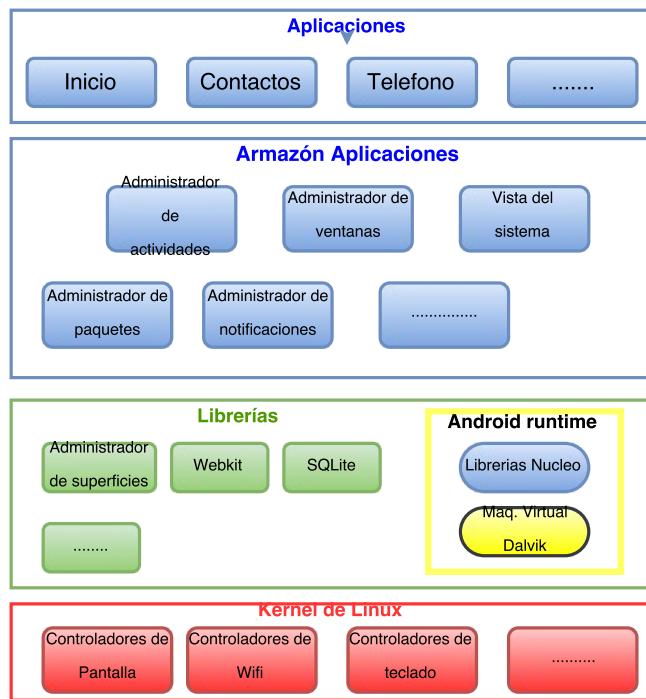


Figura 2.7: Capas del sistema operativo de android, adoptado para la realización del proyecto. Herramienta de android de [2]

El avance de la tecnología, el sistema operativo android, se ha utilizado para diferentes tipos de dispositivos como ser para tablet PC, tableta y otros. Es por el cual, el presente proyecto se realiza un diseño adaptativo.

2.7. Diseño adaptativo

Las satisfacción de visitar una página web, a través de diferentes dispositivos, ha generado la filosofía de Responsive Web Design, esto fue establecida por, Steven Champeon, (2003), que propone realizar un único diseño web para todos los dispositivos y accesible para el usuario. [13]

2.8. Plataformas de desarrollo

Para el presente proyecto, se ha utilizado diferentes herramientas para el desarrollo.

2.8.1. Entorno de desarrollo VIM

Según la página oficial de VIM, es un editor de texto altamente configurable construido para creaciones y cambios de cualquier tipo de texto muy eficiente. Tiene las siguientes ventajas:

- Persistente multinivel
- Extensivos plugin para el sistema
- Soporta muchos lenguajes de programación y formatos de archivos
- Poderoso buscador y remplazar
- Se integra muchas herramientas.

2.8.2. Navegador web - Chrome

Según la página oficial de Google Chrome, es un navegador web rápido, seguro y gratuito, diseñado para la Web actual. También es un navegador para dispositivos móviles. La ventaja que nos brinda es el verificar los errores para celulares, minimizan el gastos de datos, almacena la sesión y permite crear base datos en el navegador.

2.9. Herramientas de Desarrollo

Para el presente proyecto, se ha utilizado diferentes framework que se basan en el lenguaje de javaScript, el cual se han utilizado para la implementación del servicio web y la aplicación móvil.

2.9.1. Framework ionic

Según la página oficial de ionic, es un SDK for HTML5 esto nos ayuda a construir aplicaciones móviles híbridas usando las tecnologías como HTML, CSS y Javascript. Este se caracteriza por ser multiplataforma. Utiliza el Framework AngularJS.

2.9.2. Framework angularJS

Según la página oficial de angularJS, 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 modificada o remplazado, para adaptarse a su flujo de trabajo.

2.9.3. Framework node.js

Según la página oficial de nodeJS, es un entorno de ejecución para javaScript construido con el motor JavaScript V8 de Chrome. Nodejs usa un modelo de operaciones 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.

Capítulo 3

Análisis de la planilla de notas

En este capítulo se explica la funcionalidad y herramientas para descargar y subir la planilla de notas; que son parte de la página del SAGAA. Para modificar la planilla de notas y comprender la funcionalidad de la aplicación del transcriptor.exe se ha realizado entrevista con el personal de la UPSI para comprender las herramientas que utilizan.

3.1. Entrevista

La siguiente entrevista se realiza al *Ing. Cristian, responsable del proyecto Transcriptor.exe en la UPSI de la UMSS* a continuación se muestra los resultados de la entrevista.

- **1.** Explicación de la estructura de la planilla de notas, el archivo sis, donde define sus restricción en el archivo.
 - a) El código de la UPSI.
 - b) El código de template, define la estructura del template del Transcriptor.exe.
 - c) En el template el ARB son valores permitidos.
 - d) En los datos del grupo están definidos el template y las restricciones de las calificaciones.
- **2.** Mencionamos los aspectos, mas importantes y/o limite que se debe considerar a la hora de modificar la planilla de notas en el Transcriptor.exe.
 - a) Mostrar la retroalimentación, en grillas.
 - b) Los estados para volver atras.
 - c) Cuando modifica la lista de estudiantes, no debe permitir cerrar sin guardar.
 - d) Habilitar la 2da opción y tomar en cuenta el mayor entre la 1ra y 2da opción.
 - e) En el enter, ir al siguiente dato para introducir.
 - f) Analizar la usabilidad de la aplicacion móvil.

La entrevista, ha mejorado la comprensión de la estructura y la funcionalidad de llenar la planilla de notas.

3.2. La funcionalidad o operaciones para llenar la planilla de notas

La funcionalidad del proceso para llenar la planilla de notas, se realizan los siguientes pasos que se explica en el capítulo 5 en la sección de 5.1.1, una vez concluido con la explicación cada proceso, en conclusión se ha realizado diagrama general del proceso de llenar la planilla de notas en la figura 3.1

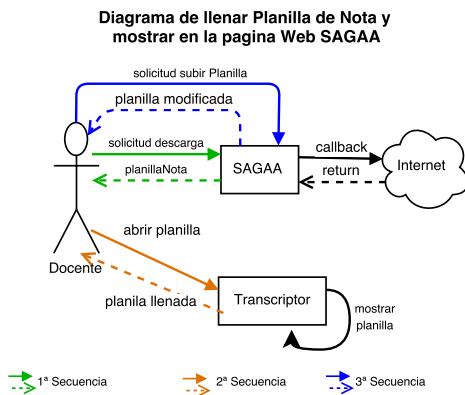


Figura 3.1: Diagrama de la funcionalidad de llenar y mostrar las notas, en la página del SAGAA Fuente: Elaboración propia

A continuación se explica la funcionalidad del llenado de planilla de notas de la página del SAGAA, y las herramientas que utilizan.

3.3. El llenado de planilla de notas de la página del SAGAA

La página del SAGAA, es un Sistema de Apoyo a la Gestión Académica y Administrativa para la FCYT de la UMSS, el cual tiene un alrededor de 10 años, proporcionando un servicio a los docentes y estudiantes. El encargado de realizar mantenimiento es el Lic. Cristian Lazarte Rioja con el cual tiene el cargo de web master, en el departamento de la UTI de la FCYT. Para la elaboración de este proyecto, se ha enfocado en los pasos para descargar y subir la planilla de notas, se utilizan dos herramientas que se explican a continuación:

3.3.1. Pasos para descargar la planilla de notas

Los pasos para descargar la planilla de notas y las restricciones, son las siguientes:

- **Paso 1:** Primeramente se debe escribir la siguiente dirección de página: <http://pruebas.fcyt.umss.edu.bo/sagaa> en el navegador y después se muestra la figura 3.2.

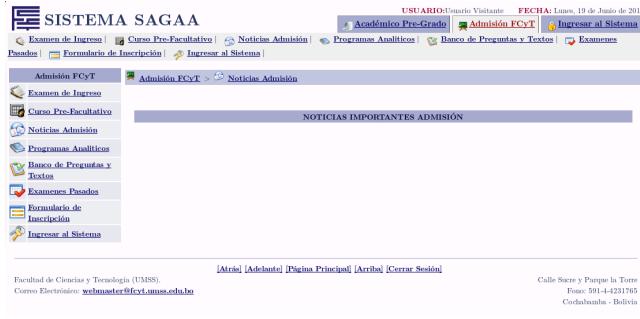


Figura 3.2: Inicio de página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 2:** Hacer click en el link *Ingresar al sistema*, debe ingresar su cuenta de usuario y contraseña, se muestra en la figura 3.3.



Figura 3.3: Iniciar sesión en la página del SAGAA Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 3** Elegir el menú de usuario *Academico Pre-Grado*, se muestra en la figura 3.4.

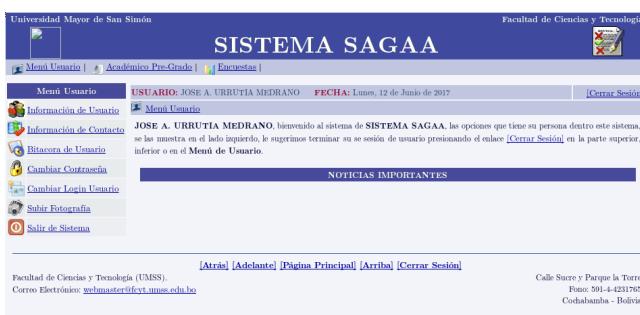


Figura 3.4: Elegir menú Acádemico Pre-Grado Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 4:** Elegir la opción *Descargar Planilla de Notas*, se muestra en la figura 3.5.



Figura 3.5: Elegir opción descargar planilla de notas Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 5:** Seleccionar la gestión de la lista de opciones, después hacer click en *Seleccionar Gestión*, se muestra en la figura 3.6.



Figura 3.6: Opción de gestión, Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 6:** Lista de carreras y detalle, elegir una carrera para descargar la planilla de notas, se muestra en la figura 3.7.

NRO	PLAN	ARCHIVO	NÚMERO DE DESCARGAS	ULTIMA DESCARGA	DESCARGAR ZIP
1	ELECTRICA(29701)	URRUTIA_MEDRANO_JOSE_A_19880028_29701_3-2016.xls	181	12-20-06-2017	
2	ELECTROMECANICA(60001)	URRUTIA_MEDRANO_JOSE_A_19880028_60001_2-2016.xls	2	08-11-2016-2017	
3	ELECTRONICA(429701)	URRUTIA_MEDRANO_JOSE_A_19880028_429701_3-2016.xls	0	-	

[\[Desmarcar todos\]](#) [\[Marcar todos\]](#) [\[Descargar\]](#)

Figura 3.7: Elegir la descarga de una planilla de notas, Fuente:
Proporcionado por la UPSI de la FCYT

Finalmente tenemos el documento de planilla de notas, en donde se debe llenar las notas, después continuamos con la funcionalidad de subir la planilla de notas.

3.3.2. Pasos para subir planilla de notas

El subir la planilla de notas, se realiza los pasos 1-3, de la funcionalidad descargar 3.3.1 y se elige el menú de *Administración Pre-Grado*:

- **Paso 1:** Elegir la opción, *Habilitar Estudiantes*, *Subir Planilla de Notas Parciales (Archivo S/S)*, se muestra en la figura 3.8.

Figura 3.8: Opción de habilitar estudiantes Fuente: Proporcionado por la
UPSI de la FCYT

- **Paso 2:** Subir archivo y seleccionar la gestión de la lista de opciones, después de hacer click en *Subir Archivo*, se muestra en la figura 3.9.



Figura 3.9: Opción de subir archivo seleccionado Fuente: Proporcionado por la UPSI de la FCYT

- **Paso 3:** Lista de grupo, seleccionar el grupo para subir la notas y por ultimo, click en *Finalizar Operación*, se muestra en la figura3.10

NRO	CODIGO MATERIA	NOMBRE MATERIA	GRUPO	SUBIR NOTAS
1	201014	ELECTRONICA DIGITAL I	1	<input checked="" type="checkbox"/>
2	201014	ELECTRONICA DIGITAL I	ME	<input type="checkbox"/>

Figura 3.10: Paso 3: Subir planilla de notas por grupo Fuente: Proporcionado por la UPSI de la FCYT

Al finalizar, se muestra el mensaje *Finalizo la habilitación de estudiantes y la planilla de notas.*

3.3.3. Las restricciones de la página SAGAA

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

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

3.4. Modificar la planilla de notas con la aplicación del transcriptor.exe

El trascritor es un programa ejecutable, que los docentes de la UMSS, tiene la funcionalidad de modificar las notas en el documento de planilla de notas. El creador el Lic. Cristian , que actualmente trabaja en la UPSI realizando mantenimiento a dicha aplicación, el cual tiene las siguientes funcionalidades:

3.4.1. La funcionalidad de la aplicación del transcriptor

La funcionalidad del transcriptor de llenar la planilla de notas, se debe previamente descargar la planilla de nota de la página del SAGAA. Una vez concluido con lo anterior, se pasa a describir los pasos para realizar el llenado de la planilla de notas.

- **Paso 1:** Se debe realizar click en el círculo azul, el botón *Abrir* y luego buscar la ubicación de la planilla de notas, se muestra en la figura 3.11.

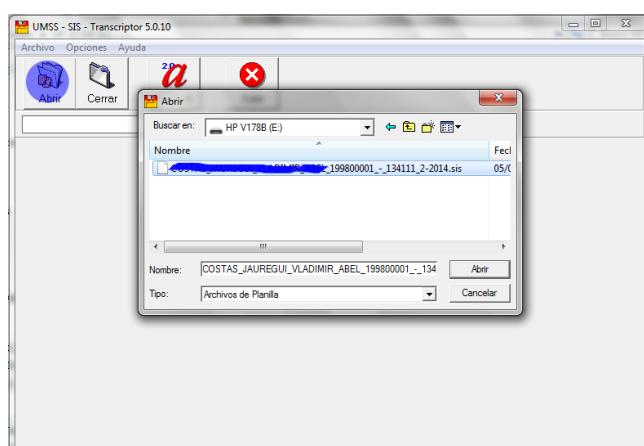


Figura 3.11: Buscar planilla de notas del Transcriptor Fuente: Proporcionado por el MEMI

- **Paso 2:** Se debe elegir un grupo y hacer click en el círculo azul, en el botón *Editar*, se muestra en la figura 3.12.

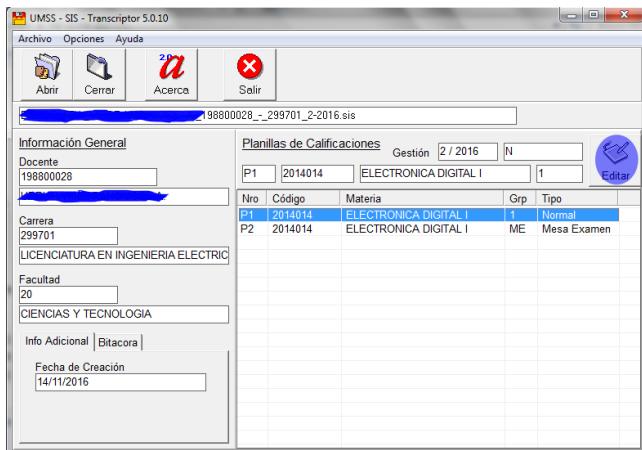


Figura 3.12: Buscar planilla de notas Fuente: Proporcionado por el MEMI

- **Paso 3:** Se muestra el detalle de la materia, la lista de estudiantes, en el círculos rosado se debe llenar las notas, en forma ordenada, se muestra en la figura 3.13.

Figura 3.13: Lista de estudiantes, Fuente: Proporcionado por el MEMI

- **Paso 4** Debe guardar y hacer click en el círculo azul botón *Guardar*, para cerrar la ventana y guardar los datos en el archivo de la planilla de nota,, se muestra en la figura 3.13.

3.4.2. Detalle del Transcriptor

El transcriptor, interpreta los datos de la planilla de notas que se encuentra con la extensión sis, se muestra en los siguiente figura 3.14.

- **Caso 1:** se muestra la información general y lista de materias en la figura:

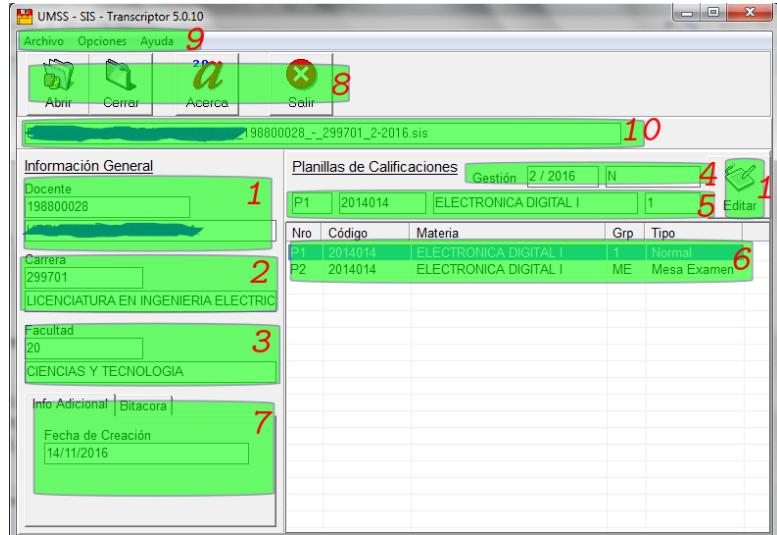


Figura 3.14: Detalle de la lista de grupo Fuente: Proporcionado por el MEMI

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

- a) Los datos del docente el código y el nombre y apellido.
- b) Los datos de la carrera el código y nombre.
- c) Los datos de la facultad el código y nombre.
- d) Los datos de la gestión.
- e) Los datos del grupo el código, nombre, código de reconocimiento y numero de grupo.
- f) Lista de grupos muestra el número, código, nombre, identificador y tipo.
- g) Muestra la información adicional fecha de creación y bitácora del archivo.
- h) El menú los eventos son: abrir, cerrar, acerca y salir.
- i) El menú los evento son: archivo, opciones y ayuda.

- **Caso 2:** se muestra el contenido de la lista de estudiantes, se muestra en la figura 3.15.

Figura 3.15: Detalle de la lista de estudiantes Fuente: Proporcionado por el MEMI

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

- Lista de estudiantes con los siguientes datos: numero, código, nombre, 1er parcial, 2do parcial, promedio, examen final, 2da instancia, nota final, nota contado.
- Datos del grupo: código de reconocimiento, tipo, numero de grupo.
- Datos de la materia: el código, nombre.
- Dato gestión.
- Datos extra de los estudiantes la cantidad, porcentaje de reprobado, aprobado y abandonado.
- Lista de menú son guardar, imprimir, imprimir 2da instancia, pesos, copia todo, importar csv, exportar csv y salir.

3.4.3. Restricciones del transcriptor

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

- Solo funciona en el sistema operativo windows.
- Te permite abrir solo archivos con la extensión sis.
- Reconoce la estructura de almacenar información en caso contrario no permite abrir el archivo.
- Controla cada dato a ser insertado.
- Protege la información del archivo y solo permite modificar la nota del estudiante.
- Solo te permite modificar la nota por grupo y controlar la información almacenada.
- No te permite cerrar la lista de grupos, sin previamente haber guardado.

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

3.5. Planilla de notas

Es un archivo que tiene una estructura predefinida, por lo cual se utiliza para compartir información y el formato. Es por el cual la UMSS, lo utiliza para el proceso de llenado de las planilla de notas. La UPSI, realiza un mantenimiento al sistema generador de la planilla de notas, con la extensión sis.

3.5.1. Información de la planilla de notas

La planilla de notas almacena información de los docentes, estudiantes y el formato, se explica en la estructura:

- **La información de la planilla de notas:** en cada línea de comando se explica la información.

```
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
```

- **La estructura del grupo normal y mesa:** La estructura del grupo normal y mesa es para el formato de la aplicación del transcriptor.exe, como se muestra en la figura 3.13, el formato del grupo normal y 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
```

Detalle de los grupos: a) 1 es el orden de la fila, b) NUMERO es identificador de la etiqueta ordenado por fila, c) RD o WR permiso de lectura o escritura, d) 0 es el inicio del rango de datos, e) 100 o 51 es el final del rango de datos, f) A o R o B es el valor permitido, g) NRO

nombre de la etiqueta ordenado por fila, h) 38 o 72 o 230 o 52 o 50 ancho de la columna, i) N para habilitar si tiene nota final, j) N para habilitar si tiene segunda instancia k) NONE palabra reservado.

La diferencia entre el grupo normal y mesa es la cantidad de columnas que cada uno tiene, el grupo normal que tiene 10 columnas y el grupo mesa tiene 7 columnas.

- **La estructura de información del grupo normal y mesa:** tiene la lista de grupos, el tipo de grupo que se explica a continuación . 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
```

a) P1 o P2 es el código grupo para formato, b) 2014014 es el código de la materia, c) 1 o ME son tipo de grupo, d) ELECTRONICA DIGITAL I es el nombre de la materia, e) 263 tipo de template, f) 100 peso global de T1 y T2, g) 0 peso del examen final, h) 50 es T1, i) 50 es T2, j) 0 es T3, k) 0 es P1, l) 0 es P2, m) 0 es P3. Los P son parciales y T son trabajos. La estructura para los grupos es estándar para todas las carreras de la UMSS, esto significa que cada carrera tiene su modalidad de calificación.

- **La estructura de información de los estudiantes:** es una lista de estudiantes, los cuales son los datos de la nota de cada estudiante divididos por grupo.

- a) Lista de estudiantes de grupo normal:

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

En el template normal define 10 columnas, es lo que define la lista de estudiantes 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.

- b) Lista de estudiantes del grupo mesa:

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

En el template normal define 7 columnas, es lo que define la lista de estudiantes los cuales 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. Estructura de la planilla de notas

Tiene una estructura de etiqueta de inicio y final, el cual divide la información de manera organizada, para compartirla.

```
<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)
</column>(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. Restricciones de la planilla de notas

La planilla de notas es un archivo con extensión sis, el cual 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 el transcriptor.

Después de concluir con el análisis de las funciones y procesos, el cual nos ayuda para el desarrollo del servicio web y la aplicación móvil.

Capítulo 4

Proceso de desarrollo de servicio

El proceso de desarrollo de servicio web es representar la abstracción de reutilización de una aplicación a través del análisis, los detalles del diseño, la clasificación del tipo de servicio ofrecido y las etapas de la ingeniería de servicio.

4.1. Análisis del servicio web

El documento según [1], ayuda a conocer la ubicación y de los detalles de la interfaz de donde se obtiene la información, para el servicio web, los cuales son:

- **Qué:** Se realiza un documento llamado interfaz es donde, se define las operaciones que soporta el servicio y define el formato de los mensajes que se envían y reciben por parte del servicio.
- **Cómo:** Se realiza un documento llamado enlace, mapea la interfaz abstracta a un conjunto concreto de protocolos, especifica los detalles técnicos de cómo comunicarse con un servicio.
- **Dondé:** Se realiza un documento, en donde se describe la ubicación de la implementación del servicio web en su punto final.

Estos documentos de análisis del servicio web, ayuda a conocer los campos, los mensajes de entrada y salida y otros. El cual se tiene como resultado los detalles de diseño.

4.2. Detalles de diseño

Los detalles de diseño nos ayuda a organizar las actividades del servicio web. Las herramientas para el desarrollo de diseño del servicio web son:

- a) **Flujo de trabajo**, es un conjunto de actividades orientadas para realizar parte del trabajo, el cual establece los pasos necesarios para alcanzar una meta particular.

- b) **Acción de compensación** La acción de compensación se utilizan para deshacer acciones, que se completan, el cual deben cambiar el resultado de posteriores actividades del flujo de trabajo.

El diseño de flujo de trabajo, es analizar los procesos planeados para comprender, las diferentes actividades, comó intercambian información, definen las etapas de proceso. Se muestra el diseño de una representación gráfica como el diagrama de actividad UML. Tiene algunos conceptos, centrales de BPMN que se usan para crear modelos de flujo de trabajo.

- a) Las actividades se representan mediante un rectángulo con esquinas redondeadas. Una actividad puede ejecutarse por una persona o por un servicio automatizado.
- b) Los eventos se representan por medio del círculo. Un evento es algo que sucede durante un proceso: evento inicial(círculo claro), evento final(círculo oscuro), evento intermedio (círculo doble no se muestra) se ejecuta periódicamente.
- c) Las compuertas se representa mediante un diamante, es una etapa en el proceso de donde se hace una elección.
- d) La secuencia de actividades, se representa mediante una flecha sólida, se usa para mostrar la secuencia de actividades. Los mensajes que fluyen entre actividades se representan en una flecha punteada.

La descripción y características del flujo de trabajo, apoya en el desarrollo de la ingeniería de servicio.

4.3. Ingeniería de servicio

La ingeniería de servicio es el proceso, de desarrollo de servicios para la reutilización en aplicaciones orientadas a servicios, que representan la fiabilidad de los sistemas. El punto de partida, para el proceso de servicio, es un servicio existente, o un componente que se convertirá en servicio.

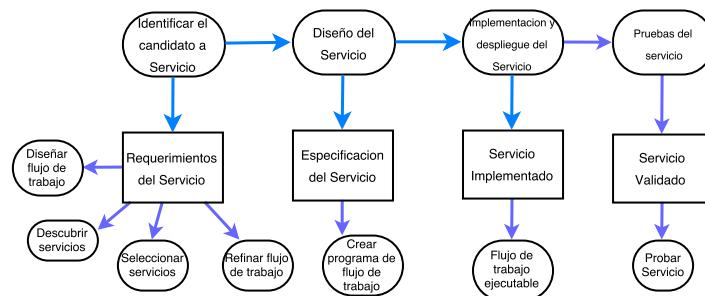


Figura 4.1: El proceso de ingeniería de servicio, adoptado para la realización del proyecto. El proceso de ingeniería de servicio de [1]

El desarrollo de software es combinar y configurar servicios; para crear nuevos servicios compuestos. Los nuevos servicios, son composiciones de servicios que pueden usarse para dar

un proceso integrado que ofrezca funcionalidad mas extensa, para cumplir una meta.

4.3.1. Identificación de candidatos a servicio

Para identificar, el servicio implica comprender y analizar los procesos empresariales de la organización para decidir cuál servicio se podrían reutilizar para implementar el proceso.

Los servicios pueden ser orientados a entidades que son como objetos y orientados a tareas, que son aquellas asociadas con alguna actividad. Para el presente proyecto, es un servicio orientado a tareas, porque se asocia con actividad. Para identificar el servicio, se plantean las siguientes preguntas.

- a) El servicio de entidades esta asociado, con una solo entidad lógica que se usa en diferentes procesos empresariales?
- b) Que operaciones deben soportarse se realizan usualmente sobre dicha 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 usarlo, clientes fuera de la organización?
- g) Diferentes usuarios, tienen distintos requerimientos?

El resultado del proceso de selección, son un conjunto de servicios identificados y la lista de requerimientos funcionales del servicio, es definir que hace el servicio. Los requerimientos no funcionales deben definir los requerimientos de seguridad, rendimiento y disponibilidad del servicio. Los requerimientos funcionales del servicio, apoyan en crear el flujo de trabajo para identificar el servicio web.

- a) **Formular el bosquejo de flujo de trabajo**, para el flujo de trabajo, se utilizan los requerimientos para el servicio compuesto.
- b) **Descubrimiento de servicios**, se buscan los servicios existen, quien los proporcionan y los detalles.
- c) **Identificar el servicio**, una vez la identificado, se tiene los criterios de selección, las funcionalidades de los servicios ofrecidos.
- d) **Refinar el flujo de trabajo**, en la base de la información acerca de servicio, que se ha seleccionado, se refina el flujo de trabajo. Se vuelve a repetir la etapa de descubrimiento y selección.

4.4. Diseño de servicio web

Una vez concluido la identificación del servicio, se realiza las operaciones asociadas con el servicio y sus párametros, el diseño de operaciones y los mensajes de servicio. El diseño se representa en 3 tipos, pero para el presente proyecto solo se utiliza los siguientes:

- a) Diseño de interfaz lógica, se identifica las operaciones asociadas con el servicio, sus entradas y salidas, así como las excepciones asociadas con dichas operaciones. Después se realiza los requerimientos del servicio, define los nombres y parámetros de operación. Se debe definir las excepciones que surgen cuando se ubica una operación de servicio.
- b) Diseño de mensajes, es la estructura de los mensajes que envía y recibe el servicio. Se define los mensajes de entrada y salida, se debe manejar los mensajes como objetos, tiene la estructura en operación, entradas, salidas y excepciones.

Una vez concluido, con el diseño para análisis del servicio, se realiza el diseño para desarrollar el servicio con flujo de trabajo.

- a) **Crear un programa de flujo de trabajo**, es el diseño de flujo de trabajo que se transforma en un programa ejecutable.
- b) **Crear un diseño de datos de servicio**, es el diseño para mostrar el formato de información, para utilizar el servicio web.

4.5. Implementación del servicio web

El modelo puede realizar algunas iteraciones, hasta que se cree un diseño que permite la reutilización de los servicios disponibles. Utilizamos el diseño de desarrollo del servicio de flujo de trabajo, que se debe convertir en un programa ejecutable, esto implica la actividad de:

- a) Implementar los servicios, los servicios son independientes del lenguaje de implementación, dichos servicios se pueden escribir en cualquier lenguaje de programación.

4.6. Pruebas del servicio

Las pruebas, son importante en todos los procesos de desarrollo del sistema se muestran que un sistema, cumple con los requerimientos funcionales, cuando se utiliza un servicio externo, ésto significa que no se tiene acceso a su código fuente, es por lo cual que no se debe hacer en base a código fuente. Para comprender lo implementación se debe realizar prueba y combinación de los servicios:

- 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.

Capítulo 5

Diseño y Implementación del Servicio Web

Como se ha mencionado en la Cápítulo 3, el principal objetivo del desarrollo del servicio, optimizar la elaboración del servicio web a través del análisis, diseño e implementación para identificar los servicios que brinda la página del SAGAA.

5.1. Análisis del servicio web

Para el presente proyecto, analizar cada proceso de la página del SAGAA, es identificar la ubicación y descripción del detalle de la interfaz. Para el documento de interfaz se utilizan las operaciones que soportan el servicio web y el documento de enlace, se muestra una interfaz abstracta con un conjunto de protocolos.

5.1.1. Documento de interfaz

El documento interfaz que responde el *Qué* tipo de operaciones realiza el servicio en cada proceso. Para el presente proyecto, como caso de estudio se ha utilizado a la página del SAGAA, el cual tiene tres procesos que se muestran a continuación:

- **Proceso 1:** la secuencia de procesos para la descarga de la planilla de notas en el formato .sis, de la página del SAGAA, se representa en la figura 5.1.

DESCARGAR LA PLANILLA DE NOTAS

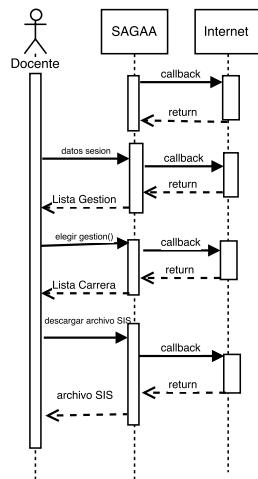


Figura 5.1: Proceso de descargar Planilla Notas. Figura: Elaboración propia

- **Proceso 2:** la secuencia de procesos para modificar la planilla de notas, se utiliza la aplicación del Transcriptor.exe para abrir el documento, se representa en la figura 5.2.

Modificar Planilla de Nota

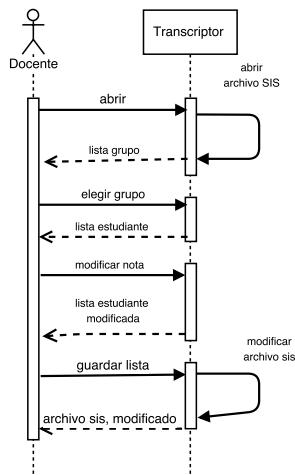


Figura 5.2: Proceso de modificar la Planilla de Notas, Fuente: Elaboración propia

- **Proceso 3:** la secuencia de procesos para finalizar, es el subir la planilla de notas, en la formato sis, a la página del SAGAA, se representa en la figura 5.3.

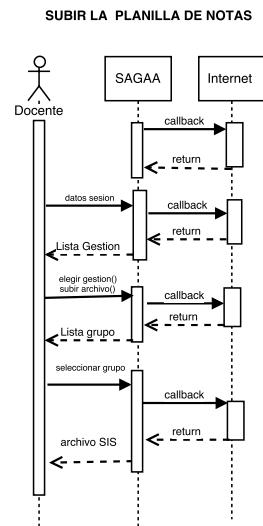


Figura 5.3: Proceso de subir a la página del SAGAA, Fuente: Elaboración propia

5.1.2. Documento enlace

El documento de enlace, responde el *Cómo* se representa graficamente la interfaz abstracta a un conjunto concreto de protocolos, especifica los detalles técnicos de cómo se comunica el servicio.

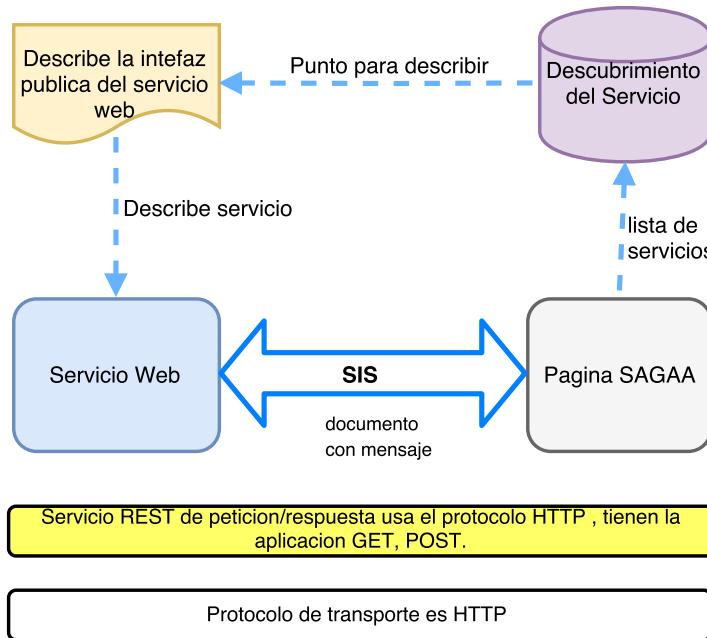


Figura 5.4: Protocolos de servicio, Fuente: Elaboración propia

En la figura 5.4, se muestra los protocolos que utiliza la página web SAGAA.

5.2. Identificar el servicio web

Para el presente proyecto, implica identificar el servicio web, se plantean a través del siguiente cuestionario.

5.2.1. Cuestionario de 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 estudiantes, pero no está habilitado para ellos.

7.- Diferentes usuarios, tienen distintos requerimientos?

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

El resultado del proceso de selección, se realiza la lista de requerimientos para el servicio web.

5.2.2. Requerimientos del servicio web

Los requerimientos son un conjunto de servicios, las cuales son definidas las funciones del servicio.

- a) Descargar la planilla de notas según la gestión y la carrera correspondiente.
- b) Proporcionar una unidad de datos de la planilla de notas.
- c) Adjuntar la planilla de nota a la página del SAGAA.

Una vez concluido con los requerimientos funcionales, se realiza el diseño, de flujo de trabajo. El diseño de flujo de trabajo, es combinar y configurar el servicio, para crear nuevos servicios compuestos.

5.2.3. Formular el bosquejo de flujo de trabajo

Es la etapa inicial, del diseño del servicio, se ha creado la secuencia de pasos del llenado de planilla de notas, se muestra en la figura 5.5.

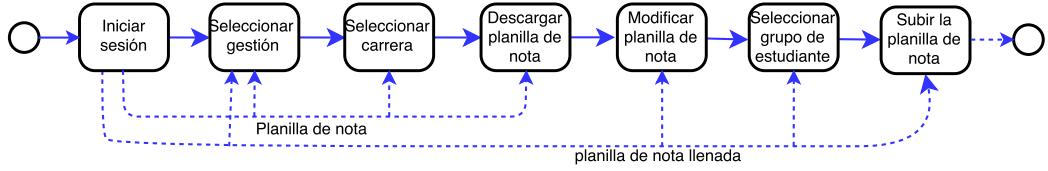


Figura 5.5: Diagrama de flujo de trabajo, Fuente: Elaboración propia

5.2.4. Descubrimiento del servicio

Durante la etapa de descubrir, se ha definido el servicio web, como un servicio de sistemas heredados, por el cual brinda acceso a las funciones y datos de la página del SAGAA. Para el servicio heredado de la página del SAGAA tenemos la siguiente lista de servicio:

Lista de servicio

- Realizar la sesión.
- Listar la gestión.
- Seleccionar la gestión
- Listar las carreras.
- Seleccionar carrera, para descargar la planilla de notas.
- Crear una unidad de datos de la planilla de nota.
- Subir la planilla de nota modificado.

5.2.5. Seleccionar los posibles servicios

En la etapa de seleccionar, se trabaja a partir del conjunto de lista de servicio, las cuales son las funciones y procesos que se desarrollan para obtener el servicio web, se realizan las especificaciones del servicios a continuación.

Especificaciones de los servicios

- Descargar la planilla de notas, se debe realizar la sesión del docente, seleccionar la gestión y seleccionar la carrera para descargar la planilla de nota.
- Modificar la planilla de notas, se debe guardar la planilla de datos local, eliminar el inicio y fin para modificar la planilla de notas a unidad de datos json, y con la ayuda de la aplicación móvil mostrar y modificar la planilla de notas y enviar al servicio web, el cual envie a la pagina web SAGAA.
- Subir la planilla de notas, se debe elegir el grupo de materia que desea, subir la planilla de notas.

5.2.6. Refinar el flujo de trabajo

Las especificaciones de 5.2.5 y listas 5.2.4, contribuyen para refinar el flujo de trabajo y la comunicación con el cliente. Donde se ha añadido los detalles de la descripción abstracta de cada especificación del servicio.

- a) **Diseño de flujo de trabajo de descarga**, el diseño de descargar la planilla de notas, es la secuencia de procesos que se representa a través del diagrama de flujos de trabajo, se muestra en la figura 5.6.

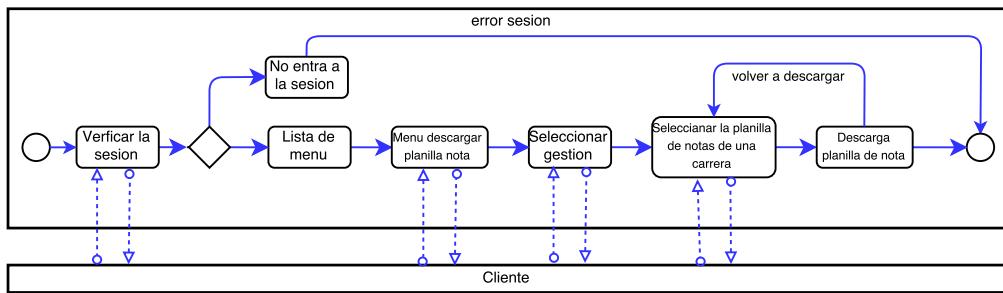


Figura 5.6: Diseño de flujo de trabajo de la descarga de la planilla de notas,
Fuente: Elaboración propia

- b) **Diseño del flujo de trabajo modificar**, el diseño de modificar la planilla de notas, se realiza con la aplicación del Transcriptor.exe, el cual es la secuencia de procesos, se representa a través del diagrama de flujos de trabajo, se muestra en la figura 5.7.

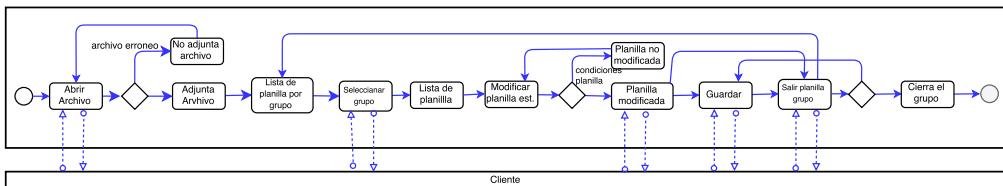


Figura 5.7: Diseño de flujo de trabajo de la descarga de la planilla de notas,
Fuente: Elaboración propia

- c) **Diseño del flujo de trabajo subir**, el diseño de subir la planilla de notas, es la secuencia de procesos, se representa a través del diagrama de flujo de trabajo, se muestra en la figura 5.8.

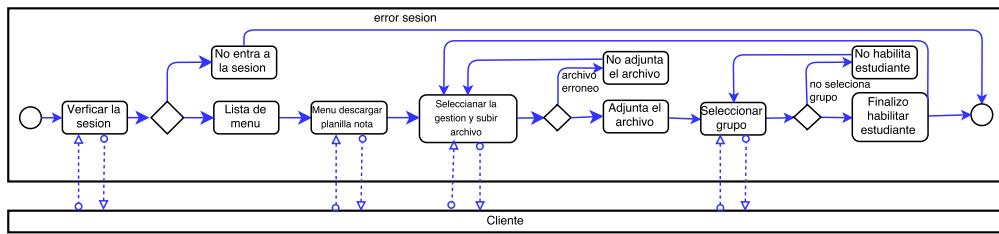


Figura 5.8: Diseño de flujo de trabajo de subir la planilla de notas, Fuente:
Elaboración propia

5.3. Diseño del servicio web

Una vez concluido con la selección del servicio, aporta con las operaciones asociadas con el servicio y sus parámetros de entrada y salida. El diseño de servicio se dividen en tres partes, pero en este caso solo se utilizan dos el diseño de interfaz lógica y de mensajes que aportan con el reconocimiento de protocolos de comunicación.

5.3.1. Diseño de interfaz lógica

El Diseño de interfaz lógica, es la comunicación entre el cliente y servidor de la página del SA-GAA, se detallan las operaciones que realiza, los parámetros de entradas y salidas al realizar el descargar y subir la planilla de notas.

- **Diseño de interfaz lógica de descargar planilla de notas**, en la figura 5.9 es la secuencia de peticiones ordenadas y los parámetros de entrada y salida o mensajes y las direcciones de url para realizar las peticiones de descargar la planilla de notas y obtener los datos que se utilizan para el servicio web.

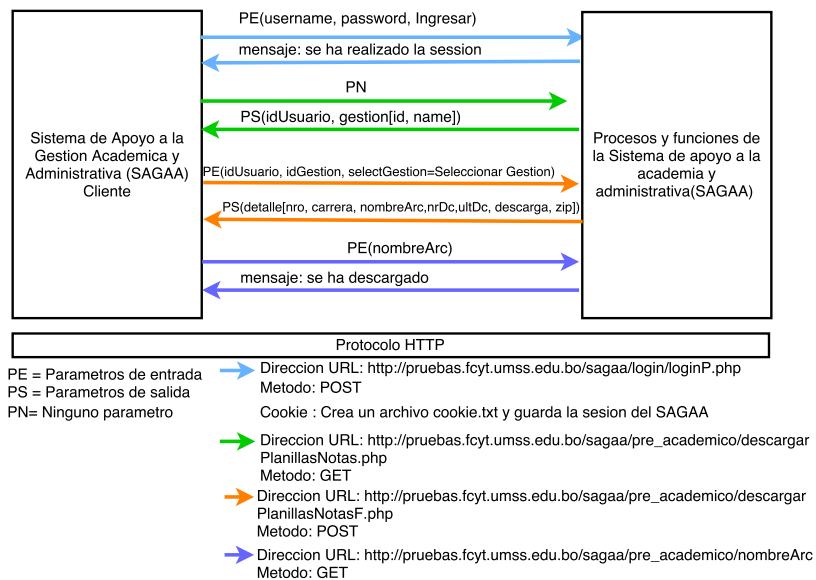


Figura 5.9: Diseño de interfaz lógico para descargar la planilla de nota,
Fuente: Elaboración propia

- **Diseño de interfaz lógica de subir planilla de notas**, en la figura 5.10 es la secuencia de peticiones ordenadas y los parámetros de entrada y salida o mensajes y las direcciones de url para realizar las peticiones de subir la planilla de notas y obtener los datos para el servicio web.

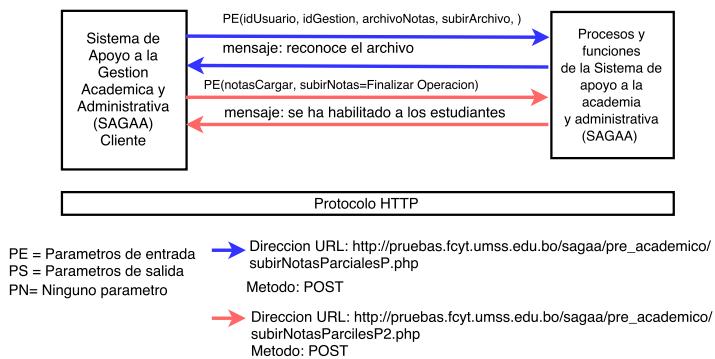


Figura 5.10: Diseño de interfaz lógico para subir la planilla de nota, Fuente:
Elaboración propia

5.3.2. Diseño de mensajes

El diseño de mensajes, son los mensajes que transcurren en la comunicación entre el cliente y servidor cuando se realizan una petición, los mensajes son para mostrar una respuesta correcta o incorrecta y si tienen excepciones que pueden ocurrir durante la comunicación.

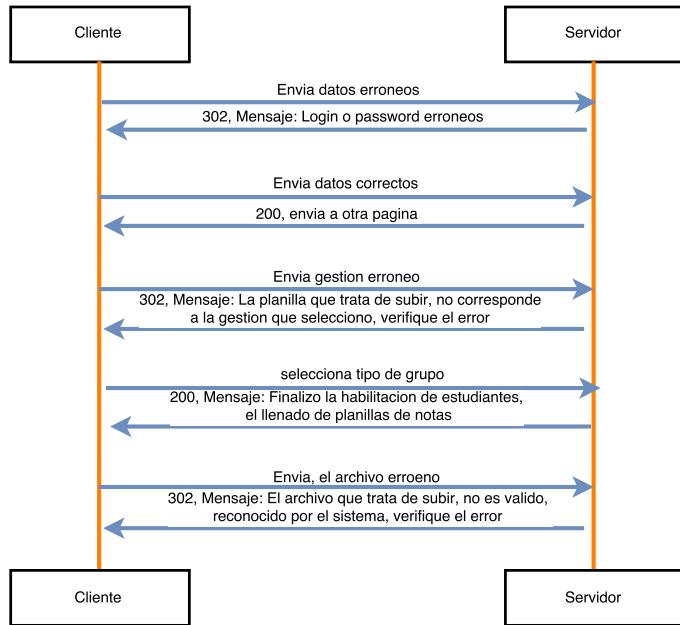


Figura 5.11: Diseño de mensaje de entradas, salidas y excepciones, Fuente:
Elaboración propia

En la figura 5.11, se representa la secuencia de peticiones que realiza el cliente y el servidor y los mensajes que la página del SAGAA responde, que se utilizan para el desarrollo del servicio web.

5.3.3. Crear un programa de flujo de trabajo

El diseño de crear un programa de flujo de trabajo, es la unión de la figura 5.6, 5.7, 5.8 y la comunicación entre la secuencia de procesos, se representa en la figura 5.12, el cual apoya para la crear el servicio web.

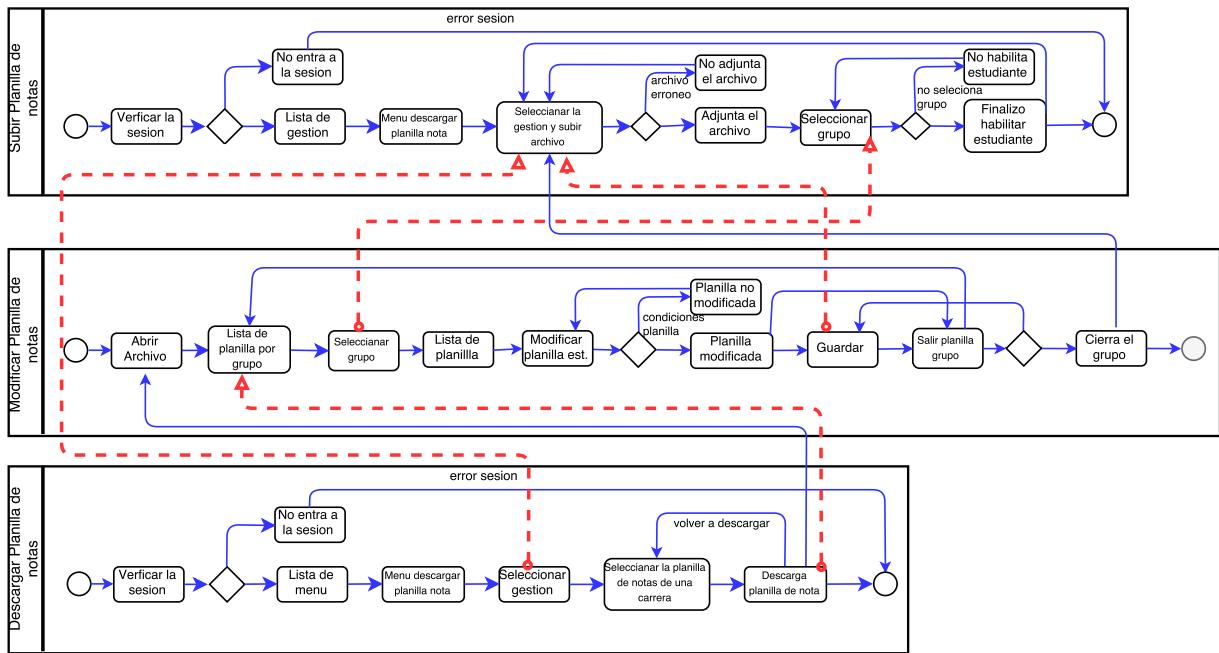


Figura 5.12: Diseño de flujo de trabajo planilla de notas, Fuente:
Elaboración propia

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

5.3.4. Diseño de unidad de datos de servicio

El formato que se utiliza para intercambiar información, es un formato ligero conocido como **JSON** es representado en objetos y arreglos.//

La unidad de datos de la página del SAGAA es definida con etiquetas y tiene la extensión sis, para el presente proyecto se ha utilizado el mismo diseño.//

La unidad de datos sis, ha sido modificado para convertirlo a un formato json, y así la aplicación móvil pueda reconocerlo.

5.4. Implementación del servicio web

Para la implementación del servicio web se realiza a partir del diseño de crear un programa de flujo de trabajo 5.3.3, el cual se utiliza para crear un nuevo diseño del flujo de trabajo, para el desarrollo del programa ejecutable del servicio web, se muestra en la figura 5.13.

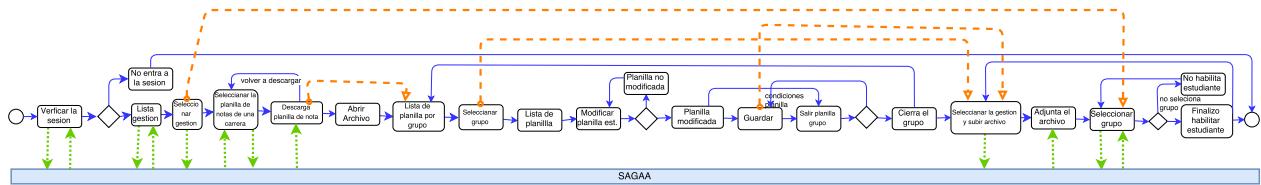


Figura 5.13: Diseño de flujo de trabajo planilla de notas para la implementación, Fuente: Elaboración propia

Es la secuencia de proceso, para el desarrollo del servicio web, las líneas naranjas son parámetros que comparten entre actividades.

5.4.1. Herramientas y configuración del servicio web

Para el presente proyecto, se ha utilizado el lenguaje de programación de Javascript y Express es un framework de NodeJs, el cual 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.

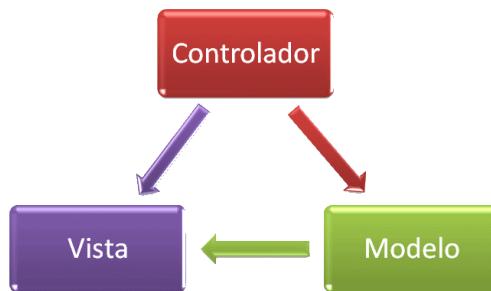


Figura 5.14: Arquitectura de Express, adoptado para la realización del proyecto. Arquitectura de Express de [3]

Para el presente proyecto se ha implementado solo la parte del modelo, para explicar la comunicación del servicio web, se muestra la figura 5.15, la arquitectura de NodeJs.

- **Arquitectura de NodeJs**, nodeJs utiliza la arquitectura orientado cliente/servidor.

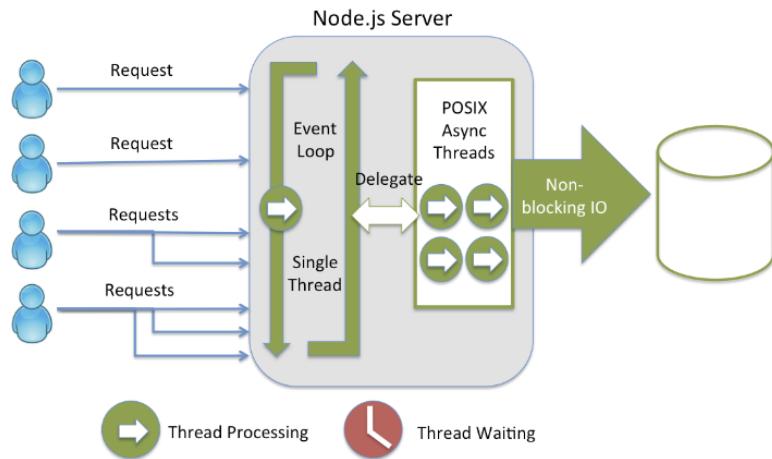


Figura 5.15: Arquitectura de NodeJs, adoptado para la realización del proyecto. Arquitectura de NodeJs de [4]

En la arquitectura de NodeJs, se puede analizar el trabajo del servidor para este proyecto, la parte de peticiones de entrada y salida.

5.4.2. Estructura del servicio web

Como se ha mencionado anteriormente, para el presente proyecto se ha utilizado el modelo, el cual se aumenta a la estructura de Ionic y se ha agregado el 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
```

5.4.3. Configuración del servicio web

Previamente se debe instalar node y npm, luego se puede 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)
```

5.4.4. Implementación del servicio web en Express

Para implementar el servicio web, se ha utilizado la herramienta Curl, Cherrie y una cuenta de docente que ha sido proporcionada por la UPSI de la FCYT y comienza a trabajar según la figura 5.13, se obtiene los siguientes servicio 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.
- Depende del anterior item seleccionar grupo para habilitar estudiante.

Cada servicio web, es muy importante para lograr el objetivo general; se utilizan los datos de las figuras 5.9, 5.10 para realizar cada servicio web. Para mostar la implementación del servicio web, se elige algunas para explicar a continuación:

- a) **Servicio web verificar la sesión**, se debe conocer la dirección de url, el método que utiliza, los parámetros, agentes de autorización y verificar si énvia 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
});
.....
```

- b) **Servicio web seleccionar la gestión y adjuntar planilla de notas**, se debe conocer la dirección de url, el método que utiliza, parámetros encriptados, httpheader donde esta los agentes de autorización, la ubicación de planilla de notas para adjuntar y verificar el mensaje.

```
.....
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
.....
```

- c) **Servicio web filtrar la planilla de notas**, se muestra la planilla de notas y se elimina algunos datos por columna, para convertir a una unidad de datos de servicio.

```
.....
nuevoFile = data.split(/\n/);
inicio = nuevoFile[0];
nuevoFile.splice(nuevoFile[0],1);
fin = nuevoFile[nuevoFile.length-1];
nuevoFile.splice([nuevoFile.length-1],1);
nuevoFile.join('');
.....
```

- d) **Servicio web crear la unidad de datos json**, para crear la unidad de datos se debe convertir la planilla de notas, a la unidad de datos json.

```
var parseString = require( 'xml2js' ).parseString;
.....
app.post('/detalle', function(req, res){
.....
```

```

parseString( fileS( body ), function( error, result ){
    .....
    sisJson = result;
    .....
});
.....
}

```

- e) **Buscar en la planilla de notas según la unidad de datos modificada**, en la planilla de notas, verifica los datos modificados por la unidad de datos.

```

\\ 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);
                    }
                }
            .....
        }
}

```

- f) **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');
        });
    .....
}

```

- g) **La configuración la sesión de token**, en el servicio web el cual crea una clave se realiza en el archivo www/js/server.js.

```

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 que se explica en el capítulo 9.

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 obtiene a partir del requerimiento del servicio web en el capítulo 5 en la sección 5.2.2 y la recolección de datos en el capítulo 3 y cumple con el objetivo específico en la sección 1.4.

- 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.

Para el presente proyecto, después de concluir con los requerimientos, se divide por modulos para el desarrollo del presente proyecto.

6.2. Módulos del proyecto

Para la creación de módulos, se analiza los requerimientos y procesos del proyecto, se dividieron en 4 módulos:

6.2.1. Los módulos para el 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. Dividiendo por sub modulo

- 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 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.

Después de la lista de módulos, se realiza el diseño de interfaz para la aplicación móvil.

6.3. Diseño de interfaz

El diseño de interfaz, especifica los detalles de los módulos de 6.2. El cual debe contener operaciones para acceder a los datos y actualizar. Para el presente proyecto el modificar la planilla de notas, se realiza el diseño de la interfaz para el cliente, se utiliza la herramienta de wireframe.

Para el presente proyecto, se realizaron 2 intentos de diseños, están representadas de forma secuencial para cumplir con los objetivo los específicos 1.4, los cuales son:

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

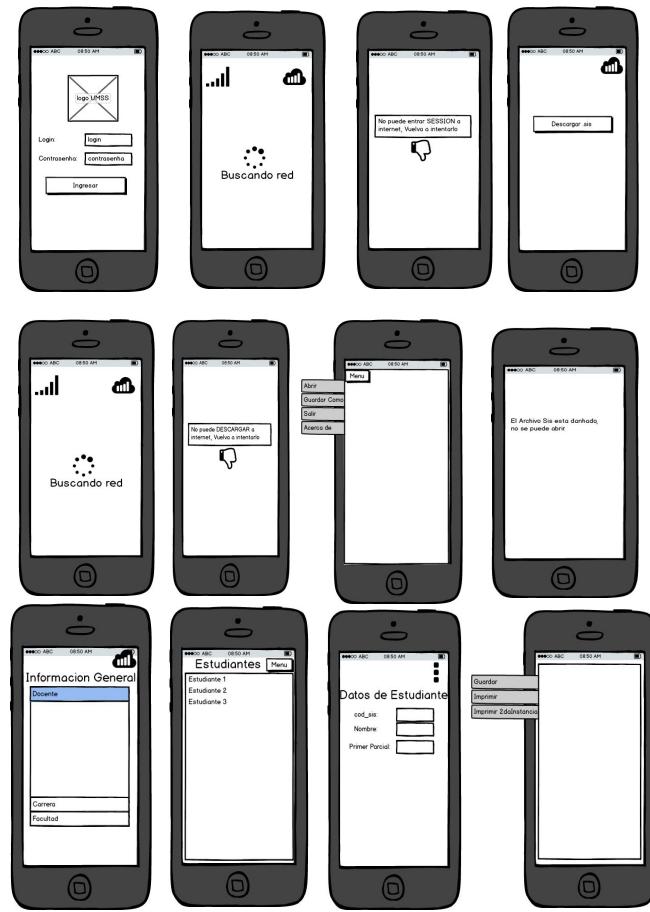


Figura 6.1: Interfaz de Usuario primer intento, Fuente: Elaboración Propia

- 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.

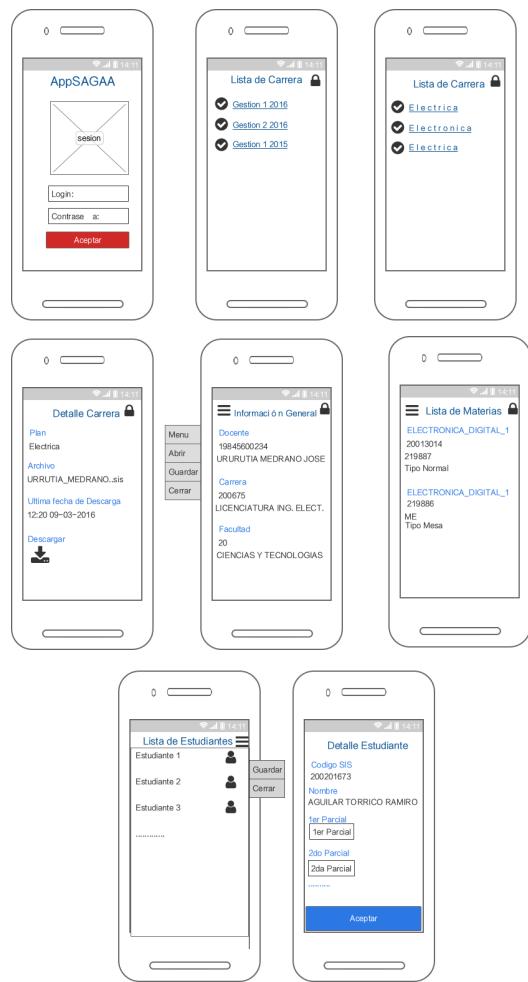


Figura 6.2: Interfaz de Usuario representado en mockups, Fuente:
Elaboración Propia

Una vez concluido con la interfaz 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

Para el presente proyecto, el diseño de comunicación entre el cliente y el servicio web, son las peticiones y las respuestas de la interfaz del usuario al servicio web, que se muestra en la figura 6.3.

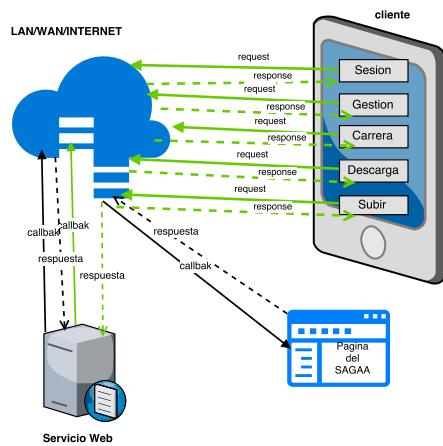


Figura 6.3: Diseño de comunicación entre el cliente y servicio web
Fuente:
Elaboración Propia

Después de concluir con el análisis y el diseño el cual aporta para la implementación de la aplicación móvil.

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.

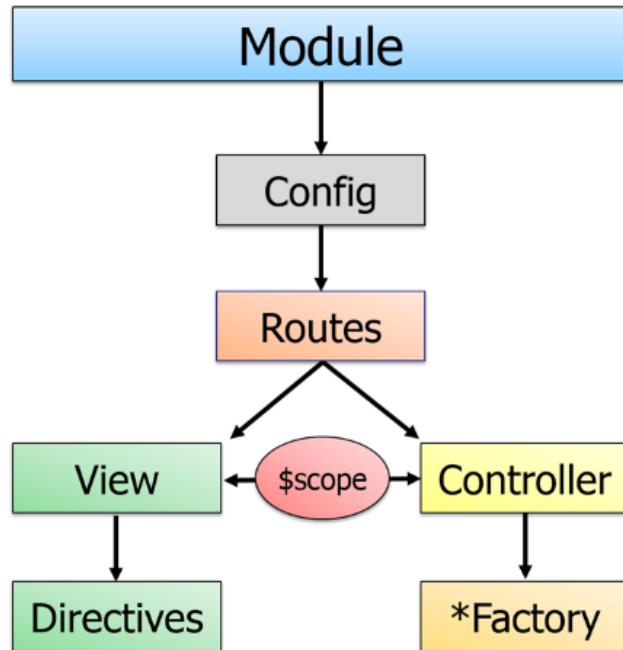


Figura 7.1: Arquitectura de Ionic, adoptado para la realización del proyecto
Fuente: [5]

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.

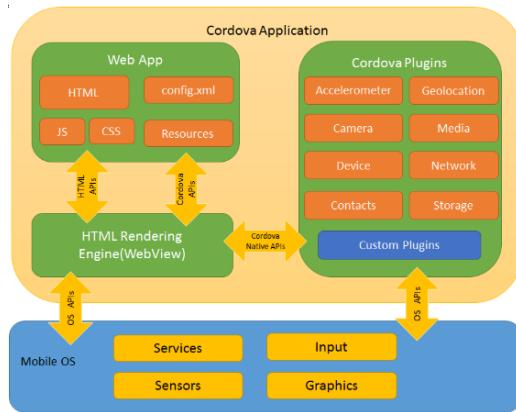


Figura 7.2: Arquitectura de Cordova, adoptado para la realización del proyecto Fuente: [6]

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.

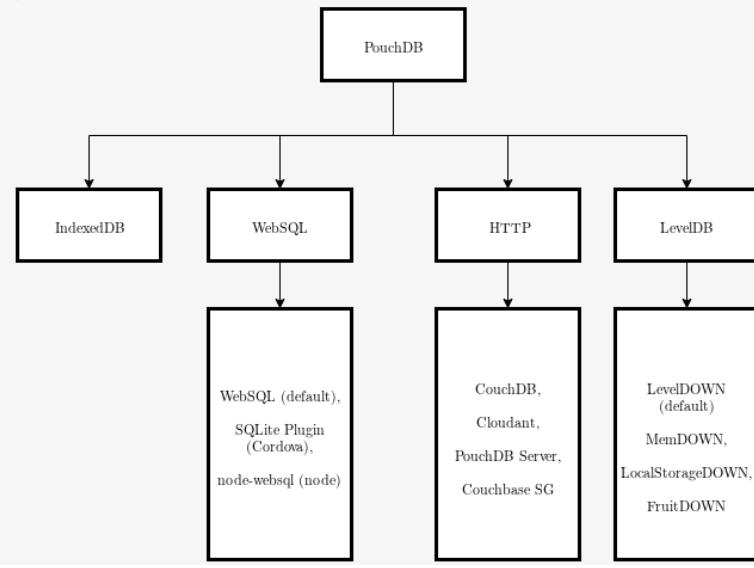


Figura 7.3: Adaptadores de PouchDB, adoptado para la realización del proyecto Fuente: [7]

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.

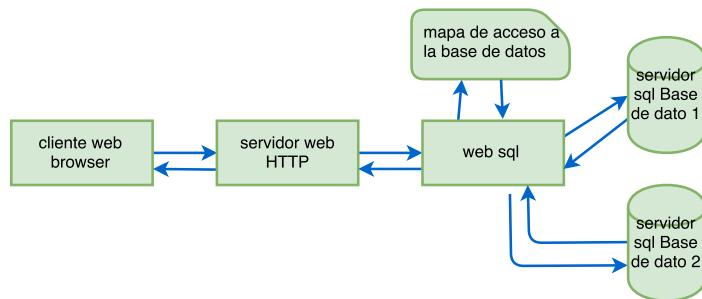


Figura 7.4: Base de datos websql, adoptado para la realización del proyecto Fuente: [8]

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 [14].

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 arreglo 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 autenticación 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:

- a) 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'
  }
};
```

- b) 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 1.4, 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. [13].

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 movil
//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.

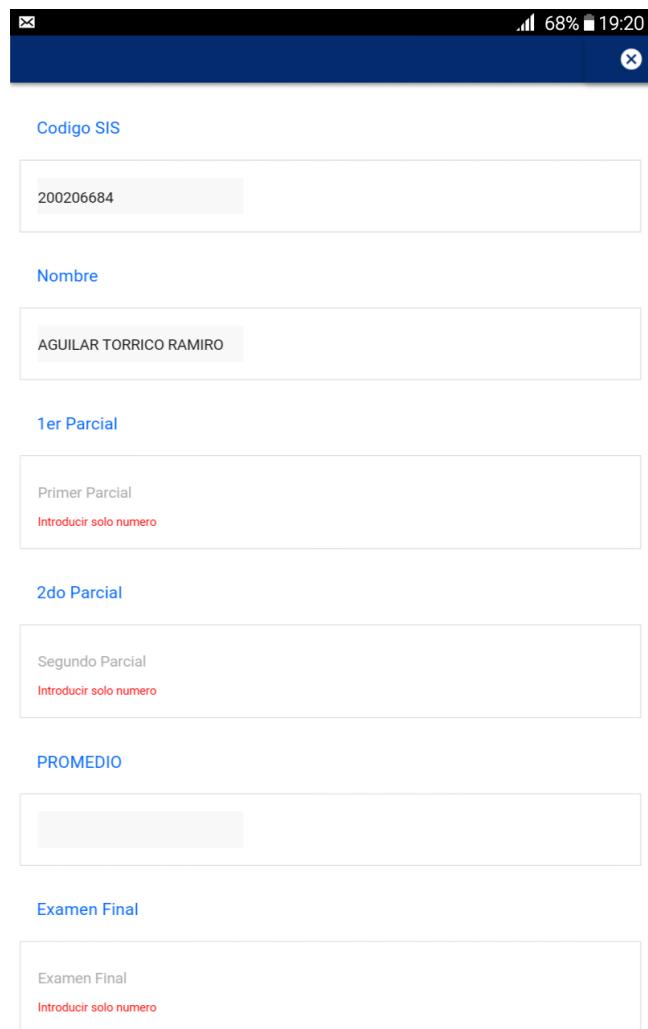


Figura 8.2: Visualizacion de contenido vertical en la tablet, Fuente:
Elaboración propia

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

The screenshot shows a mobile application interface with the following data:

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 /FCB7/detail/e
Hostname was found in DNS cache
Hostname in DNS cache was stale, zapped
HTTP/1.1 200 OK
Date: Fri, 18 Aug 2017 16:54:50 GMT
Server: Apache/2.2.3 (CentOS) is not blacklisted
Content-Type: application/json; charset=UTF-8
Content-Length: 102
Content-Duration: 0
Content-Transfer-Encoding: binary
Content-Disposition: attachment; filename=ERICK_A_MEDRANO_JOSÉ_A_19880028_299701_2_2016.xls
Content-Type: application/octet-stream

```

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  
.am sacando el inicio:88  
.am sacando el final:87  
{ pcd: { __ ' \r,\r,\r,' , head: [ [Object] ], body: [ [Object] ] } }  
--Json-----  
{ pcd: { __ ' \r,\r,\r,' , 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

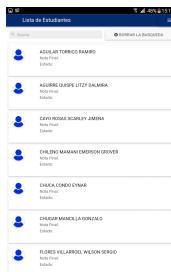


Figura 9.9: La lista de estudiantes, Fuente: Elaboración propia

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.

```

POST /sagaa/pre_academico/subirNotasParcialesF.php HTTP/1.1
Host: pruebas.fcyt.unms.edu.bo
Accept: */*
Cookie: PHPSESSID=4nd4cdvuonis27utfucflnl2qp3
Connection: keep-alive
User-Agent: Mozilla/5.0
Cache-Control: max-age=0
Accept-Encoding: gzip, deflate
Content-Length: 2999
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----09db0a5b1dd64e

-----09db0a5b1dd64e

```

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, AQUI LAR TORRI CO RAM RQ 60,60,60,,60, A----size:50
aux: 1, 200206684, AQUI LAR TORRI CO RAM RQ ,.....----size aux:41
busca nuevo: 1, 200206684, AQUI LAR TORRI CO RAM RQ 60,60,,60, A
busca antiguo: 200206684, AQUI LAR TORRI CO RAM RQ ,.....
dato a ser analizado: 2,201400485, AQUI PRE QUI SPE LI IZY DALM RA ,.....----size:47
aux: 2,201400485, AQUI PRE QUI SPE LI IZY DALM RA ,.....----size aux:47

```

Figura 9.11: El servicio web, busca y reemplaza los datos modificando 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 /sagaa/pre_academico/subirNotasParcialesF.php HTTP/1.1
Host: pruebas.fcyt.unms.edu.bo
Accept: */*
Cookie: PHPSESSID=4nd4cdvuonis27utfucflnl2qp3
Connection: keep-alive
User-Agent: Mozilla/5.0
Cache-Control: max-age=0
Accept-Encoding: gzip, deflate
Content-Length: 2999
Expect: 100-continue
Content-Type: multipart/form-data; boundary=-----09db0a5b1dd64e

-----09db0a5b1dd64e
< HTTP/1.1 100 Continue
< HTTP/1.1 302 Found
< Date: Fri, 18 Aug 2017 19:14:35 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
< Location: subirNotasParcialesF.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.unms.edu.bo (167.157.27.7) port 80 (#1)
> POST /sagaa/pre_academico/subirNotasParcialesFZ.php HTTP/1.1
Host: pruebas.fcyt.unms.edu.bo
Accept: */*
Cookie: PHPSESSID=4nd4cdvuonis27utfucflnl2qp3
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 19:14:35 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
< Content-Length: 411
< Connection: close
< Content-Type: text/html; charset=ISO-8859-1
<
* Closing connection 11

```

Figura 9.13: El servicio web, envía 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 cual responde con un mensaje *Finalizo la habilitación 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 habilitación de estudiantes y el cargado de planillas.
```

Figura 9.14: En el servicio web, es la respuesta de la página del SAGAA,
Fuente: Elaboración 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] Kindberg y Blair Colouris, Dollimore. *Sistemas Distribuidos*. Copyright., 5ta edición, 2012.
- [10] Navarro R. Rest vs web services. *Facultad de Informática - Universidad Politecnica de Valencia*, 2006.
- [11] El desarrollo de aplicaciones móviles nativas, web o híbridas. *Copyright IBM Corporation, Elaborado en los Estados Unidos de América* , 2012.
- [12] J. Morillo. *Introducción a los dispositivos móviles*. PhD thesis, Universidad Oberta de Catalunya, España, 2014.
- [13] A. Vega. *Responsive Web Design: Interfaces Web Adaptables al dispositivo empleado HTML5 Y CSS3*. PhD thesis, Escuela Politécnica Superior de Ingeniería Informática, 2012.
- [14] C. Cardenas. Localstorage. *Blog de Platzi*, 2015.