

Informe final de práctica de especialidad

Práctica Profesional

Unidad desconcentrada de Ingeniería en Computación

Ingeniería de Computación, Plan 410.

Centro Académico Alajuela

Estudiante: Brayan González Chaves

Carnet: 201264796

Contenido

Datos del proyecto	4
Objetivo	4
Contexto del proyecto	4
Stakeholders	5
Estrategia	5
Estrategia para los requerimientos	5
Estrategia de investigación	6
Investigación API Calendar Office 365	6
Investigación Android	6
Investigación Web Sockets	7
Diseño de aplicación	7
Implementación	7
Pruebas y Ajustes	8
Despliegue	9
Requerimientos	9
Requerimientos técnicos:	9
Requerimientos funcionales:	9
Requerimientos no funcionales:	10
Alcances	11
Realización del diseño necesario	11
Implementación de un servidor basado en websocket o REST	11
Creación de una aplicación Android	12
Documentación	12
Entrega de Proyecto	12
Costos	13
Riesgos:	13
Diseño de la solución	15
Arquitectura conceptual	15
Componentes y servicios	16
Cliente HTTP	16
Capa de servicios	16
Diagrama de clases	17
Servidor	17
Móvil Android	20

Conclusiones	23
Cumplimiento de objetivos	23
Descripción de documentos y productos entregados.	23
Código fuente servidor	23
Código fuente de aplicación Android.....	23
Experiencias adquiridas.....	23

Datos del proyecto

Proyecto: Book My Room

Solicitante: Empresa Snap Technology

Patrocinador: Empresa Snap Technology

Director de Proyecto: Manuel Azofeifa

Recursos Técnicos: Estudiante Brayan González

Objetivo

Crear una aplicación Android abastecida por el calendario de eventos de Office 365 para poder gestionar la reservación de salas creada como parte del curso de Práctica Profesional de la carrera de Ingeniería en Computación del Instituto Tecnológico de Costa Rica para la empresa Snap Technology con un costo no mayor a 660 horas, y con un tiempo de entrega estimado de no más de 25 de noviembre del 2016, asumiendo el cumplimiento de los requerimientos.

Contexto del proyecto

Snap Technology es una empresa con sede en Costa Rica creada para apoyar los objetivos, las estrategias, iniciativas, y el éxito final de las empresas en el espacio financiero. La compañía ofrece desarrollo de software, implementación y gestión de la tecnología a otras empresas del grupo y los demás asociados.

El departamento directamente involucrado es el de desarrollo, pero en sí el producto final es para la empresa en general, por lo que cualquier departamento va a hacer uso de este.

Stakeholders

Nombre	Brayan González	Manuel Azofeifa
Ocupación	Estudiante universitario	Director de Tecnología.
Descripción de labores	Análisis de requerimientos, diseño, implementación, pruebas y despliegue del proyecto.	Desarrollo y administración de proyectos.
Responsabilidades del proyecto	Entrega del proyecto en el tiempo definido, así como todos sus entregables.	Seguimiento y control del proyecto.
Criterios de éxito	Creación de una aplicación móvil para la gestión de salas de la empresa Snap Technology, cumpliendo los requerimientos especificados.	Obtención de aplicación por parte del estudiante con todas las funcionalidades propuestas.

Estrategia

La estrategia que se propone es la siguiente:

Estrategia para los requerimientos

1. Análisis de requerimientos: reunión con los involucrados del proyecto para definir las distintas características del proyecto
2. Elaboración del ERS: conforme a las características definidos en el análisis de requerimientos, se realizará un documento de ERS para tener por escrito los reglas que se deben de cumplir, así como las funcionalidades.
3. Historias de usuario: además de las dos estrategias anteriormente, se va a retroalimentar para por medio de historias de usuario para poder definir incluso un primer prototipo de interfaz.

Estrategia de investigación

Investigación API Calendar Office 365

1. Autenticación con el API: se buscará en la documentación del API correspondiente como realizar una autenticación correcta para luego consumir sus métodos.
2. Obtener eventos desde el API: se revisará la documentación del API calendar Office 365 para poder realizar este método.
3. Sincronizar eventos desde el API: se revisará la documentación del API calendar Office 365 para poder realizar este método.
4. Actualizar eventos desde el API: se revisará la documentación del API calendar Office 365 para poder realizar este método.
5. Remover eventos desde el API: se revisará la documentación del API calendar Office 365 para poder realizar este método.
6. Crear eventos desde el API: se revisará la documentación del API calendar Office 365 para poder realizar este método.

Investigación Android

1. Interfaz de usuario: se investigará los diversos componentes que posee Android para la elaboración de una interfaz fácil de usar y a la vez amigable, la documentación de Android Studio se utilizará como referencia.
2. Pruebas en hardware físico: la empresa ha invertido en dispositivos para el proyecto, por lo que se investigará poder realizar directamente pruebas en los dispositivos, para ello se revisará en documentación correspondiente de Android.
3. Estructura de código: para un buen rendimiento y estandarización se investigará en proyectos libres ya creados las estructuras que debe de tener un proyecto para una aplicación Android.

Investigación Web Sockets

1. Formas de implementación: se buscará ejemplos concretos de códigos fuentes ya elaborados para familiarizarse con los web sockets, además de que se revisará la documentación de JAVA para poder observar las distintas formas de implementación y así poder escoger la más apta para el proyecto.
2. Casos de empleo: además de la investigación de formas de implementación, se investigará acorde a los ejemplos de investigación los correctos casos en que se pueden utilizar los web sockets esto para verificar la correcta solución brindada.

Diseño de aplicación

1. Diseño de arquitectura: Conforme a las funcionalidades y tecnologías que demanda la aplicación se realizará una arquitectura basada en capas para la correcta modularización mediante el conocimiento adquirido.
2. Diagramas de clases: para el código Android, el cliente API y los sockets se realizará un diagrama de clases, estableciendo los atributos que deben de tener cada clase, así como las funciones de cada una mediante el conocimiento adquirido.
3. Diagrama entidad relación: mediante diagramas de entidad-relación se mostrará las diversas asociaciones que podrá tener cada entidad mediante el conocimiento adquirido.
4. Diagrama de casos de uso: basándose en los requerimientos y las funciones que el cliente quiere, se realizará un diagrama de casos de uso.

Implementación

1. Desarrollo del socket (Servidor): basándose en la investigación realizada previamente, y el conocimiento adquirido por estudios, se implementará un socket servidor que logre abastecer a los dispositivos Android cumpliendo el diseño establecido.

2. Desarrollo del socket (Cliente): basándose en la investigación realizada previamente, y el conocimiento adquirido por estudios, se implementará un socket cliente capaz de obtener los datos suministrados por el socket servidor cumpliendo el diseño establecido.
3. Desarrollo del cliente http para API Office 365: basándose en investigación anterior, y conocimiento adquirido por estudios se implementará un cliente capaz de consumir el API de Office 365 cumpliendo el diseño establecido.
4. Desarrollo controlador: dado que se conoce que se deben de transmitir datos de un lado para otro, se implementará una capa la cual controlará las distintas peticiones, asociando el modelo de la aplicación, capa que va a ser utilizada como un controlador de peticiones basado en el diseño predefinido anteriormente.
5. Desarrollo de interfaz Android: basado en el diseño definido anteriormente, la interfaz Android se realizará de manera simple siempre y cuando cumpliendo con los requerimientos establecidos.
6. Desarrollo modelo de datos: El modelo de datos se va a basar el modelo propuesto por el API Calendar Office 365, basado en los Objetos JSON que retorna, esto para mayor facilidad de manipulación, además esta implementación se va a basar en el diseño de clases descrito anteriormente.
7. Desarrollo de servicios: se debe también basarse en los diseños de clases descritos anteriormente, van a ser una capa más de la aplicación.

Pruebas y Ajustes

1. Generación de pruebas: Se van a definir las pruebas necesarias para verificar el correcto funcionamiento propuesto por el cliente.
2. Ejecución de pruebas: Una vez teniendo las pruebas, se debe proceder a ejecutarlas.
3. Ajustes a la aplicación: Al ejecutar las pruebas, si alguna prueba de las funciones principales da como resultado erróneo, se debe de realizar los ajustes correspondientes a la aplicación
4. Reporte de pruebas: Una vez habiendo realizado los ajustes si fuese necesario se debe de realizar un informe con el respectivo resultado de las pruebas realizadas

Despliegue

1. Monitoreo de aplicación: Una vez lanzada la aplicación, se debe de monitorear el correcto funcionamiento.

Requerimientos

Requerimientos técnicos:

- **Dispositivo móvil.**

Sistema: Android 4.4 (KitKat) como mínimo

Memoria: 500 MB

Tamaño de pantalla: 5 pulgadas.

Requerimientos funcionales:

Configurar la tableta específicamente para una sala.

Cada sala va a poseer un dispositivo, se quiere que la aplicación sea capaz de poder ser configurada a una determinada sala mediante un identificador para a partir de esta configuración poder gestionar correctamente la reservación de salas.

Esta configuración deberá persistir en caso de que el dispositivo se apague, por lo que sólo si el usuario desea cambiar esta configuración.

Reservar una sala.

Poder realizar la reservación de una sala mediante la aplicación Android que debe ser reflejada en el calendario de eventos de Office 365 que posee la empresa.

Eliminar una reserva de una sala.

Poder realizar la cancelación de la reservación de una sala mediante la aplicación Android que debe ser reflejada en el calendario de eventos de Office 365 que posee la empresa.

Modificar la reserva de una sala.

Poder realizar la modificación de una reservación de una sala mediante la aplicación Android que debe ser reflejada en el calendario de eventos de Office 365 que posee la empresa.

Poder ver el estado de una sala mediante la aplicación.

Poder ver el estado mediado la interfaz de la aplicación Android de la sala, si se encuentra ocupada o no, además dentro de cuánto tiempo será esta liberada.

Mostrar en pantalla el usuario que actualmente posee la sala.

La aplicación Android deberá ser capaz de poder mostrar a los usuarios una imagen correspondiente al usuario que en ese momento está utilizando la sala, esta imagen se extraerá automáticamente del calendario de eventos de Office 365.

Requerimientos no funcionales:

Requerimientos Funcionales del problema

Usabilidad: Las funciones en la interfaz deben ser claras, textos legibles para que el usuario sienta satisfacción al usar la aplicación móvil

Portabilidad: La aplicación debe de funcionar en los dispositivos Android mayor a la versión KitKat.

Disponibilidad: La disponibilidad de la aplicación debe ser continua, a no ser que ocurra algún cambio dentro del código y se deba realizar un nuevo despliegue

Alcances

Realización del diseño necesario

Descripción: Para la correcta implementación de la aplicación se necesita anteriormente crear un diseño correcto que identifique los diversos problemas y una manera eficaz de desarrollo.

Medición: Correcto diseño de arquitectura de aplicación, además de los correspondientes diagramas de clases para cada capa definida.

Responsables: Estudiante Brayan Gonzalez

Tareas:

1. Diagrama de arquitectura: 7 horas
2. Diagramas de clases: 21 horas
3. Modelo de datos: 7
4. Diagrama entidad relación 14 horas
5. Diagrama casos de uso: 14 horas
6. Diseño interfaz Android: 7 horas

Costo Total: 70 horas

Implementación de un servidor basado en websocket o REST

Descripción: existirá un servidor que se va a conectar al API de Office 365, el cual debe suministrar la información necesaria para los dispositivos Android.

Medición: Código fuente el cual permite el correcto funcionamiento del servidor en base a la descripción.

Responsables: Estudiante Brayan González

Tareas:

1. Desarrollo del socket servidor: 14 horas
2. Desarrollo del cliente http para el API Office 365: 28 horas
3. Desarrollo de controlador de peticiones: 35 horas

Costo Total: 77 horas

Creación de una aplicación Android

Descripción: Es necesario la creación de una aplicación Android la cual el usuario va a interactuar con la misma, desde esta se deberán poder realizar todos los requerimientos funcionales estipulados anteriormente.

Medición: Código fuente de una aplicación Android la cual permita realizar los requerimientos funcionales estipulados por el cliente.

Responsables: Estudiante Brayan González

Tareas:

1. Desarrollo de interfaz Android: 105 horas
2. Desarrollo de modelo de datos: 21 horas
3. Desarrollo de servicios: 63 horas

Costo total: 189 horas

Documentación

Descripción: Se realizará toda la documentación necesaria para realizar un pase a producción.

Métrica: Entrega de manual de usuario, que contenga la explicación de todos los requerimientos funcionales

Responsables: Estudiante Brayan González

Costo Total: 10 horas

Tareas:

1. Realización de manual de usuario: 8 horas

Costo total: 10 horas

Entrega de Proyecto

Descripción: Se realizará la entrega del proyecto completo. Junto con la documentación de usuario.

Métrica: Entrega de proyecto.

Responsables: Estudiante Brayan González

Costo Total: 2 horas

Costos

A continuación, Se detallan los costos asociados al proyecto Book My Room

Detalle	Costo
Proyecto	Costo una única vez
1. Diseño de la aplicación	70 h
2. Código fuente de servidor basado en sockets	77 h
3. Código fuente de aplicación móvil	189
4. Manual de usuario	10 h
Total de horas	346 H

Riesgos:

API Calendar Office 365 incompleta

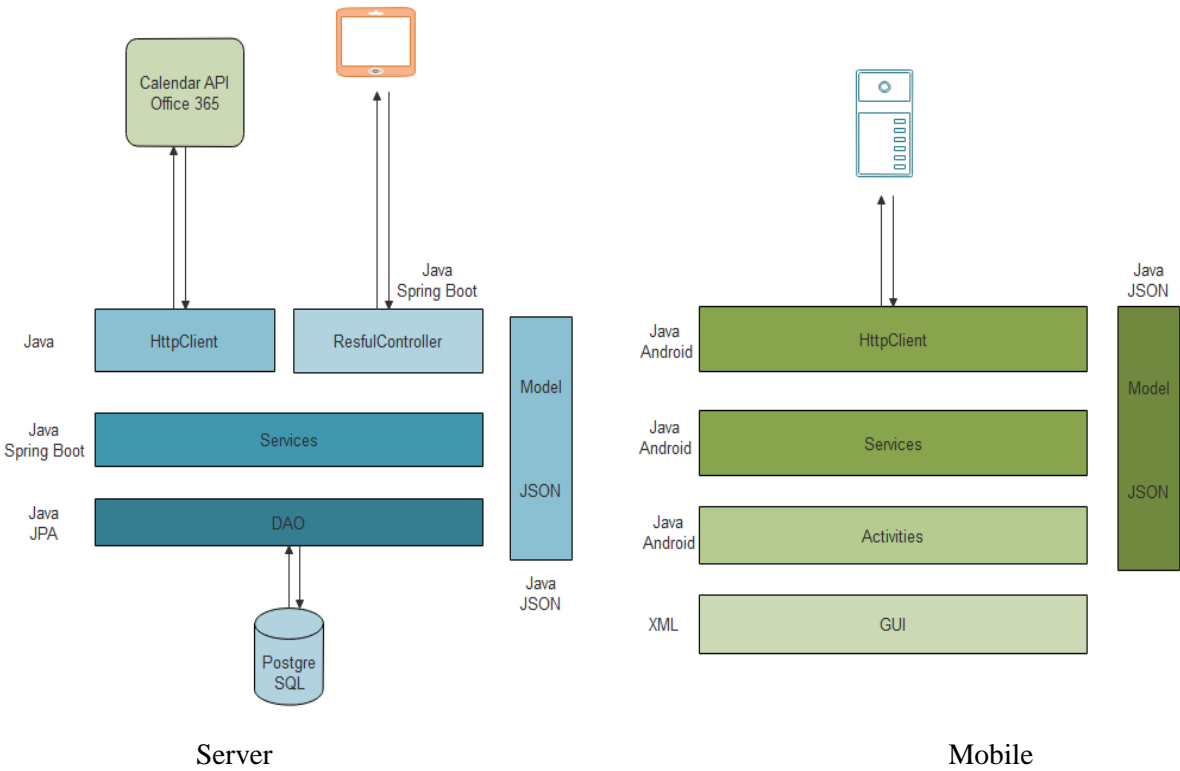
- Descripción: API calendar office 365 incompleta para la correcta gestión de reservación de salas.
- Categoría: tecnológica
- Causas: limitación de métodos desarrollados en el API de Office 365, no poder contar con un método necesario para el proyecto.
- Impacto para el proyecto: 0.9
- Probabilidad de ocurrencia 0.2
- Exposición ante el riesgo: 0.18
- Estrategia de evasión: No se puede evadir
- Estrategia de mitigación: el proyecto depende de la correcta funcionalidad de esta API.
- Escalabilidad del proyecto, y negociación con las partes interesadas para la continuación del proyecto.

Imposibilidad de autenticación Calendar Office 365

- Descripción: para el uso del API calendar office 365 es necesario autenticar con Microsoft, además de una serie de permisos que se deben de otorgar a la cuenta administrativa
- Categoría: tecnológica
- Causas: no obtener las credenciales necesarias para autenticar o que la empresa no tenga la suscripción de Azure AD si fuera necesario.
- Impacto para el proyecto: 0.9
- Probabilidad de ocurrencia 0.5
- Exposición ante el riesgo: 0.45
- Estrategia de evasión: No se puede evadir
- Estrategia de mitigación: el proyecto depende de la correcta funcionalidad de esta API.
- Escalabilidad del proyecto, y negociación con las partes interesadas para la continuación del proyecto.

Diseño de la solución

Arquitectura conceptual



Componentes y servicios

Cliente HTTP

Este módulo es utilizado para realizar las peticiones ya sea el caso de las peticiones que el servidor debe de hacer al API de Office 365, así como de las peticiones que debe realizar cada dispositivo Android al servidor.

Este módulo fue implementado con la biblioteca de Apache Http Client.

Capa de servicios

Se implementaron servicios para la manipulación de los eventos, que en este caso son las reuniones realizadas en las salas, y un servicio para las salas, ambos reciben información del Cliente HTTP para realizar las actualizaciones necesarias a los datos o para resolver las peticiones de los dispositivos Android. Además de un servicio que provee los URL's necesarios para cada petición. Los servicios mencionados cumplen prácticamente la misma lógica ya sea en el servidor o en el dispositivo Android.

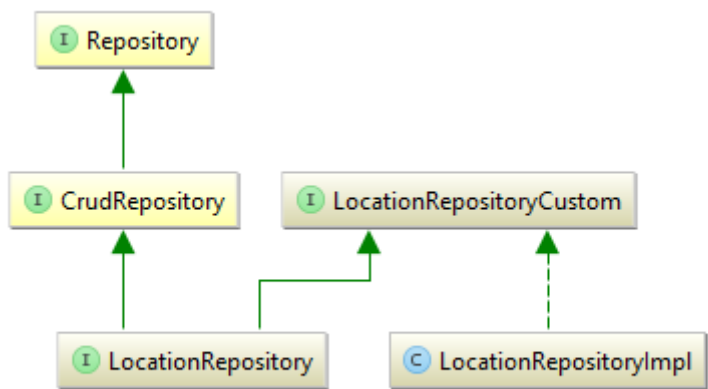
Además, en la aplicación Android si es necesario un servicio adicional, llamado TimeService que es el encargado de la manipulación de con todo lo respecta fechas.

Otra capa de la aplicación es el acceso a datos, este se encuentra implementado con JPA, y su función en la aplicación es la lectura de las salas para enviarlas a los dispositivos Android y parte de las peticiones para el API de Office 365

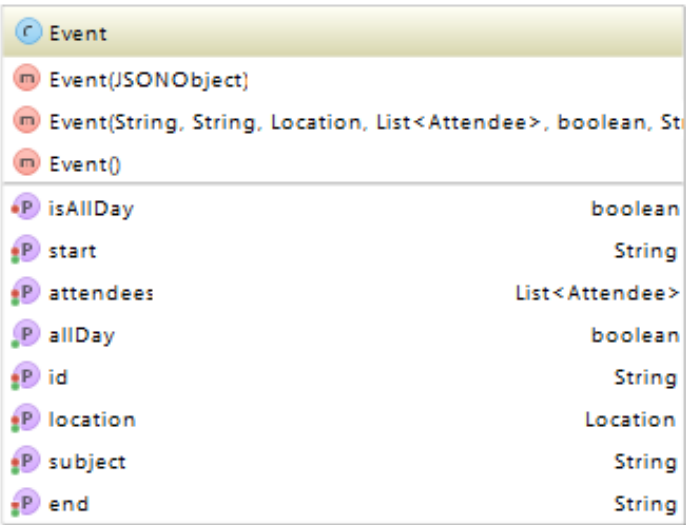
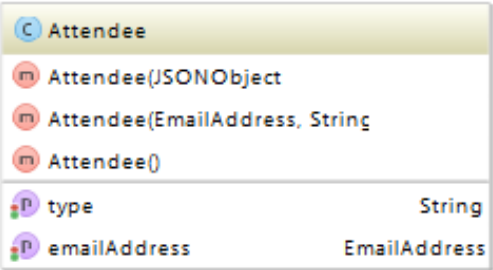
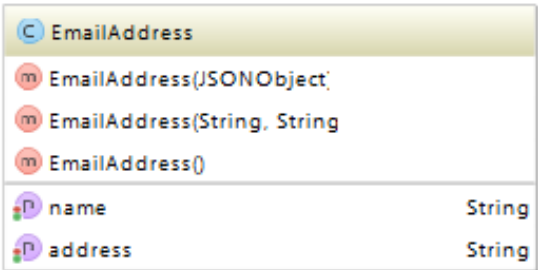
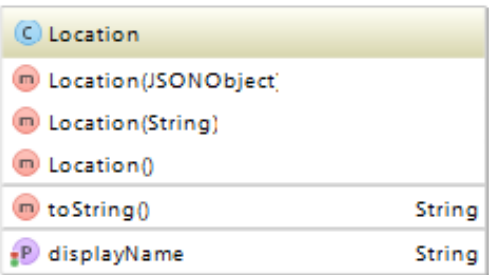
Diagrama de clases

Servidor










Acceso a Datos















Modelo








Servicios



C URLService	
 logger	Logger
 getURLEvents(String, String)	String
 getURLUpdateEvent(Event)	String
 getURLDeleteEvent(Event)	String
 getURLSynchronizeEvents(String, String, String) g	
 delta	String
 path	String
 location	Location
 URLCreateEvent	String

C LocationService	
 repository	LocationRepository
 allLocations	List<Location>

C Office365Service	
 client	ApacheHttpClient
 delta	String
 logger	Logger
 getEvents(EventVO)	List<Event>
 createEvent(Event)	int
 updateEvent(Event)	int
 deleteEvent(Event)	int
 synchronizedEvents(Location, String, String, String)	List<Event>
 setClient(ApacheHttpClient)	void
 delta	String
 urlService	URLService

Restful Controller

C EventController	
 office365Service	Office365Service
 getEvents(EventVO)	List<Event>
 createEvent(Event)	Map<String, Integer>
 updateEvent(Event)	Map<String, Integer>
 deleteEvent(Event)	Map<String, Integer>

C LocationController	
 locationService	LocationService
 findAll()	List<Location>

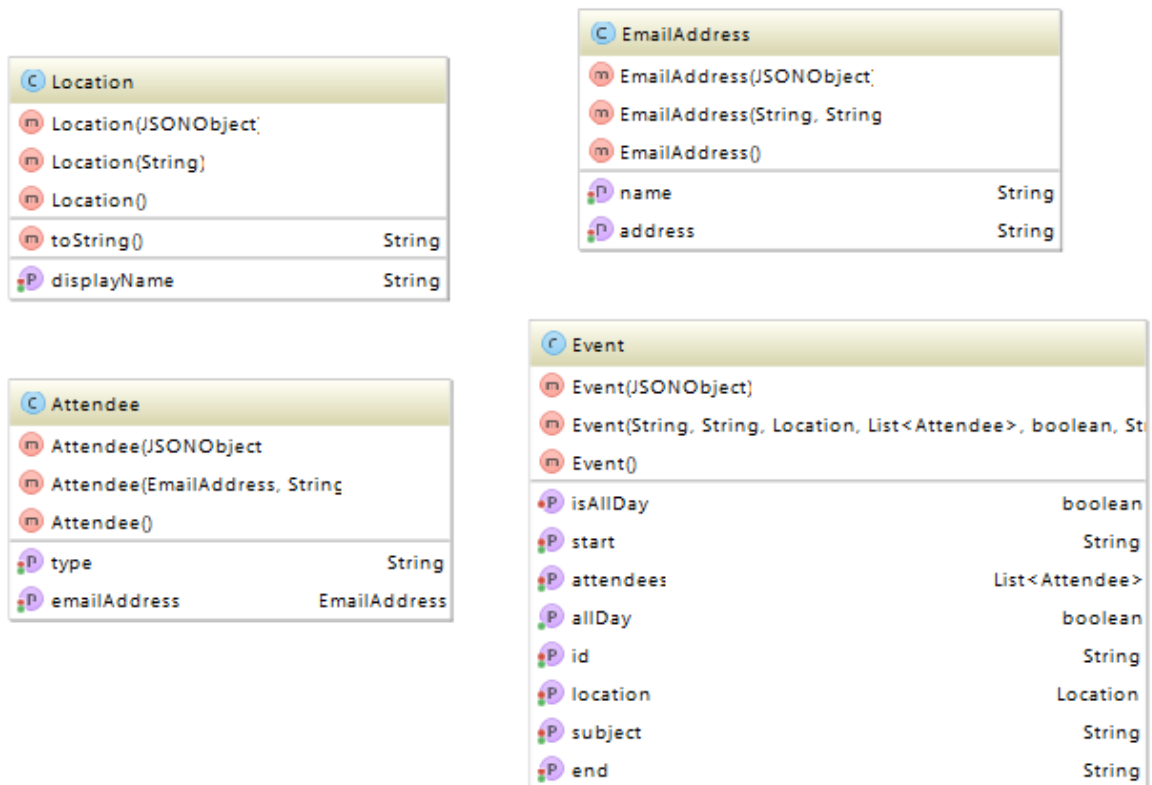
Cliente HTTP

ApacheHttpClient		
log		Logger
client		HttpClient
getRequest		HttpGet
postRequest		HttpPost
deleteRequest		HttpDelete
response		HttpResponse
patchRequest		HttpPatch
br		BufferedReader
output		StringBuffer
password		String
encoder		Encoder
codeBasicAuth		String
getRequest(String)		StatusLine
deleteHttpRequest(String)		StatusLine
postHttpRequest(String, String)		StatusLine
patchHttpRequest(String, String)		StatusLine
setGetRequest(String)		void
setPostRequest(String, String)		void
setDeleteRequest(String)		void
setPatchRequest(String, String)		void
setRequestHeaders(HttpRequestBase)		void
setResponse(HttpRequestBase)		void
setBr()		void
setOutput()		void
setClient()		void
output		String
location		Location

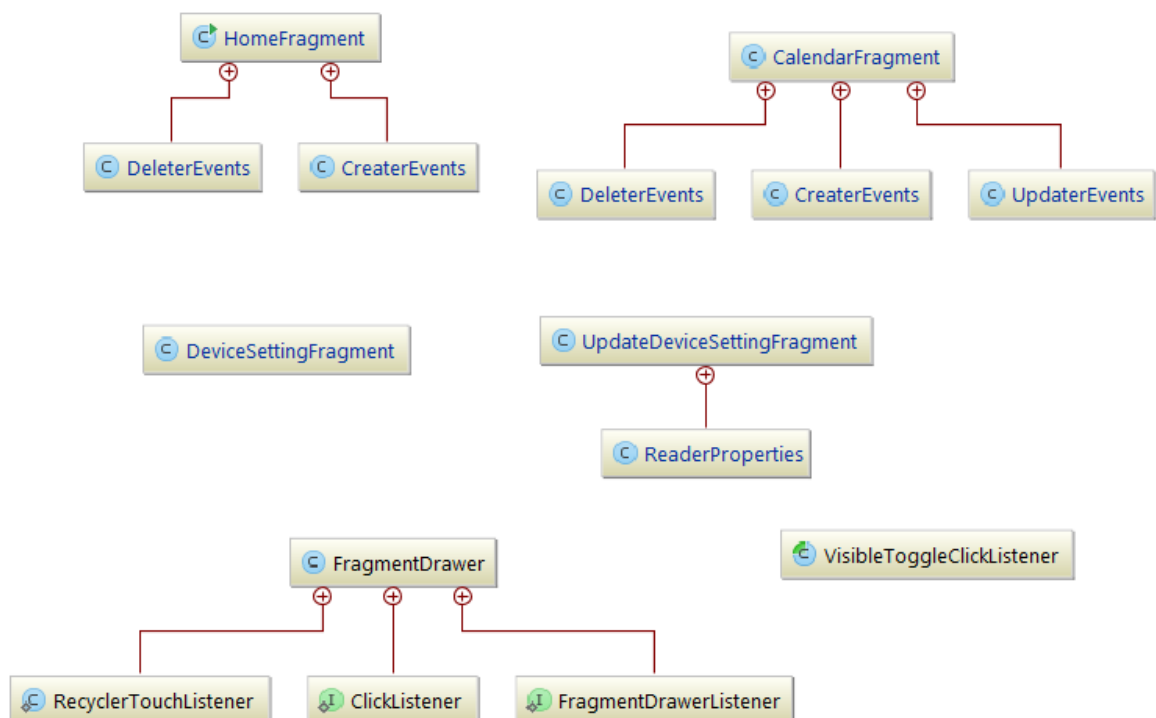
Encoder		
encode(String, String)		String
code		String

Móvil Android

Modelo



Activities



Servicios

EventService		
client	ApacheHttpClient	
urlService	URLService	
context	Context	
instance	EventService	
events	List<Event>	
getInstance(Context)	EventService	
createEvent(Event)	boolean	
updateEvent(Event)	boolean	
deleteEvent(Event)	boolean	
updateEvents()	void	
getEvents()	List<Event>	
setEvents(List<Event>)	void	
timeService	TimeService	
eventMapper	Map<String, Event>	
event	List<Event>	

URLService		
server	String	
EVENT	String	
LOCATION	String	
URLDeleteEvent	String	
URLAllLocations	String	
URLEvents	String	
URLUpdateEvent	String	
URLCreateEvent	String	

LocationService		
client	ApacheHttpClient	
urlService	URLService	
locations	List<Location>	

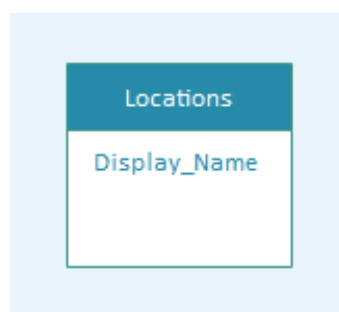
TimeService		
getRangeToRequest()	Map<String, String>	
getInitialID()	String	
getInitialTimeDay()	String	
changeToNextDay(String)	String	
addMinutes(String)	String	
lessMinutes(String)	String	
customizeDate(String)	String	
addMinutes(String, Int)	String	
updateDatesToCalendar()	String[]	
resetHoursStringDate(String)	String	
resetHoursDate(Date)	Date	
addADay(String)	String	
isGreaterDate(String, String)	boolean	
getDates()	String[]	
calculateDifferenceInString(String, String)	String	
calculateDifferenceDates(String, String)	long	
convertStringToDate(String)	Date	
convertDateToString(Date)	String	
roundDateToHigherInString(String)	String	
roundDateToLessInString(String)	String	
getActualTime()	Date	
getInOfActualDay(String)	Int	
convertSimpleHourToComplexHour(String, Int)	String	
convertComplexHourToSimpleHour(String)	String	

Cliente HTTP

ApacheHttpClient	
log	Logger
client	HttpClient
getRequest	HttpGet
postRequest	HttpPost
deleteRequest	HttpDelete
response	HttpResponse
patchRequest	HttpPatch
br	BufferedReader
output	StringBuffer
password	String
encoder	Encoder
codeBasicAuth	String
getHttpRequest(String)	StatusLine
deleteHttpRequest(String)	StatusLine
postHttpRequest(String, String)	StatusLine
patchHttpRequest(String, String)	StatusLine
setGetRequest(String)	void
setPostRequest(String, String)	void
setDeleteRequest(String)	void
setPatchRequest(String, String)	void
setRequestHeaders(HttpRequestBase)	void
setResponse(HttpRequestBase)	void
setBr()	void
setOutput()	void
setClient()	void
output	String
location	Location

Encoder	
encode(String, String)	String
code	String

Diseño de la base de datos



Conclusiones

Cumplimiento de objetivos

Los requerimientos propuestos por el cliente al inicio del proyecto fueron cumplidos en su totalidad, se ha creado una aplicación capaz de gestionar las salas de una empresa sincronizando los eventos con la plataforma de Office 365, con un rendimiento satisfactorio.

Descripción de documentos y productos entregados.

Código fuente servidor

Se entrega al cliente el código del servidor, al cual los dispositivos móviles realizan las peticiones con la cuenta configurada en ellos, este servidor se conecta con la plataforma de Office 365 para sincronizar los eventos creados desde ahí, y además los eventos creados desde los dispositivos móviles.

Código fuente de aplicación Android

Se entrega el código fuente de la aplicación Android, capaz de gestionar los eventos de una sala, previamente configurada, dicha aplicación se conecta al servidor mencionado anteriormente para obtener la información necesaria.

Manual de Usuario

Manual de usuario explicando el funcionamiento de la aplicación.

Experiencias adquiridas

Considero la práctica profesional muy importante como parte de la formación que obtuve por medio del Tecnológico de Costa Rica, aparte de ser una etapa en la carrera en que se deben de poner en práctica, en una situación real, los conocimientos adquiridos, es una etapa en la cual nos ayuda a adaptarnos a una empresa, y poder entrar en el mercado laboral en correctas condiciones, dado que además la formación académica del Tecnológico de Costa Rica la considero una de las mejores.

Además, el hecho de crear un proyecto sólo para una empresa fue una buena experiencia, en mi caso gracias a ello adquirí conocimiento en desarrollo de aplicaciones móviles, que pienso es un conocimiento importante que me puede ayudar en un futuro como Ingeniero.